

ESM 2003

SCIENTIFIC PROGRAM

KEYNOTE PAPERS

Function Approximation by Random Neural Networks with a Bounded Number of Layers

Erol Gelenbe, *Fellow IEEE* *
School of Electrical Engineering & Computer Science
University of Central Florida
Orlando, FL 32816
erol@cs.ucf.edu Fax: 407 823 5419

Zhi-Hong Mao and Yan-Da Li †
Department of Automation
Tsinghua University
Beijing 100084
P.R. China
maozh@jerry.au.tsinghua.edu.cn

February 14, 2001

Abstract

This paper discusses the function approximation properties of the “Gelenbe” random neural network (GNN) [5, 6, 9]. We use two extensions of the basic model: the bipolar GNN (BGNN) [7] and the clamped GNN (CGNN). We limit the networks to being feedforward and consider the case where the number of hidden layers does not exceed the number of input layers. With these constraints we show that the feedforward CGNN and the BGNN with s hidden layers (total of $s + 2$ layers) can uniformly approximate continuous functions of s variables.

1 Introduction

A novel neural network model – the GNN or “Gelenbe’s Random Neural Network” [5, 6, 9] – has had significant applications in various engineering areas [7, 8, 10, 11, 12, 13, 14, 15], using the network’s learning algorithm [9] or its ability to act as an optimizing network. These random neural network differ significantly from standard connexionist models in that information travels between neurons in this model in the form of random spike trains, and network state is represented by the probability distributions that the n neurons in the network are excited. These

*This author’s work was supported by the Office of Naval Research under Grant No. N00014-97-1-0112, and is currently supported by NAWC Grants No. N61339-97-K-005 and N61339-00-K-002.

†These authors’ work was supported by the National Science Foundation of the P.R. of China under Grant No. 69682010.

models have a mathematical structure which is significantly different from that of the sigmoidal connexionist model, the Hopfield model, or the Boltzman machine [3]. Thus the approximation capability of these networks also needs to be established in a manner distinct from that of previous models [4]. In particular, the ‘‘Gelenbe’’ random neural network model [5, 6, 9] does not use sigmoid functions which are basic to the standard models’ approximation capabilities.

In recent work [16] we have studied the approximation of arbitrary continuous functions on $[0, 1]^s$ using the GNN. We have shown that the clamped GNN and the bipolar GNN [7] have the universal approximation property, using a constructive method which actually exhibits networks constructed from a polynomial approximation of the function to be approximated. However the constructions in [16] place no restrictions on the structure of the networks except for limiting them to being feedforward. For instance, in [16] we were not able to limit the size of the network as a function of other meaningful characteristics such as the number of input variables or the number of layers.

In this paper we will discuss the design of GNN approximators with a bounded number of layers. In Section 2, a brief introduction to the GNN and the bipolar GNN (BGNN) is given. In Section 3, we establish the technical premises for our main results. Then in Section 4 we prove the universal approximation capability of the feedforward BGNN and CGNN when the number of input variables is bounded. The last section presents conclusions.

2 The GNN and its Extensions

Consider a GNN [5, 6, 9] with n neurons in which ‘‘positive’’ and ‘‘negative’’ signals circulate. The i -th neuron’s state is represented at any time t by its ‘‘potential’’ $k_i(t)$, which is a non-negative integer. In the RNN (Gelenbe (1989,90) [5, 6]) signals in the form of spikes of unit amplitude circulate among the neurons. Positive signals represent excitation and negative signals represent inhibition. Each neuron’s state is a non-negative integer called its potential, which increases when an excitation signal arrives to it, and decreases when an inhibition signal arrives. An excitatory spike is interpreted as a ‘‘+1’’ signal at a receiving neuron, while an inhibitory spike is interpreted as a ‘‘-1’’ signal. Neural potential also decreases when the neuron fires. Thus a neuron i emitting a spike, whether it be an excitation or an inhibition, will lose potential of one unit, going from some state whose value is k_i to the state of value $k_i - 1$. In general, this is a ‘‘recurrent network’’ model, *i.e.* a network which is allowed to have feedback loops of arbitrary topology.

The state of the n -neuron network at time t , is represented by the vector of non-negative integers $k(t) = (k_1(t), \dots, k_n(t))$, where $k_i(t)$ is the potential or integer state of neuron i . We will denote by k and k_i arbitrary values of the state vector and of the i -th neuron’s state. Neuron i will ‘‘fire’’ (*i.e.* become excited and send out spikes) if its potential is *positive*. The spikes will then be sent out at a rate $r(i) \geq 0$, with independent, identically and exponentially distributed inter-spike intervals. Spikes will go out to some neuron j with probability $p^+(i, j)$ as excitatory signals, or with probability $p^-(i, j)$ as inhibitory signals. A neuron may also send signals out of the network with probability $d(i)$, and $d(i) + \sum_{j=1}^n [p^+(i, j) + p^-(i, j)] = 1$. Figure ?? shows the representation of a neuron in the RNN.

Exogenous excitatory signals arrive to neuron i in a Poisson stream of rate $\Lambda(i)$. Similarly exogenous inhibitory signals arrive to neuron i in a Poisson stream of rate $\lambda(i)$. These different

Poisson streams for $i = 1, \dots, n$ are independent of each other. To simplify the notation, in the sequel we will write:

$$\omega^+(i, j) = r(i)p^+(i, j), \quad (1)$$

$$\omega^-(i, j) = r(i)p^-(i, j). \quad (2)$$

The state transitions of the network are represented by Chapman-Kolmogorov equations [1] for the probability distribution:

$$p(k, t) = Prob[k(t) = k], \quad (3)$$

where $k = (k_1, \dots, k_n)$ denotes a particular value of the state vector.

Let $\lambda^+(i)$ and $\lambda^-(i)$ denote the average arrival rates of positive and negative signals to each neuron i . The key results about the GNN developed in [5, 6, 9] are summarized below.

Theorem 1. (Proposition 1 in the Appendix of [9]) *There always exists a non-negative solution ($\lambda^+(i) \geq 0$, $\lambda^-(i) \geq 0$) to the equations:*

$$\lambda^+(i) = \Lambda(i) + \sum_{j=1}^n q_j \omega^+(j, i), \quad (4)$$

$$\lambda^-(i) = \lambda(i) + \sum_{j=1}^n q_j \omega^-(j, i), \quad (5)$$

for $i = 1, \dots, n$ where

$$q_i = \frac{\lambda^+(i)}{r(i) + \lambda^-(i)}. \quad (6)$$

The next important result concerns the stationary joint probability distribution of network state:

$$p(k) = \lim_{t \rightarrow \infty} p(k, t). \quad (7)$$

Theorem 2. (Theorem 1 of [5]) *For an n neuron GNN, let the vector of neuron potentials at time t be $k(t) = (k_1(t), k_2(t), \dots, k_n(t))$, and let $k = (k_1, k_2, \dots, k_n)$ be an n vector of non-negative integers. Then if the q_i in (6) satisfy $0 \leq q_i < 1$, the stationary probability of network state is given by:*

$$p(k) = \prod_{i=1}^n (1 - q_i) q_i^{k_i}. \quad (8)$$

Note that if the conditions of Theorem 2 are satisfied then the stationary probability distribution of the state of neuron i denoted by $p(k_i) = \lim_{t \rightarrow \infty} p(k_i(t) = k_i)$, is given by:

$$p(k_i) = (1 - q_i) q_i^{k_i}, \quad (9)$$

and

$$q_i = \lim_{t \rightarrow \infty} Prob\{k_i(t) > 0\}. \quad (10)$$

2.1 The BGNN Model

In order to represent bipolar patterns taking values such as $\{+1, -1\}$, and to strengthen the associative memory capabilities of the GNN, in some early work Gelenbe, Stafylopatis and Likas [7] extended the original model by introducing the artifact of “positive and negative” neurons. The resulting Bipolar GNN (BGNN) can also be viewed as the coupling of two complementary standard GNN models.

In the BGNN the two types of neurons have opposite roles. A positive neuron behaves exactly as a neuron in the original GNN. A negative neuron has a completely symmetrical behavior, namely only negative signals can accumulate at this neuron, and the role of positive signals arriving to a negative neuron is to eliminate negative signals which have accumulated in a negative neuron’s potential. A positive signal arriving to a negative neuron i cancels a negative signal (adds +1 to the neuron’s negative potential), and has no effect if $k_i = 0$.

This extension is in fact mathematically equivalent to the original GNN described above, with respect to the specific form taken by the stationary solution (Theorems 1 and 2). However the use of both positive and negative neurons allows the BGNN to become convenient universal approximator for continuous functions because of the possibility of using both positive and negative valued functions of the input variables. Let P and N denote, respectively, the indices of the positive and negative neurons in the network. In the BGNN the state of the network is represented by the vector $k(t) = (k_1(t), \dots, k_n(t))$ so that $k_i(t) \geq 0$ if $i \in P$ and $k_i(t) \leq 0$ if $i \in N$.

In the BGNN, the emission of signals from a positive neuron is the same as in the original GNN. Similarly, a negative neuron may emit negative signals. A signal leaving negative neuron i arrives to neuron j as a negative signal with probability $p^+(i, j)$ and as a positive signal with probability $p^-(i, j)$. Also, a signal departs from the network upon leaving neuron i with probability $d(i)$. Other assumptions and denotations retain as in the original model.

Let us consider a BGNN with n nodes. Since negative signals account for the potential of negative neurons, we will use negative values for k_i if neuron i is negative. If we take into account the distinction between positive and negative neurons, Theorems 1 and 2 can be summarized as follows for the BGNN. The flow of signals in the network is described by the following equations:

$$\lambda^+(i) = \Lambda(i) + \sum_{j \in P} q_j \omega^+(j, i) + \sum_{j \in N} q_j \omega^-(j, i), \quad (11)$$

$$\lambda^-(i) = \lambda(i) + \sum_{j \in P} q_j \omega^-(j, i) + \sum_{j \in N} q_j \omega^+(j, i), \quad (12)$$

and

$$q_i = \frac{\lambda^+(i)}{r(i) + \lambda^-(i)}, \quad i \in P, \quad (13)$$

$$q_i = \frac{\lambda^-(i)}{r(i) + \lambda^+(i)}, \quad i \in N. \quad (14)$$

Using a direct extension of the results for the conventional GNN, it can be shown that a non-negative solution $\{\lambda^+(i), \lambda^-(i), i = 1, \dots, n\}$ exists to the above equations. If the $q_i < 1$, $i = 1, \dots, n$, then the steady-state joint probability distribution of network state is given by [7]:

$$p(k) = \prod_{i=1}^n (1 - q_i) q_i^{|k_i|}, \quad (15)$$

where the quantity q_i is the steady-state probability that node i is “excited”. Note the $|k_i|$ exponent in the above product form, since the k_i 's can be positive or negative, depending on the polarity of the $i - th$ neuron. In the sequel we will consider how the BGNN, as well as a simpler extension of the feedforward (i.e. non-recurrent) GNN, can be used to approximate arbitrary continuous functions.

Another extension of the GNN – the clamped GNN (CGNN) – will be introduced in Section 3.3.

3 Approximation of Functions of One Variable by the GNN with a Bounded Number of Layers

All feedforward models considered in this section are guaranteed to have an unique solution for the $q_i, i = 1, \dots, r\theta$ as a result of Theorems 2 and 3 of [6]. Thus from now on we do not revisit this issue.

Consider a continuous function $f : [0, 1]^s \mapsto R$ of an input vector $X = (x_1, \dots, x_s)$. Since an $[0, 1]^s \mapsto R^w$ function can always be separated into a group of w distinct functions $[0, 1]^s \mapsto R$, we will only consider outputs in one dimension. The sequel of this paper is therefore devoted to how a continuous function $f : [0, 1]^s \mapsto R$ can be approximated by neural networks derived from the GNN model. To approximate f , we will construct s -input, 1-output, L -layer feedforward GNN's. We will use the index (l, i) for the $i - th$ neuron at the $l - th$ layer. Furthermore, when we need to specify this, we will denote by M_l the number of neurons in the $l - th$ layer.

The network under consideration is organized as follows:

- In the first layer, i.e. the input layer, we set $\Lambda(1, i) = x_i$, $\lambda(1, i) = 0$, $r(1, i) = 1$, so that $q_{1,i} = x_i$, for $i = 1, \dots, s$.
- In the l -th layer ($l = 2, \dots, L$), $\Lambda(l, i)$, $\lambda(l, i)$, and $r(l, i)$ are adjustable parameters, and $q_{l,i}$ is given by

$$q_{l,i} = \frac{\Lambda(l, i) + \sum_{1 \leq h < l} \sum_{1 \leq j \leq M_h} q_{h,j} \omega^+((h, j), (l, i))}{\lambda(l, i) + r(l, i) + \sum_{1 \leq h < l} \sum_{1 \leq j \leq M_h} q_{h,j} \omega^-((h, j), (l, i))} \quad (16)$$

where the connection “weights” $\omega^+(\cdot, \cdot)$ and $\omega^-(\cdot, \cdot)$ are also adjustable parameters.

- In the $L - th$ or output layer there is only one neuron. As suggested in [5] we can use the output function

$$A_{L,1} = \frac{q_{L,1}}{1 - q_{L,1}} \quad (17)$$

whose physical meaning is that it is the average potential of the output neuron as the output of the network. In this manner, we will have $A_{L,1} \in [0, +\infty)$, rather than just $q_{L,1} \in [0, 1]$.

3.1 Technical Premises

Before we proceed with the developments concerning GNN approximations we need some technical results. They are similar to some technical results used in [16] concerning continuous and

bounded functions $f : [0, 1] \mapsto R$ for a scalar variable x . The generalization to $f : [0, 1]^s \mapsto R$ is direct and will be examined in Section 4. The proofs are given in the Appendix.

Lemma 1. *For any continuous and bounded $f : [0, 1] \mapsto R$ and for any $\epsilon > 0$, there exists a polynomial*

$$P(x) = c_0 + c_1 \left(\frac{1}{1+x}\right) + \dots + c_m \left(\frac{1}{1+x}\right)^m, \quad 0 \leq x \leq 1, \quad (18)$$

such that $\sup_{x \in [0, 1]} |f(x) - P(x)| < \epsilon$ is satisfied.

The second technical result concerns the relationship between polynomials of the form (18) and the GNN.

Lemma 2. *Consider a term of the form*

$$\frac{1}{(1+x)^v},$$

for $0 \leq x \leq 1$, and any $v = 1, 2, \dots$. There exists a feedforward GNN with v hidden neurons and input $x \in [0, 1]$ such that

$$q_{v+1,1} = \left(\frac{1}{1+x}\right)^v. \quad (19)$$

The following Lemma shows that any arbitrary polynomial of degree m with positive coefficients can be realized by a feedforward GNN.

Lemma 3. *Let $P^+(x)$ be a polynomial of the form (18) with the restriction that $c_v \geq 0$, $v = 1, \dots, m$. Then there exists a feedforward GNN with a single output neuron (O) such that:*

$$q_O = \frac{P^+(x)}{1 + P^+(x)}, \quad (20)$$

so that the average potential of the output neuron is $A_O = P^+(x)$.

The fourth technical result will be of use in proving the approximating power of the ‘clamped GNN’ discussed below.

Lemma 4. *Consider a term of the form*

$$\frac{x}{(1+x)^v},$$

for $0 \leq x \leq 1$, and any $v = 1, \dots, m$. There exists a feedforward GNN with a single output neuron ($v + 1, 1$) and input $x \in [0, 1]$ such that

$$q_{v+1,1} = \left(\frac{x}{1+x}\right)^v. \quad (21)$$

We state without proof another lemma, very similar to Lemma 3, but which uses terms of both forms of $1/(1+x)^v$ and $x/(1+x)^v$ to construct polynomials. Its proof uses Lemma 3 and 4, and follows exactly the same lines as Lemma 3.

Lemma 5. Let $P^o(x)$ be a polynomial of the form

$$P^o(x) = c_0 + \sum_{v=1}^m [c_v \frac{1}{(1+x)^v} + d_v \frac{x}{(1+x)^v}], \quad 0 \leq x \leq 1, \quad (22)$$

with non-negative coefficients, i.e. $c_v, d_v \geq 0, v = 1, \dots, m$. Then there exists a feedforward GNN with a single output neuron (O) such that:

$$q_O = \frac{P^o(x)}{1 + P^o(x)}, \quad (23)$$

so that the average potential of the output neuron is $A_O = P^o(x)$.

The next lemma is a technical premise of Lemma 7.

Lemma 6. For any $(\frac{1}{1+x})^i$ ($0 \leq x \leq 1, i = 1, 2, \dots$) and for any $\epsilon > 0$, there exists a function

$$P_1(x) = b_0 + \frac{b_1}{x + a_1} + \frac{b_2}{x + a_2} + \dots + \frac{b_r}{x + a_r}, \quad 0 \leq x \leq 1, \quad (24)$$

where $a_k > 0, k = 1, \dots, r$, such that $\sup_{x \in [0,1]} |(\frac{1}{1+x})^i - P_1(x)| < \epsilon$ is satisfied.

Proof: We proceed by induction. For $i = 1$, the conclusion obviously holds. Now assume it is true for $i = j$, i.e., for any $\epsilon > 0$, there exists a

$$P^{(j)}(x) = b_0^{(j)} + \frac{b_1^{(j)}}{x + a_1^{(j)}} + \frac{b_2^{(j)}}{x + a_2^{(j)}} + \dots + \frac{b_m^{(j)}}{x + a_m^{(j)}}, \quad 0 \leq x \leq 1, \quad (25)$$

where $a_k^{(j)} > 0, k = 1, \dots, m$, such that $\sup_{x \in [0,1]} |(\frac{1}{1+x})^j - P^{(j)}(x)| < \epsilon$.

Then for $i = j + 1$,

$$\left(\frac{1}{1+x}\right)^{j+1} = \left(\frac{1}{1+x}\right)^j \left(\frac{1}{1+x}\right) = b_0^{(j)} \frac{1}{1+x} + \sum_{k=1}^m \frac{b_k^{(j)}}{x + a_k^{(j)}} \frac{1}{1+x}. \quad (26)$$

When $a_k^{(j)} \neq 1$,

$$\frac{b_k^{(j)}}{x + a_k^{(j)}} \frac{1}{1+x} = \frac{b_k^{(j)}}{a_k^{(j)} - 1} \left(\frac{1}{1+x} - \frac{1}{x + a_k^{(j)}} \right) \quad (27)$$

which is in the form of (24). When $a_k^{(j)} = 1$,

$$\left(\frac{1}{1+x}\right)^2 = \lim_{\eta \rightarrow 0} \frac{1}{(1-\eta+x)(1+\eta+x)} = \lim_{\eta \rightarrow 0} \frac{1}{2\eta} \left(\frac{1}{1-\eta+x} - \frac{1}{1+\eta+x} \right) \quad (28)$$

which can be arbitrarily approximated by a function of the form (24).

Therefore $(\frac{1}{1+x})^{j+1}$ can also be approximated by a function in the form of (24). Through mathematical induction, the conclusion holds for any $i = 1, 2, \dots$. **Q.E.D.**

The following lemma is the preparation for the construction of a single-hidden-layered BGNN for the approximation of one dimensional continuous function.

Lemma 7. For any continuous function $f : [0, 1] \mapsto R$ and for any $\epsilon > 0$, there exists a function $P_1(x)$ in the form of (24) such that $\sup_{x \in [0,1]} |f(x) - P_1(x)| < \epsilon$ is satisfied.

Proof: This is a direct consequence of Lemma 1 and Lemma 6. **Q.E.D.**

3.2 BGNN Approximation of Continuous Functions of One Variable

The technical results given above now pave the way for the use of the Bipolar GNN (BGNN) with a bounded number of layers. Specifically in Theorem 4, we show that a BGNN with a single hidden layer can uniformly approximate functions of one variable. The multivariable case is discussed in Section 4.

Let us first recall a result from [16] concerning the case when the number of layers is not bounded.

Theorem 3. *For any continuous function $f : [0, 1] \mapsto R$ and any $\epsilon > 0$, there exists a BGNN with one positive output neuron $(O, +)$, one negative output neuron $(O, -)$, the input variable x , and the output variable $y(x)$ such that:*

$$y(x) = A_{O,+} + A_{O,-}, \quad (29)$$

$$A_{O,+} = \frac{q_{O,+}}{1 - q_{O,+}}, \quad (30)$$

$$A_{O,-} = \frac{-q_{O,-}}{1 - q_{O,-}}, \quad (31)$$

and $\sup_{x \in [0,1]} |f(x) - y(x)| < \epsilon$. We will say that the BGNN's output uniformly approximates $f(x)$.

Proof: The result is a direct application of Lemmas 1 and 3. Apply Lemma 1 to f and express the approximating polynomial as $P(x) = P^+(x) + P^-(x)$ so that the coefficients of $P^+(x)$ are nonnegative, while the coefficients of $P^-(x)$ are negative:

$$P^+(x) = \sum_{i=1}^m \max\{0, c_i\} \left(\frac{1}{1+x}\right)^i, \quad (32)$$

$$P^-(x) = \sum_{i=1}^m \min\{0, c_i\} \left(\frac{1}{1+x}\right)^i. \quad (33)$$

Now simply apply Lemma 3 to obtain the feedforward GNN with an output neuron $(O, +)$ whose value is

$$q_{O,+} = \frac{P^+(x)}{1 + P^+(x)}, \quad (34)$$

and the average potential of the output neuron is $A_{O,+} = P^+(x)$. Similarly, using the non-negative polynomial $|P^-(x)|$ construct a feedforward BGNN which has positive neurons throughout, except for its output neuron, along the ideas of Lemma 4. It's output neuron $(O, -)$ however is a negative neuron, yet all the parameter values are the same as those prescribed in Lemma 4 for the output neuron, as they relate to the polynomial $|P^-(x)|$. Thus the output neuron takes the value

$$q_{O,-} = \frac{|P^-(x)|}{1 + |P^-(x)|}, \quad (35)$$

and the average potential is

is $A_{O,-} = -|P^-(x)|$, completing the proof.

Q.E.D.

The next theorem shows the approximation capability of a BGNN with a single hidden layer.

Theorem 4. *For any continuous function $f : [0, 1] \mapsto R$ and any $\epsilon > 0$, there exists a BGNN of three layers (only one hidden layer), one positive output neuron $(O, +)$, one negative output*

neuron $(O, -)$, the input variable x , and the output variable $y(x)$ determined by (29) such that $\sup_{x \in [0,1]} |f(x) - y(x)| < \epsilon$.

Proof: The result is obtained by using Lemma 7. Applying Lemma 7 to f we express the approximating function as $P_1(x) = P_1^+(x) + P_1^-(x)$ so that the coefficients of $P_1^+(x)$ are non-negative, while the coefficients of $P_1^-(x)$ are negative:

$$P_1^+(x) = \max\{0, b_0\} + \sum_{k=1}^r \max\{0, b_k\} \frac{b_k}{x + a_k}, \quad (36)$$

$$P_1^-(x) = \min\{0, b_0\} + \sum_{k=1}^r \min\{0, b_k\} \frac{b_k}{x + a_k}. \quad (37)$$

Now construct a BGNN of three layers: one output layer with one positive output neuron $(O, +)$ and one negative output neuron $(O, -)$ in it, one input layer with one input neuron $(1, 1)$ and r other input neurons $(2, 1), \dots, (2, r)$ in it. Now set:

- $\Lambda(1, 1) = x$, $\lambda(1, 1) = 0$, $r(1, 1) = 1$, $d(1, 1) = 0$,
- $\omega^+((1, 1), (2, k)) = 0$, $\omega^-((1, 1), (2, k)) = 1/r$,
 $r(2, k) = a_k/r$, $\Lambda(2, k) = a_k/r$, $\lambda(2, k) = 0$, $k = 1, \dots, r$,
- $p^+((2, k), (O, +)) = p^-((2, k), (O, +)) = (\max\{b_k, 0\}r)/(2a_k^2 C_{MAX})$,
 $p^+((2, k), (O, -)) = p^-((2, k), (O, -)) = (|\min\{b_k, 0\}|r)/(2a_k^2 C_{MAX})$,
for $k = 1, \dots, r$, where $C_{MAX} = \max\{1, |b_0|, \frac{|b_k|r}{a_k^2}, k = 1, \dots, r\}$,
- $\Lambda(O, +) = \lambda(O, +) = \max\{b_0, 0\}/(2C_{MAX})$, $r(O, +) = 1/(2C_{MAX})$,
 $\Lambda(O, -) = \lambda(O, -) = |\min\{b_0, 0\}|/(2C_{MAX})$, $r(O, -) = 1/(2C_{MAX})$.

It is easy to see that $q_{1,1} = x$, and that

$$q_{2,k} = \frac{a_k}{a_k + x}, \quad k = 1, \dots, r, \quad (38)$$

$$q_{O,+} = \frac{\frac{P^+(x)}{2C_{MAX}}}{\frac{1}{2C_{MAX}} + \frac{P^+(x)}{2C_{MAX}}} = \frac{P^+(x)}{1 + P^+(x)}, \quad (39)$$

$$q_{O,-} = \frac{\frac{|P^-(x)|}{2C_{MAX}}}{\frac{1}{2C_{MAX}} + \frac{|P^-(x)|}{2C_{MAX}}} = \frac{|P^-(x)|}{1 + |P^-(x)|}. \quad (40)$$

Therefore, $A_{O,+} = P^+(x)$, $A_{O,-} = -|P^-(x)|$, and $y(x) = P_1(x)$, completing the proof. **Q.E.D.**

3.3 CGNN Approximation of Continuous Functions of One Variable

We can also demonstrate the approximating power of a normal feedforward GNN by just adding a ‘‘clamping constant’’ to the average potential of the output neuron. We call this extension

the “clamped GNN (CGNN)” since the additive constant c resembles the clamping level in an electronic clamping circuit. Let us first see the corresponding result from our previous work [16].

Theorem 5. *For any continuous function $f : [0, 1] \mapsto R$ and any $\epsilon > 0$, there exists a GNN with ~~two~~ *output neurons* $(O, 1)$, $(O, 2)$, and a constant c , resulting in a function $y(x) = A_{O,1} + A_{O,2} + c$ which approximates f uniformly on $[0, 1]$ with error less than ϵ .*

Proof: Use Lemma 1 to construct the approximating polynomial (18), which we write as $P(x) = P^+(x) + P^-(x)$ where $P^+(x)$ only has non-negative coefficients c_v^+ , while $P^-(x)$ only has non-positive coefficients c_v^- :

$$\begin{aligned} c_v^+ &= \max\{0, c_v\}, \\ c_v^- &= \min\{0, c_v\}. \end{aligned}$$

Notice that

$$-\frac{1}{(1+x)^i} = 1 - \frac{1}{(1+x)^i} - 1 = \sum_{j=1}^i \frac{x}{(1+x)^j} - 1,$$

so that

$$P^-(x) = \sum_{v=1}^m |c_v^-| \sum_{j=1}^v \frac{x}{(1+x)^j} + \sum_{v=1}^m c_v^-. \quad (41)$$

Call $c = c_0 + \sum_{v=1}^m c_v^-$ and for some $d_v \geq 0$ write:

$$P(x) = c + \sum_{v=1}^m \left[c_v^+ \frac{1}{(1+x)^v} + d_v \frac{x}{(1+x)^v} \right]. \quad (42)$$

Let us write $P(x) = c + P^*(x) + P^o(x)$ where both $P^*(x)$ and $P^o(x)$ are polynomials with non-negative coefficients, and

$$\begin{aligned} P^*(x) &= \sum_{v=1}^m c_v^+ \frac{1}{(1+x)^v}, \\ P^o(x) &= \sum_{v=1}^m d_v \frac{x}{(1+x)^v}. \end{aligned}$$

Then by Lemma 5 there are two GNN’s whose output neurons $(O, 1)$, $(O, 2)$ take the values:

$$\begin{aligned} q_{O,1} &= \frac{P^+(x)}{1 + P^+(x)}, \\ q_{O,2} &= \frac{P^o(x)}{1 + P^o(x)}. \end{aligned}$$

Clearly, we can consider that these two GNN’s constitute one network with two output neurons, and we have $y(x) = c + P^*(x) + P^o(x) = P(x)$, completing the proof. **Q.E.D.**

This result can be extended to the CGNN with only one output neuron by applying Lemma 5. However let us first consider the manner in which a positive “clamping constant” $c > 0$ can be added to the average potential of an output neuron of a GNN using the ordinary structure of the network.

Remark 1 (Adding a Positive Clamping Constant). Consider a GNN with an output neuron \hat{q} and an input vector x which realizes the function $\hat{q}(x) = P(x)$. Then there is another GNN with output neuron $Q(x)$ which, for real $c > 0$ realizes the function:

$$Q(x) = \frac{P(x) + c}{1 + P(x) + c} \quad (43)$$

and hence whose average potential is $P(x) + c$. More generally we can exhibit a GNN with output neuron $Q_1(x)$ whose average potential is $bP(x) + c$, for $b > 0, c > 0$.

Proof: The proof is by construction. We first take the output of the neuron of the original network (whose firing rate is denoted $2r$), and feed it into a new neuron with probability 0.5 as an excitatory signal and with probability 0.5 as an inhibitory signal. We set the firing rate of the new neuron to r , and introduce additional exogenous inhibitory and excitatory arrivals to the new neuron, both of rate rc . As a result we have:

$$\begin{aligned} Q(x) &= \frac{rP(x) + rc}{r + rP(x) + rc}, \\ &= \frac{P(x) + c}{1 + P(x) + c}. \end{aligned}$$

As a result, the new neuron's average potential is:

$$\frac{Q(x)}{1 - Q(x)} = P(x) + c.$$

and we have been thus able to obtain a new neuron with an added positive “clamping constant” c with respect to the average potential $P(x)$ of the original neuron. The extension to a neuron with average potential $bp(x) + c$ is straightforward. Let the additional neurons firing rate be $R > 0$ rather than r and take its exogenous excitatory and inhibitory arrival rates to be Rc . We then obtain:

$$\begin{aligned} Q(x) &= \frac{rP(x) + Rc}{R + rP(x) + Rc}, \\ &= \frac{\frac{r}{R}P(x) + c}{1 + \frac{r}{R}P(x) + c}, \end{aligned}$$

so that if we call $b = \frac{r}{R}$ this leads to an average potential of $bP(x) + c$. **Q.E.D.**

Theorem 6. For any continuous function $f : [0, 1] \mapsto R$ and any $\epsilon > 0$, there exists a GNN with one output neuron (O), and a constant c , resulting in a function $y(x) = A_O + c$ which approximates f uniformly on $[0, 1]$ with error less than ϵ .

Proof: Use Lemma 1 to construct the approximating polynomial of (18), which we write as $P(x) = P^+(x) + P^-(x)$ where $P^+(x)$ only has non-negative coefficients c_v^+ , while $P^-(x)$ only has non-positive coefficients c_v^- :

$$\begin{aligned} c_v^+ &= \max\{0, c_v\}, \\ c_v^- &= \min\{0, c_v\}. \end{aligned}$$

Notice that

$$-\frac{1}{(1+x)^i} = 1 - \frac{1}{(1+x)^i} - 1 = \sum_{j=1}^i \frac{x}{(1+x)^j} - 1,$$

so that

$$P^-(x) = \sum_{v=1}^m |c_v^-| \sum_{j=1}^v \frac{x}{(1+x)^j} + \sum_{v=1}^m c_v^- . \quad (44)$$

Call $c = c_0 + \sum_{v=1}^m c_v^-$ and for some $d_v \geq 0$ write:

$$P(x) = c + \sum_{v=1}^m [c_v^+ \frac{1}{(1+x)^v} + d_v \frac{x}{(1+x)^v}] . \quad (45)$$

Let us write $P(x) = c + P^o(x)$ where $P^o(x)$ is a polynomial with non-negative coefficients. Then by Lemma 5 there is a GNN whose output neurons (O) takes the value:

$$q_O = \frac{P^o(x)}{1 + P^o(x)} .$$

Clearly, we can consider that this GNN constitutes one network with only one output neuron, and we have $y(x) = c + P^o(x) = P(x)$, completing the proof. **Q.E.D.**

The next theorem shows that a CGNN with _____ hidden layer is also a universal approximator to continuous functions on $[0, 1]$. We omit the proof, which follows closely the approach and used in the proofs of Theorems

Theorem 7. *For any continuous function $f : [0, 1] \mapsto R$ and any $\epsilon > 0$, there exists a GNN of three layers (only one hidden layer), one output neuron (O), and a constant c called the clamping constant, resulting in a function $y(x) = A_O + c$ which approximates f uniformly on $[0, 1]$ with error less than ϵ .*

4 Approximation of Continuous Functions of s Variables

Now that the process for approximating a one-dimensional continuous functions with the BGNN or the CGNN having a single hidden layer is well understood, consider the case of continuous functions of s variables, i.e. $f : [0, 1]^s \mapsto R$. As a starting point, consider the straightforward extension of Lemma 1 to the case of s -inputs such that there is a polynomial:

$$P(x) = \sum_{m_1 \geq 0, \dots, m_s \geq 0, \sum_{v=1}^s m_v = m} c(m_1, \dots, m_s) \prod_{v=1}^s \frac{1}{(1+x_v)^{m_v}} , \quad (46)$$

with coefficients $c(m_1, \dots, m_s)$ which approximates f uniformly. We now extend Lemma 2 to Lemma 8 and Theorem 8 which are given below.

Lemma 8. *Consider a term of the form*

$$\frac{1}{(1+x_{z1})^{m_{z1}}} \cdots \frac{1}{(1+x_{zK})^{m_{zK}}}$$

for $0 \leq x_{zj} \leq 1$, positive integers $m_{zj} > 0$ and $j = 1, \dots, K$. There exists a feedforward GNN with a single output neuron $(\mu + 1, 1)$ and input $x \in [0, 1]$ such that

$$q_{\mu+1,1} = \frac{1}{(1+x_{z1})^{m_{z1}}} \cdots \frac{1}{(1+x_{zK})^{m_{zK}}} . \quad (47)$$

Proof: Without loss of generality set $m_{z1} \leq m_{z2} \leq \dots \leq m_{zK}$. The resulting network is a cascade connection of a set of networks. The first network is identical in structure to the one of Lemma 2, and has $m_{z1} + 1$ neurons numbered $(1, 1), \dots, (1, m_{z1} + 1)$. Now set:

- $\Lambda(1, 1) = x_{z1}$, $\Lambda(1, 2) = 1/m_{z1}$, and $\Lambda(1, j) = 0$ for $j = 3, \dots, m_{z1} + 1$,
- $\lambda(1, j) = 0$ for all $j = 1, \dots, m_{z1} + 1$, and $d(1, j) = 0$ for $j = 1, \dots, m_{z1}$,
- $\omega^-((1, 1), (1, j)) = 1/m_{z1}$, and $\omega^+((1, 1), (1, j)) = 0$ for $j = 2, \dots, m_{z1} + 1$,
- $r(1, j) = \omega^+((1, j), (1, j+1)) = 1/m_{z1}$ for $j = 2, \dots, m_{z1} + 1$,
- Finally the connection from the first network into the second network is made via $p^+((1, m_{z1} + 1), (2, 2)) = m_{z1}/m_{z2} \leq 1$, with $d(1, m_{z1} + 1) = (1 - m_{z1}/m_{z2})$.

It is easy to see that $q_{1,1} = x_{z1}$, and that

$$q_{1, m_{z1}+1} = \frac{1}{(1 + x_{z1})^{m_{z1}}}. \quad (48)$$

The second network has $m_{z2} + 1$ neurons numbered $(2, 1), \dots, (2, m_{z2} + 1)$. Now set:

- $\Lambda(2, 1) = x_{z2}$ and $\Lambda(2, j) = 0$ for $j = 2, \dots, m_{z2} + 1$,
- $\lambda(2, j) = 0$ for all $j = 1, \dots, m_{z2} + 1$, and $d(2, j) = 0$ for $j = 1, \dots, m_{z2}$,
- $\omega^-((2, 1), (2, j)) = 1/m_{z2}$, and $\omega^+((2, 1), (2, j)) = 0$ for $j = 2, \dots, m_{z2} + 1$,
- $r(2, j) = \omega^+((2, j), (2, j+1)) = 1/m_{z2}$ for $j = 2, \dots, m_{z2} + 1$,
- The connection from the second network into the third network is made via $p^+((2, m_{z2} + 1), (3, 2)) = m_{z2}/m_{z3} \leq 1$, with $d(2, m_{z2} + 1) = (1 - m_{z2}/m_{z3})$.

It is easy to see that $q_{2,1} = x_{z2}$, and that

$$q_{2, m_{z2}+1} = \frac{1}{(1 + x_{z1})^{m_{z1}}} \frac{1}{(1 + x_{z2})^{m_{z2}}}. \quad (49)$$

The remaining construction just pursues the same scheme. **Q.E.D.**

Theorem 8. For any continuous function $f : [0, 1]^s \mapsto R$ and any $\epsilon > 0$, there exists a BGNN with one positive output neuron $(O, +)$, one negative output neuron $(O, -)$, s input variables $X = (x_1, \dots, x_s)$, and the output variable $y(X)$ such that:

$$y(X) = A_{O,+} + A_{O,-}, \quad (50)$$

$$A_{O,+} = \frac{q_{O,+}}{1 - q_{O,+}}, \quad (51)$$

$$A_{O,-} = \frac{-q_{O,-}}{1 - q_{O,-}}, \quad (52)$$

and $\sup_{x \in [0,1]^s} |f(X) - y(X)| < \epsilon$. We will say that the BGNN's output uniformly approximates $f(X)$.

Proof: The proof follows the proof of Theorem 3, using the polynomial of (46). Lemma 7 establishes that the terms of such a polynomial can be realized by a GNN. We then construct two polynomials, one with non-negative coefficients only, and the other with negative coefficients, and show how they are realized with the BGNN. We will not go through the steps of the proof since it is a step by step duplicate of the proof of Theorem 3. **Q.E.D.**

We now extend Lemma 7 to the case of s -inputs.

Lemma 9. *For any continuous function $f : [0, 1]^s \mapsto R$ and for any $\epsilon > 0$, there exists a function of the form*

$$P_s(x) = \sum_{i=1}^r \sum_{0 \leq m_1 \leq 1, \dots, 0 \leq m_s \leq 1} b(m_1, \dots, m_s, i) \prod_{v=1}^s \frac{1}{(a_{v,i} + x_v)^{m_v}}, \quad (53)$$

where $a_{v,i} > 0$, $v = 1, \dots, s$, $i = 1, 2, \dots$, such that $\sup_{x \in [0,1]} |f(x) - P_s(x)| < \epsilon$ is satisfied.

Proof: This is simply an extension of Lemma 7. **Q.E.D.**

As a consequence we can now establish the following general result.

Theorem 9. *For any continuous function $f : [0, 1]^s \mapsto R$ and any $\epsilon > 0$, there exists a BGNN of no more than $s + 2$ layers (s hidden layers), one positive output neuron ($O, +$), one negative output neuron ($O, -$), s input variables $X = (x_1, \dots, x_s)$, and the output variable $y(X)$ determined by (50) such that $\sup_{x \in [0,1]} |f(X) - y(X)| < \epsilon$.*

Proof: The proof is by construction. By Lemma 9, we only need to find an appropriate BGNN of the form as described in Theorem 9 to realize any function of the form (53). We construct a BGNN with s input neurons $(1, 1), \dots, (1, s)$, one positive output neuron ($O, +$), one negative output neuron ($O, -$), and M parallel sub-networks between the input layer and the output layer, where

$$M \equiv \sum_{i=1}^r \sum_{0 \leq m_1 \leq 1, \dots, 0 \leq m_s \leq 1} 1(b(m_1, \dots, m_s, i) \neq 0), \quad (54)$$

$1(X) = 1$ when X is true otherwise $1(X) = 0$. Each sub-network is a cascade connection of no more than s neurons. The output of the last neuron of each sub-network takes the value in proportion to each term in function (53).

Without loss of generality, we consider a term of the form

$$\frac{1}{a_{z1} + x_{z1}} \dots \frac{1}{a_{zK} + x_{zK}} \quad (55)$$

where $a_{z1} \geq a_{z2} \geq \dots \geq a_{zK}$. Now we want to construct a sub-network which has K neurons and of which the last neuron's output takes the value in proportion to the term. Number the K neurons as $(2, 1), (3, 1), \dots, (K + 1, 1)$, and set:

- $\Lambda(1, i) = x_i$, $\lambda(1, i) = 0$, $r(1, i) = 1$, for $i = 1, \dots, s$,
- $\omega^+((1, z1), (2, 1)) = 0$, $\omega^-((1, z1), (2, 1)) = 1/M$,
- $r(2, 1) = a_{z1}/M$, $\Lambda(2, 1) = a_{z1}/M$, $\lambda(2, 1) = 0$.

It is easy to see that

$$q_{2,1} = \frac{a_{z1}}{a_{z1} + x_{z1}}. \quad (56)$$

Then set:

- $p^+((k, 1), (k + 1, 1)) = a_{zK}/a_{z(K-1)}$, for $k = 2, \dots, K$,
- $\omega^+((1, zk), (k + 1, 1)) = 0$, $\omega^-((1, zk), (k + 1, 1)) = 1/M$, for $k = 2, \dots, K$,
- $r(k + 1, 1) = a_{zk}/M$, $\Lambda(k + 1, 1) = 0$, $\lambda(k + 1, 1) = 0$, for $k = 2, \dots, K$.

We will find

$$q_{3,1} = \frac{a_{z1}a_{z2}}{(a_{z1} + x_{z1})(a_{z2} + x_{z2})}, \quad (57)$$

...

$$\frac{a_{z1} \cdots a_{zK}}{(a_{z1} + x_{z1}) \cdots (a_{zK} + x_{zK})} \quad (58)$$

which is in proportion to (55).

Next we connect all the last neurons of the sub-networks to $(O, +)$ or $(O, -)$. The parameter setting follows the steps in the proof of Theorem 4 which connect the neurons in the hidden layer to the output neurons. Since the sub-networks are parallel and each sub-network contains of no more than s neurons, there are totally no more than s hidden layers in this constructed BGNN. Thus, we complete the construction. **Q.E.D.**

We can now obtain Theorems 10 and 11, which generalize Theorems 6 and 7, in a similar manner.

Theorem 10. *For any continuous function $f : [0, 1]^s \mapsto R$ and any $\epsilon > 0$, there exists a GNN with one output neuron (O), and a constant c called the clamping constant, resulting in a function $y(X) = A_O + c$ which approximates f uniformly on $[0, 1]^s$ with error less than ϵ .*

Theorem 11. *For any continuous function $f : [0, 1]^s \mapsto R$ and any $\epsilon > 0$, there exists a GNN of no more than $s + 2$ layers (s hidden layers), one output neuron (O), and a constant c called the clamping constant, resulting in a function $y(X) = A_O + c$ which approximates f uniformly on $[0, 1]^s$ with error less than ϵ .*

5 Conclusions

The approximation of functions by neural networks is central the learning theory of neural networks. It is also a key to many applications of neural networks such as pattern recognition, data compression, time series prediction, adaptive control, etc..

The random neural network introduced and developed in [5, 6, 9, 17] differs significantly from standard connexionist models in that information travels between neurons in this model in the form of random spike trains, and network state is represented by the joint probability distributions that the n neurons in the network are excited. This model has a mathematical structure which is significantly different from that of the connexionist model, the Hopfield model, or the Boltzman machine [3]. Thus the approximation capability of these networks also needs to

be examined in a manner distinct from that of previous models [4]. In particular, the “Gelenbe” random neural network model [5, 6, 9], which are basic to the standard models’ approximation capabilities.

The most basic requirement for a neural network model is that it should be a universal function approximator: i.e. to any continuous function f on a compact set, we should be able to find a specific network which implements a mapping close enough, in some precise sense to f , to a given degree of accuracy. Furthermore, among all networks which satisfy this property, we may wish to choose the one with the “smallest” size or the most simple structure.

In [16] we showed that the BGNN and the CGNN, two simple extensions of the basic “Gelenbe” Random Neural Network model, are universal approximators of continuous real-valued functions of s real variables. However we had not previously established the specific “size” constraints for the approximating networks.

In this paper we limit the networks to being feedforward and consider the case where the number of hidden layers does not exceed the number of input variables. With these constraints we show that the feedforward CGNN and the BGNN with s hidden layers (total of $s + 2$ layers) can uniformly approximate continuous functions of s variables. We also extend a theorem in [16] on universal approximation using the CGNN with two output neurons, to the CGNN with only one output neuron.

The theoretical results we report in this paper are not only needed to justify the empirically observed success obtained in a variety of applications of the “Gelenbe” random neural network [7, 8, 10, 11, 12, 13, 14, 15], and to support further applied work in spiked stochastic neural network models. We believe that these results will lead to new developments in the design of network structures which are adapted to certain specific learning or approximation tasks.

REFERENCES

- [1] W.E. Feller, *An Introduction to Probability Theory and Its Applications*, Vol. I (3rd Edition) and Vol. II, Wiley, 1968, 1966.
- [2] M. J. D. Powell, *Approximation Theory and Methods*. Cambridge University Press, 1981.
- [3] J.L. McClelland, D.E. Rumelhart, D.E., et al. *Parallel Distributed Processing*, Vols. I and II, MIT Press, 1986.
- [4] K. Funahashi, “On the approximate realization of continuous mapping by neural network,” *Neural Networks*, vol. 2, pp. 183-192, 1989.
- [5] E. Gelenbe, “Random neural networks with negative and positive signals and product form solution,” *Neural Computation*, vol. 1, no. 4, pp. 502-511, 1989.
- [6] E. Gelenbe, “Stability of the random neural network model,” *Neural Computation*, vol. 2, no. 2, pp. 239-247, 1990.
- [7] E. Gelenbe, A. Stafylopatis, and A. Likas, “Associative memory operation of the random network model,” in *Proc. Int. Conf. Artificial Neural Networks*, Helsinki, pp. 307-312, 1991.

- [8] E. Gelenbe, F. Batty, "Minimum cost graph covering with the random neural network," *Computer Science and Operations Research*, O. Balci (ed.), New York, Pergamon, pp. 139-147, 1992.
- [9] E. Gelenbe, "Learning in the recurrent random neural network," *Neural Computation*, vol. 5, no. 1, pp. 154-164, 1993.
- [10] E. Gelenbe, V. Koubi, F. Pekergin, "Dynamical random neural network approach to the traveling salesman problem," *Proc. IEEE Symp. Syst., Man, Cybern.*, pp. 630-635, 1993.
- [11] A. Ghanwani, "A qualitative comparison of neural network models applied to the vertex covering problem," *Elektrik*, vol. 1, pp. 1-14, 1994.
- [12] E. Gelenbe, C. Cramer, M. Sungur, P. Gelenbe "Traffic and video quality in adaptive neural compression", *Multimedia Systems*, Vol. 4, pp. 357-369, 1996.
- [13] C. Cramer, E. Gelenbe, H. Bakircioglu "Low bit rate video compression with neural networks and temporal subsampling," *Proceedings of the IEEE*, Vol. 84, No. 10, pp. 1529-1543, October 1996.
- [14] E. Gelenbe, T. Feng, K.R.R. Krishnan "Neural network methods for volumetric magnetic resonance imaging of the human brain," *Proceedings of the IEEE*, Vol. 84, No. 10, pp. 1488-1496, October 1996.
- [15] E. Gelenbe, A. Ghanwani, V. Srinivasan, "Improved neural heuristics for multicast routing," *IEEE J. Selected Areas in Communications*, vol. 15, no. 2, pp. 147-155, 1997.
- [16] E. Gelenbe, Z. H. Mao, and Y. D. Li, "Function approximation with the random neural network," *IEEE Trans. Neural Networks*, vol. 10, no. 1, January 1999.
- [17] E. Gelenbe, J.M. Fourneau "Random neural networks with multiple classes of signals," *Neural Computation*, vol. 11, pp. 721-731, 1999.

Appendix:
Proof of Technical Lemmas

Proof of Lemma 1: This is a direct consequence of Weierstrass' Theorem (see [2], p. 61) which states that for any continuous function $h : [a, b] \mapsto \mathbb{R}$, and some $\epsilon > 0$, there exists a polynomial $P(u)$ such that $\sup_{u \in [a, b]} |h(u) - P(u)| < \epsilon$. Now let $u = 1/(1+x)$, $u \in [1/2, 1]$ and select $x = (1-u)/u$ with $h(u) = f(\frac{1-u}{u}) = f(x)$. If $f(x)$ is continuous, then so is $h(u)$ so that there exists an algebraic polynomial of the form

$$P(u) = c_0 + c_1u + \dots + c_mu^m, \quad 1/2 \leq u \leq 1, \quad (59)$$

such that $\sup_{u \in [1/2, 1]} |h(u) - P(u)| < \epsilon$. Therefore $P(x)$ is given by (18), and $\sup_{x \in [0, 1]} |f(x) - P(x)| < \epsilon$. **Q.E.D.**

Proof of Lemma 2: Construct a feedforward GNN with $v+1$ neurons numbered $(1, 1), \dots, (v+1, 1)$. Now set:

- $\Lambda(1, 1) = x$, $\Lambda(2, 1) = 1/v$, and $\Lambda(j, 1) = 0$ for $j = 3, \dots, v+1$,
- $\lambda(j, 1) = 0$ for all $j = 1, \dots, v+1$, and $d(j, 1) = 0$ for $j = 1, \dots, v$,
- $\omega^-((1, 1), (j, 1)) = 1/v$, and $\omega^+((1, 1), (j, 1)) = 0$ for $j = 2, \dots, v+1$,
- $r(j, 1) = \omega^+((j, 1), (j+1, 1)) = 1/v$ for $j = 2, \dots, v$,
- Finally $d(v+1, 1) = 1$.

It is easy to see that $q_{1,1} = x$, and that

$$q_{j+1,1} = \left(\frac{1}{1+x}\right)^j, \quad (60)$$

for $j = 1, \dots, v$ so the Lemma follows. **Q.E.D.**

The next result exhibits a simple a construction process for algebraic expressions using the feedforward GNN.

Remark. *If there exists a feedforward GNN with a single output neuron $(L, 1)$, and a function $g : [0, 1] \mapsto [0, 1]$ such that:*

$$q_{L,1} = g(x), \quad (61)$$

then there exists an $L+1$ layer feedfourward GNN with a single output neuron (Q) such that:

$$q_O = \frac{g(x)}{1+g(x)}. \quad (62)$$

Proof: The simple proof is by construction. We simply add an additional neuron (Q) the original GNN, and leave all connections in the original GNN unchanged except for the output connections of the neuron $(L, 1)$. Let the firing rate of neuron $(l, 1)$ be $r(L, 1)$. Then:

- $(L, 1)$ will now be connected to the new neuron $(L + 1, 1)$ by $\omega^+((L, 1), Q) = r(L, 1)/2$,
 $\omega^-((L, 1), Q) = r(L, 1)/2$,
- $r(Q) = r(L, 1)/2$.

This completes the proof. **Q.E.D.**

Proof of Lemma 3: The proof is by construction. Let C_{MAX} be the largest of the coefficients in $P^+(x)$ and write $P^*(x) = P^+(x)/C_{MAX}$. Let $c_j^* = c_j/C_{MAX} \leq 1$ so that now each term $c_j^* \frac{1}{(1+x)^j}$ in $P^*(x)$ is no greater than 1, $j = 1, \dots, m$. We now take m networks of the form of Lemma 2 with $r(j, 1) = 1$, $j = 1, \dots, m$ and output values

$$q_{j,1} = \left(\frac{1}{1+x}\right)^j, \quad (63)$$

and connect them to the new output neuron (O) by setting the probabilities $p^+((j, 1), O) = c_j^*/2$, $p^-((j, 1), O) = c_j^*/2$. Furthermore we set an external positive and negative signal arrival rate $\Lambda(O) = \lambda(O) = c_0^*/2$ and $r(O) = 1/(2C_{MAX})$ for the output neuron. We now have:

$$q_O = \frac{\frac{P^*(x)}{2}}{\frac{1}{2C_{MAX}} + \frac{P^*(x)}{2}}. \quad (64)$$

We now multiply the numerator and the denominator on the right hand side of the above expression by $2C_{MAX}$ to obtain

$$q_O = \frac{P^+(x)}{1 + P^+(x)}. \quad (65)$$

so that which completes the proof of the Lemma. **Q.E.D.**

Proof of Lemma 4: The proof is very similar to that of Lemma 2. Construct a feedforward GNN with $v + 1$ neurons numbered $(1, 1), \dots, (v + 1, 1)$. Now set:

- $\Lambda(1, 1) = x$, and $\Lambda(j, 1) = 0$ for $j = 2, \dots, v + 1$,
- $\lambda(j, 1) = 0$ for all $j = 1, \dots, v + 1$, and $d(j, 1) = 0$ for $j = 1, \dots, v$,
- $\omega^+((1, 1), (2, 1)) = 1/(v + 1)$, $\omega^-((1, 1), (j, 1)) = 1/(v + 1)$ for $j = 2, \dots, v + 1$, and $\omega^+((1, 1), (j, 1)) = 0$ for $j = 3, \dots, v + 1$,
- $r(j, 1) = \omega^+((j, 1), (j + 1, 1)) = 1/(v + 1)$ for $j = 2, \dots, v$,
- Finally $d(v + 1, 1) = 1$.

It is easy to see that $q_{1,1} = x$, and that

$$q_{j+1,1} = \frac{x}{(1+x)^j}, \quad (66)$$

for $j = 1, \dots, v$ so the Lemma follows. **Q.E.D.**

Finally, we state without proof another lemma, very similar to Lemma 4, but which uses terms of the form $x/(1+x)^v$ to construct polynomials. Its proof uses Lemma 5, and follows exactly the same lines as Lemma 4.

Lemma 6. *Let $P^o(x)$ be a polynomial of the form*

$$P^o(x) = c_0 + c_1 \frac{x}{1+x} + \dots + c_m \frac{x}{(1+x)^m}, \quad 0 \leq x \leq 1, \quad (67)$$

with non-negative coefficients, i.e. $c_v \geq 0$, $v = 1, \dots, m$. Then there exists a feedforward GNN with a single output neuron $(O, +)$ such that:

$$q_O = \frac{P^o(x)}{1 + P^o(x)}, \quad (68)$$

so that the average potential of the output neuron is $A_O = P^o(x)$.

Performance Evaluation of Priority based Schedulers in the Internet

LASSAAD ESSAFI¹ and GUNTER BOLCH¹

¹Institute of Computer Science

University of Erlangen-Nuremberg

Martensstrasse 3, D-91058 Erlangen, Germany

ldessafi@aol.com, bolch@informatik.uni-erlangen.de

31-Mar-2003

Abstract

This paper shows how the delay of jobs/packets using priority based scheduling mechanisms can be computed analytically. Static priorities and different types of time dependent priorities are considered. This paper also shows how these results are used in modern router design. Results for mono-processor and multi-processor architectures are given.

Keywords: priority queuing, multi-processor systems, quality of service

1 Introduction

In a router architecture queuing occurs when packets are received by a device's interface processor (input queue), and queuing may also occur prior to transmitting the packets to another interface (output queue) on the same device. A basic router is a collection of input processes that assemble packets as they are received, checking the integrity of the basic packet framing, one or more processors that determine the destination interface to which the packet should be passed and output processors that frame and transmit the packets on their next hop.

Priority scheduling represents a class of scheduling disciplines which can be used to provide differentiated services in the Internet. In addition to strict priority scheduling, already implemented in several router architectures, recent research on proportional differentiated services has shown that the waiting time priority scheduler is a promising mechanism for approximating the proportional delay differentiation model [1].

The aim of this paper is to give an overview of different priority based scheduling mechanisms and investigate their properties analytically. Two classes of priority based disciplines are discussed: static priorities in Section 2 and time dependent priorities with different variants in Section 3. We finally conclude in Section 4.

2 Static Priorities

One of the first queuing variations to be widely implemented was priority queuing. Here a fairly general model based on M/G/1 is used [9], but can in some cases be approximatively extended to the more general case of G/G/m. The results hold only in case of stationarity. In this queuing model we assume that an arriving packet belongs to a priority class r ($r = 1, 2, \dots, R$). The priority of a packet is constant during its whole sojourn time in the router, that is why we refer to this class of priorities as static priorities. The next packet to be sent is the packet with the highest priority r . Within a priority class the queuing discipline is FCFS (First-Come-First-Served).

The mean waiting time \bar{W}_r of an arriving packet C_r of the priority class r has three components [5, 6]:

1. The mean remaining service time \bar{W}_0 of the packet being served (if any).
2. The mean service time of the packets, found in the queue by the tagged packet on arrival, and that are served before it. These are the packets in the queue of the same and higher priority as the tagged packet.
3. Mean service time of packets that arrive at the system while the tagged packet is in the queue and are served before him. These are packets with higher priority than the tagged packet.

Note that we consider only the case where a packet being served is *not preempted* by an arriving packet with higher priority. Preemption is not considered in this paper because it is less relevant in the context of packet queuing.

We define:

\bar{N}_{ir} : Mean number of packets of class i found in the queue by the tagged packet C_r (with priority r) and being served before it,

\bar{M}_{ir} : Mean number of packets of class i which arrive during the waiting time of the tagged packet and being served before it.

Then the mean waiting time of class r packets in an M/G/1 system can be written as the sum of three components:

$$\bar{W}_r = \bar{W}_0 + \sum_{i=1}^R \bar{N}_{ir} \cdot \frac{1}{\mu_i} + \sum_{i=1}^R \bar{M}_{ir} \cdot \frac{1}{\mu_i}. \quad (1)$$

where μ_i is the service rate of the packets of class i ¹. For a multi-processor system with m processors M/G/m ($m > 1$):

$$\bar{W}_r = \bar{W}_0 + \sum_{i=1}^R \frac{\bar{N}_{ir}}{m} \frac{1}{\mu_i} + \sum_{i=1}^R \frac{\bar{M}_{ir}}{m} \cdot \frac{1}{\mu_i}, \quad (2)$$

where \bar{N}_{ir} and \bar{M}_{ir} are given by:

$$\begin{aligned} \bar{N}_{ir} &= 0 & i < r, \\ \bar{M}_{ir} &= 0 & i \leq r, \end{aligned} \quad (3)$$

and, with Little's theorem resp. the formula for the mean number of jobs of class i which arrive during the mean waiting time \bar{W}_r :

$$\begin{aligned} \bar{N}_{ir} &= \lambda_i \bar{W}_i & i \geq r, \\ \bar{M}_{ir} &= \lambda_i \bar{W}_r & i > r, \end{aligned} \quad (4)$$

where λ_i denotes the arrival rate of class- i packets. Considering the utilization factor ρ_r in class r as

$$\rho_r = \frac{\lambda_r}{\mu_r} \quad (5)$$

equations (1) and (2) can be solved to obtain:

$$\bar{W}_r = \frac{\bar{W}_0}{(1 - \sigma_r)(1 - \sigma_{r+1})}, \quad (6)$$

where:

$$\sigma_r = \sum_{i=r}^R \rho_i. \quad (7)$$

Obviously the sum of utilization factors in all classes has to fulfill the condition (stability condition):

$$\sum_{i=1}^R \rho_i < 1, \quad (8)$$

otherwise the system is unstable and the queue lengths become infinite.

Mean Remaining Service Time \bar{W}_0 : The mean remaining service time \bar{W}_0 for poisson arrival [8] is given by:

$$\bar{W}_0 = P(\text{server is busy}) \cdot \bar{R} + P(\text{server is idle}) \cdot 0, \quad (9)$$

with the main remaining service time \bar{R} of a busy server (the remaining service time of an idle server is obviously zero). When a packet arrives, the packet in

service needs \bar{R} time units on the average to be finished. This quantity is also called the mean residual life [6] and is given by:²

$$\bar{R} = \frac{\bar{T}_B^2}{2\bar{T}_B} = \frac{\bar{T}_B}{2}(1 + c_B^2). \quad (10)$$

For an M/M/1 system ($c_B^2 = 1$), we obtain:

$$\bar{R}_{M/M/1} = \bar{T}_B = \frac{1}{\mu},$$

which is related to the memoryless property of the exponential distribution. For multi-processor system with m processors we refer to the approximation in [6]:

$$\bar{R} \approx \frac{\bar{T}_B^2}{2m\bar{T}_B} \quad (11)$$

The probability that the link server is busy is for the mono-processor case given by the utilization ρ . For a multi-processor server with m processors, the probability P_m that all m processors are busy can be calculated as follows: Let p_k be the probability that k packets are being actually in the system

$$p_k = \begin{cases} p_0 \frac{(m\rho)^k}{k!} & \text{if } k \leq m \\ p_0 \frac{m^m \rho^k}{m!} & \text{if } k \geq m \end{cases} \quad (12)$$

$$p_0 = \left(\left(\sum_{k=0}^{m-1} \frac{(m\rho)^k}{k!} \right) + \frac{(m\rho)^m}{m!(1-\rho)} \right)^{-1} \quad (13)$$

For a system with m processors:

$$P_m = \sum_{k=0}^{\infty} p_k = \frac{(m\rho)^m}{m!(1-\rho)} p_0 \quad (14)$$

The exact solution is only valid for exponentially distributed service times, e.g. M/M/m-systems. For M/G/m-systems several approximations have been given in [5]:

1. $P_m = 1 - e^{-m\rho} \sum_{k=0}^{m-1} \frac{(m\rho)^k}{k!}$ for a utilization $\rho \leq 0.3$
2. $P_m = \rho^{m+1}/2$ for $0.3 \leq \rho \leq 0.7$
3. $P_m = \frac{\rho + \rho^m}{2}$ for a utilization higher than 0.7
4. $P_m = P_{m(M/M/m)}$ for all utilization values

We conclude this section by giving the mean remaining service time for different queuing systems [6]³:

$$\bar{W}_{0,M/M/1} = \sum_{i=1}^R \rho_i \frac{1}{\mu_i} \quad (15)$$

² \bar{T}_B is the mean service time and c_B^2 is the coefficient of variation

³please note that the presented results and the results in the following sections have been validated using simulations in [10, 11]

¹throughout this paper we use the same notation as in Bolch et.al. [6]

$$\bar{W}_{0,M/G/1} = \sum_{i=1}^R \rho_i \cdot \frac{1 + c_{B_i}^2}{2\mu_i} \quad (16)$$

$$\bar{W}_{0,GI/G/1,AC} \approx \sum_{i=1}^R \rho_i \cdot \frac{c_{A_i}^2 + c_{B_i}^2}{2\mu_i} \quad (17)$$

$$\bar{W}_{0,GI/G/1,KLB} \approx \sum_{i=1}^R \rho_i \cdot \frac{c_{A_i}^2 + c_{B_i}^2}{2\mu_i} \cdot G_{KLB} \quad (18)$$

$$\bar{W}_{0,GI/G/1,KUL} \approx \sum_{i=1}^R \rho_i \cdot \frac{f(c_{A_i}, c_{B_i}, \rho_i) + c_{B_i}^2}{2\mu_i} \quad (19)$$

$$\bar{W}_{0,M/M/m} = \frac{P_m}{m\rho} \sum_{i=1}^R \rho_i \cdot \frac{1}{\mu_i} \quad (21)$$

$$\bar{W}_{0,M/G/m} \approx \frac{P_m}{2m\rho} \cdot \sum_{i=1}^R \rho_i \cdot \frac{1 + c_{B_i}^2}{\mu_i} \quad (22)$$

$$\bar{W}_{0,GI/G/m,AC} \approx \frac{P_m}{2m\rho} \cdot \sum_{i=1}^R \rho_i \cdot \frac{c_{A_i}^2 + c_{B_i}^2}{\mu_i} \quad (23)$$

$$\bar{W}_{0,GI/G/m,KLB} \approx \frac{P_m}{2m\rho} \sum_{i=1}^R \rho_i \cdot \frac{c_{A_i}^2 + c_{B_i}^2}{\mu_i} G_{KLB} \quad (24)$$

$$\bar{W}_{0,GI/G/m,KUL} \approx \frac{P_m}{2m\rho} \sum_{i=1}^R \rho_i \cdot \frac{f(c_{A_i}, c_{B_i}, \rho_i) + c_{B_i}^2}{\mu_i} \quad (27)$$

For $f(c_{A_i}, c_{B_i}, \rho_i)$,

$$f(c_A, c_B, \rho) = \begin{cases} 1, c_A \in \{0, 1\}, \\ + \left[\rho(14.1c_A - 5.9) + (-13.7c_A + 4.1) \right] c_B^2 \\ + \left[\rho(-59.7c_A + 21.1) + (54.9c_A - 16.3) \right] c_B \\ + \left[\rho(c_A - 4.5) + (-1.5c_A + 6.55) \right], \\ 0 \leq c_A \leq 1, \\ -0.75\rho + 2.775, c_A > 1, \end{cases} \quad (28)$$

for $G_{KLB,GI/G/1}$

$$G_{KLB} = \begin{cases} \exp\left(-\frac{2}{3} \cdot \frac{1-\rho}{\rho} \cdot \frac{(1-c_A^2)^2}{c_A^2 + c_B^2}\right), & 0 \leq c_A \leq 1, \\ \exp\left(-(1-\rho) \frac{c_A^2 - 1}{c_A^2 + 4c_B^2}\right), & c_A > 1. \end{cases} \quad (29)$$

and for $G_{KLB,GI/G/m}$

$$G_{KLB} = \begin{cases} \exp\left(-\frac{2}{3} \frac{1-\rho}{P_m} \frac{(1-c_A^2)^2}{c_A^2 + c_B^2}\right), & 0 \leq c_A \leq 1, \\ \exp\left(-(1-\rho) \frac{c_A^2 - 1}{c_A^2 + 4c_B^2}\right), & c_A > 1, \end{cases} \quad (30)$$

3 Time Dependent Priorities

Static priorities are simple to implement in software and hardware because, to make a scheduling decision, the scheduler needs only to determine the highest priority nonempty queue. On the other hand, a

static priority scheme allows a misbehaving connection at highest priority to increase the delay and decrease the available bandwidth of connections at all lower priority levels. This leads in extreme cases to starvation of lower priority classes.

In many cases it is advantageous for a packet priority to increase with the time. This possibility can be considered if we use a priority function:

$$q_r(t) = \text{Priority of class } r \text{ at time } t.$$

Such systems are more flexible but need more expense for the administration. In this section, we investigate different types of time dependent priorities.

3.1 Class Dependent Increasing Rate

We refer to the same queuing model and assign each priority class a parameter b_r , which can be interpreted according to the the priority function

$$q_r(t) = (t - t_0)b_r \quad (31)$$

as the increasing rate (slope) of the priority in the class r , where $0 \leq b_1 \leq b_2 \leq \dots \leq b_R$. This means that the priority of a higher class increases faster than the priority of a lower class. A packet enters the system at time t_0 and then increases its priority at the rate b_r .

In order to determine the mean waiting time of a packet C_r which arrives to the system at time t_0 , we follow the same approach as for static priorities (Eqn. 1). We have to determine the mean number of packets \bar{N}_{ir} of class i found in the queues by the tagged packet (belonging to the priority class r) upon its arrival and being served before it, and the mean number of packets \bar{M}_{ir} of class i which arrive during the waiting time of the tagged packet and being served before it. For the mean remaining service times we can use the same formulae as for static priorities.

The set \bar{N}_{ir} : We first consider the packets of lower priority classes ($i < r$), which arrive to the queuing system before the tagged packet C_r (before t_0) and which are served before it. These packets are characterized by the following (see Figure 1):

- these packets arrived to the system at some point $-t_1$
- waiting time: $w_i(t_1)$ with $t_1 < w_i(t_1) < t_1 + t_2$
- have at time t_2 the same priority as C_r , which means $b_r t_2 = (t_1 + t_2)b_i$.

Now we compute $t_1 + t_2$, which is the period where the lower priority packets in N_{ir} are served before C_r :

$$t_1 + t_2 = \frac{b_r t_1}{b_r - b_i}$$

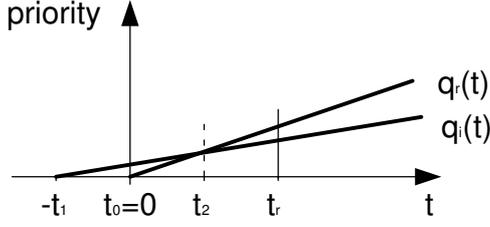


Figure 1: Priority functions with constant slopes (for the sets \overline{N}_{ir})

\overline{N}_{ir} can be written as:

$$\int_0^\infty \lambda_i P\{t < w_i(t) \leq \frac{b_r}{b_r - b_i} t_1\} dt \quad (32)$$

Eqn. (32) can be simplified using the substitution:

$$y = \frac{b_r}{b_r - b_i} t$$

and the fact

$$W_i = E[w_i] = \int_0^\infty 1 - P(w_i \leq x) dx$$

We finally get:

$$\begin{aligned} \overline{N}_{ir} &= \lambda_i \int_0^\infty 1 - P(w_i \leq t) dt \\ -\lambda_i \int_0^\infty 1 - P(w_i \leq y) dy \left(1 - \frac{b_i}{b_r}\right) \end{aligned} \quad (33)$$

hence

$$\overline{N}_{ir} = \lambda_i W_i \frac{b_i}{b_r} \text{ for all } i < r \quad (34)$$

\overline{N}_{ir} is according to Little's theorem for all $i \geq r$

$$\overline{N}_{ir} = \lambda_i W_i \quad (35)$$

The set \overline{M}_{ir} : Considering the tagged packet C_r (arrived at t_0), it is obvious that

$$\overline{M}_{ir} = 0 \text{ for all } i \leq r \quad (36)$$

because no packet of these lower priority classes will be served before C_r . For classes with higher priorities we have to consider all packets which arrive to the system after C_r , but which are served before it. These are according to Figure 2 all packets which arrive in the interval $[0, T_i)$. The crucial time T_i is determined by:

$$b_r W_r = b_i (W_r - T_i)$$

hence:

$$T_i = W_r \left(1 - \frac{b_r}{b_i}\right)$$

and:

$$\overline{M}_{ir} = \lambda_i T_i = \lambda_i \overline{W}_r \left(1 - \frac{b_r}{b_i}\right) \text{ for all } i > r \quad (37)$$

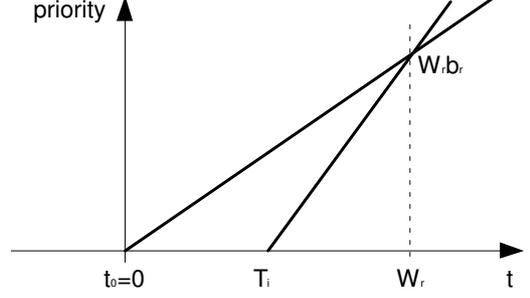


Figure 2: Priority functions with constant slope (for the sets \overline{M}_{ir})

The mean waiting time \overline{W}_r : After substituting the previous results (34), (35), (36) and (37) in (1) we get:

$$\overline{W}_r = \frac{\overline{W}_0 + \sum_{i=1}^{r-1} \rho_i \overline{W}_i \frac{b_i}{b_r} + \sum_{i=r}^R \rho_i \overline{W}_i}{1 - \sum_{i=r+1}^R \rho_i \left(1 - \frac{b_r}{b_i}\right)} \quad (38)$$

We use the conservation law [6]:

$$\sum_{i=r}^R \rho_i \overline{W}_i = \rho \overline{W}_{FIFO} \quad (39)$$

and get a recursive formula for the mean waiting times:

$$\overline{W}_r = \frac{\overline{W}_{FIFO} - \sum_{i=1}^{r-1} \rho_i \overline{W}_i \left(1 - \frac{b_i}{b_r}\right)}{1 - \sum_{i=r+1}^R \rho_i \left(1 - \frac{b_r}{b_i}\right)} \quad (40)$$

Relationship between \overline{W}_0 and \overline{W}_{FIFO} : In Section 2 the mean remaining service time \overline{W}_0 for a variety of queuing systems were given. The relationship between \overline{W}_0 and \overline{W}_{FIFO} is given by:

$$\overline{W}_{FIFO} = \frac{\overline{W}_0}{1 - \rho} \quad (41)$$

Proof:

The mean waiting time of packet in a FIFO system has two components:

1. the mean remaining service time \overline{W}_0 of the packet in service (if any),
2. the sum of the mean service times of the packets in the queue.

This sum can be written as:

$$\overline{W} = \overline{W}_0 + \overline{Q} \cdot \overline{T}_B \quad (42)$$

According to Little's theorem, the mean number of packets in the queue is:

$$\overline{Q} = \lambda \cdot \overline{W}$$

From Eqn. (42) we obtain:

$$\bar{W} = \bar{W}_0 + \lambda \cdot \bar{W} \cdot \bar{T}_B = \bar{W}_0 + \rho \cdot \bar{W}$$

and finally Eqn. (41). q.e.d.

3.2 Variants of the Priority Function

The first variant of the priority function in Eqn. 31 consists in assigning an exponent r_s (i.e. 2) to the slope b_r . The resulting priority function is then:

$$q_r^{r_s}(t) = (t - t_0)b_r^{r_s} \quad (43)$$

The mean waiting time of a packet of the class r is:

$$\bar{W}_r = \frac{\bar{W}_{FIFO} - \sum_{i=1}^{r-1} \rho_i \bar{W}_i \left(1 - \left(\frac{b_i}{b_r}\right)^{r_s}\right)}{1 - \sum_{i=r+1}^R \rho_i \left(1 - \left(\frac{b_r}{b_i}\right)^{r_s}\right)} \quad (44)$$

The use of the exponent r_s leads to a better separation of the different priority classes and it is possible to theoretically cover the whole spectrum of queuing mechanisms: from a strict differentiation of the priority classes as done with static priorities to no differentiation as it is the case with FIFO.

The second variant is the priority function:

$$q_r^n(t) = (t - t_0)^n b_r \quad (45)$$

we get for the mean waiting time:

$$\bar{W}_r = \frac{\bar{W}_{FIFO} - \sum_{i=1}^{r-1} \rho_i \bar{W}_i \left(1 - \left(\frac{b_i}{b_r}\right)^{1/n}\right)}{1 - \sum_{i=r+1}^R \rho_i \left(1 - \left(\frac{b_r}{b_i}\right)^{1/n}\right)} \quad (46)$$

The exponent n is used to weight the waiting time in the system, which also leads to a fine differentiation of the classes. For the limits $n \rightarrow \infty$ and $n \rightarrow 0$, the system tend to static priorities and FIFO respectively. The combination of r_s and n , which does not cause any mathematical problems, allows more variation possibilities in the specified spectrum.

3.3 Class Dependent Starting Priorities

A possibility to reduce the waiting time of certain packets is to assign to each packet a class dependent starting priority r_r . With a slope 1 for the priority functions of all classes and with

$$0 \leq r_1 \leq r_2 \leq \dots \leq r_R$$

we get the following form for the priority functions:

$$q_r(t) = r_r + t - t_0 \quad (47)$$

In this case again, the mean waiting time \bar{W}_r is dependent on the mean remaining service time \bar{W}_0 and the sets \bar{M}_{ip} and \bar{N}_{ip} , which have to be determined.

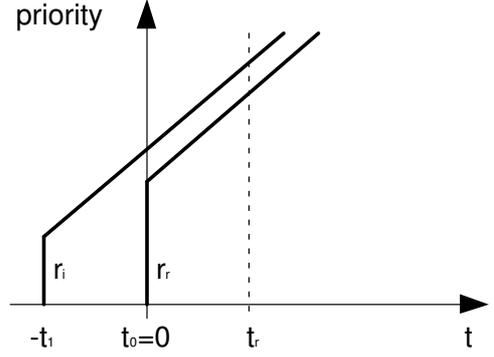


Figure 3: Determination of \bar{N}_{ir} for class dependent starting priorities

The set \bar{N}_{ir} : A packet C_r , which arrives to the system at time t_0 with the starting priority r_r cannot be served before the packets already buffered in the queues of the same or higher classes, that's why:

$$\bar{N}_{ir} = \lambda_i \bar{W}_i \text{ for all } i \geq r \quad (48)$$

For lower priority classes, the packets (indexed with i) already buffered in the queues at time t_0 and which will be served before C_r are characterized by (see Figure 3):

- arrival time: $-t_1$
- starting priority: r_i
- priority at time t_0 : $q_i(t_0) = r_i + t_1$
- waiting time: $w_i(t_1)$ with $t_1 < w_i(t_1) < \infty$
- $q_i(t_0) \geq r_r$, because they are served before C_r

These packets will get served before C_r , that's why:

$$t_1 \geq r_r - r_i$$

Hence for ($i < r$) we get:

$$\bar{N}_{ir} = \int_{r_r - r_i}^{\infty} \lambda_i P\{t < w_i(t) \leq \infty\} dt \quad (49)$$

After substitution and using:

$$W_i = E[w_i] = \int_0^{\infty} 1 - P(w_i \leq t) dt \quad (50)$$

we finally get:

$$\bar{N}_{ir} = \lambda_i W_i - \lambda_i \int_0^{r_r - r_i} P(w_i > t) dt \text{ for all } (i < r) \quad (51)$$

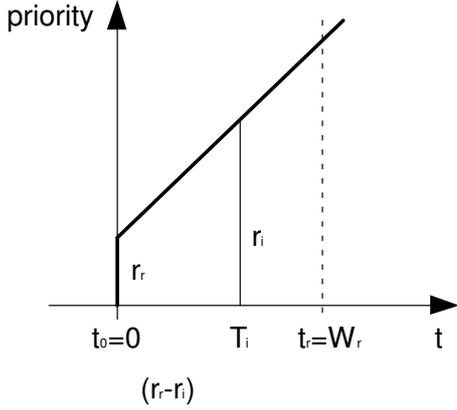


Figure 4: Determination of \bar{M}_{ip} for class dependent starting priorities

The set \bar{M}_{ir} : Considering the tagged packet C_r (arrived at t_0), it is obvious that

$$\bar{M}_{ir} = 0 \text{ for all } i \leq r \quad (52)$$

because no packet of these lower priority classes will be served before C_r .

For classes with higher priorities we have to consider all packets which arrive to the system after C_r , but which are served before it. These are according to Figure 4 all packets which arrive in the interval $[0, T_i]$, because:

$$q_i(t_r) \geq q_r(t_r) = r_r + t_r$$

We use

$$r_i + W_r - T_i = r_r + W_r$$

to calculate T_i and get:

$$T_i = r_i - r_r$$

Hence:

$$\bar{M}_{ir} = \lambda_i \int_0^{r_i - r_r} P(w_r > t) dt \text{ for all } (i > r) \quad (53)$$

Substituting (48), (48) and (53) in (1) and using the conservation law (39) gives:

$$\begin{aligned} \bar{W}_r &= \bar{W}_{FIFO} - \sum_{i=1}^{r-1} \rho_i \int_0^{r_r - r_i} P(w_i > t) dt + \\ &\sum_{i=r+1}^R \rho_i \int_0^{r_i - r_r} P(w_r > t) dt \end{aligned} \quad (54)$$

To determine the waiting probabilities $P(w_k < t)$ ($k = i, r$), which cannot be calculated exactly, we refer here to two approximations proposed in [7] for M/G/1-systems:

Approximation 1: is a heavy traffic approximation of the mean waiting time is given by:

$$\bar{W}_r = \bar{W}_{FIFO} - P_m \sum_{i=1}^R \rho_i (r_r - r_i) \quad \rho \rightarrow 1 \quad (55)$$

Approximation 2: This approximation of the mean waiting time is valid for $0 \leq \rho < 1$

$$\begin{aligned} \bar{W}_r (1 - \sum_{i=r+1}^R \rho_i (1 - e^{P_m(r_r - r_i)/\bar{W}_r})) = \\ \bar{W}_{FIFO} - \sum_{i=1}^{r-1} \rho_i \bar{W}_i (1 - e^{P_m(r_i - r_r)/\bar{W}_i}) \end{aligned} \quad (56)$$

Given \bar{W}_1 all other mean waiting times can be determined recursively, where for each i the solution of the equation (56) has to be computed numerically.

3.4 Starting Priorities r_p with Independent Increasing Rates b_q

We extend here the considered system, where not only a starting priority r_p for each class is defined, but also a class dependent increasing rate. If both parameters are dependent on each other, that means a class with a high starting priority also has a relatively high increasing rate, we then define the priority functions as:

$$q_r(t) = r_r + (t - t_0)b_r \quad (57)$$

These functions don't have any important differences if compared to the priority functions (31) and (47) [5]. If the two parameters are independent of each other, so we will characterize each class with the two parameters: p for the starting priority and q for the increasing rate, with

$$0 \leq r_1 \leq r_2 \leq \dots \leq r_P \text{ and}$$

$$0 \leq b_1 \leq b_2 \leq \dots \leq b_Q$$

A combination of a high starting priority and a low increasing rate (and vice versa) is here possible. This system is the most general one and covers all cases between FIFO- and static priority systems. The priority function is defined by:

$$q_{pq} = r_p + (t - t_0)b_q \quad (58)$$

All parameters like C_p, ρ_p and λ_p have now to extended to $C_{pq}, \rho_{pq}, \lambda_{pq}$ etc. Furthermore we define $\bar{N}_{ij,pq}$ and $\bar{M}_{ij,pq}$ as:

$\bar{N}_{ij,pq}$ mean number of packets with starting priority r_i and increasing rate b_j , which are already buffered in the queues and are will be served before the packet C_{pq} , with starting priority r_p and increasing rate b_q ,

$\overline{M}_{ij,pq}$ mean number of packets with starting priority r_i and increasing rate b_j , which arrive during the waiting time of C_{pq} and which are to be served before the packet C_{pq} .

The mean waiting time of the packet C_{pq} is given by:

$$\overline{W}_{pq} = \overline{W}_0 + \sum_{i=1}^P \sum_{j=1}^Q \frac{N_{ij,pq} + M_{ij,pq} \overline{T}_{B_{ij}}}{m} \quad (59)$$

For $\overline{N}_{ij,pq}$ we have to consider four cases:

- a) $i \geq p$ and $j \geq q$

$$\overline{N}_{ij,pq} = \lambda_{ij} \overline{W}_{ij} \quad (60)$$

- b) $i \geq p$ and $j < q$ (see Figure 5):

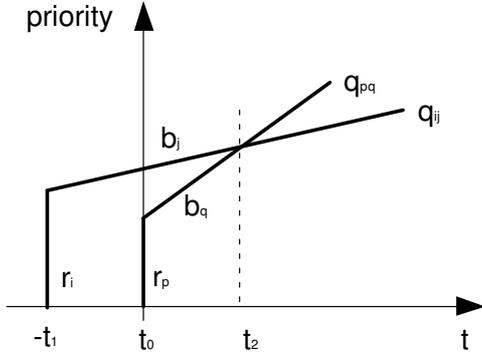


Figure 5: Determination of $N_{ij,pq}$ for $i \geq p, j < q$

The unit of time in which packets C_{ij} are served before C_{pq} can be determined using:

$$r_p + b_q t_2 = r_i + b_j (t_1 + t_2)$$

Then:

$$\begin{aligned} \overline{N}_{ij,pq} &= \int_0^\infty \lambda_{ij} P\{1 < w_{ij}(t)\} \\ &\leq \frac{r_i - r_p}{b_q - b_j} + \frac{b_q}{b_q - b_j} t_1 \} dt \end{aligned}$$

After substitution and using:

$$\overline{W}_{ij} = E[w_{ij}] = \int_0^\infty 1 - P(w_{ij} \leq x) dx$$

we get:

$$\overline{N}_{ij,pq} = \lambda_{ij} \overline{W}_{ij} \frac{b_j}{b_q} \text{ for all } i \geq p, j < q \quad (61)$$

- c) $i < p$ and $j \geq q$ (see Figure 6):

It holds for the relevant packets at $t = 0$:

$$q_{ij}(t = 0) = r_i + t_1 b_j \geq r_p$$

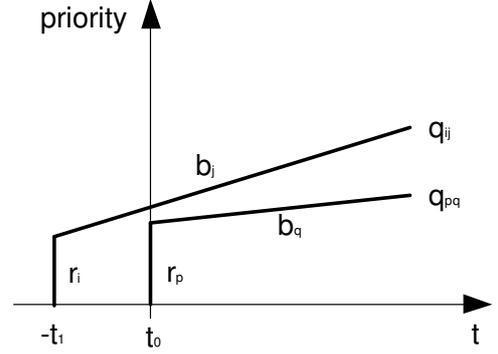


Figure 6: Determination of $N_{ij,pq}$ for $i < p, j \geq q$

with the crucial time point

$$t_1 \geq \frac{r_p - r_i}{b_j}$$

so that:

$$\begin{aligned} \overline{N}_{ij,pq} &= \lambda_{ij} \overline{W}_{ij} - \lambda_{ij} \int_0^{\frac{r_p - r_i}{b_j}} P(w_{ij} > t) dt \\ &\text{for all } i < p, j \geq q \end{aligned} \quad (62)$$

- d) $i < p$ and $j < q$ (see Figure 7):

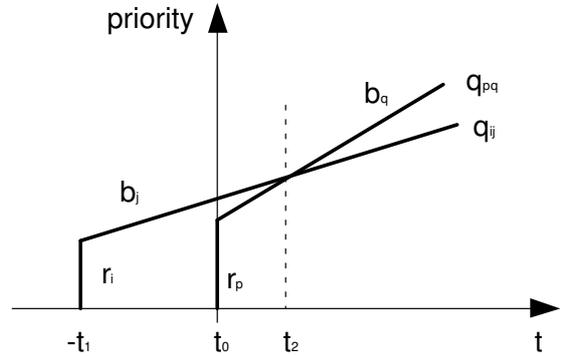


Figure 7: Determination of $N_{ij,pq}$ for $i < p, j < q$

It holds for the packets which are served before C_{pq} :

$$r_p + b_q t_2 = r_i + b_j (t_1 + t_2)$$

With

$$t_1 + t_2 = \frac{r_i - r_p}{b_q - b_j} + \frac{b_q}{b_q - b_j} t_1$$

we get with a similar way as in b):

$$\overline{N}_{ij,pq} = \lambda_{ij} \overline{W}_{ij} \frac{b_j}{b_q} \text{ for all } i < p, j < q \quad (63)$$

In order to determine the sets $\overline{M}_{ij,pq}$ we make here the difference between four cases whereas:

$$\overline{M}_{ij,pq} = 0 \text{ for all } i \leq p, j \leq q$$

All other cases can be treated similarly because the crucial time point t_{ij} is determined by

$$r_i + (W_{pq} - t_{ij})b_j > r_p - W_{pq}b_q$$

hence

$$t_{ij} < \frac{r_i - r_p}{b_j} + W_{pq} \left(1 - \frac{b_q}{b_j}\right)$$

The mean number of packets, which arrive in the time interval $[0, t_{ij})$ and which are served before C_{pq} is then:

$$\bar{M}_{ij,pq} = \lambda_{ij} \int_0^{t_{ij}} P(w_{pq} > t) dt$$

$$\text{for all } i > p \text{ and for all } i \leq p, j > q \quad (64)$$

The determined sets $\bar{N}_{ij,pq}$ and $\bar{M}_{ij,pq}$ can now be substituted in (59) together with the conservation law to get the following form for the mean waiting time:

$$\begin{aligned} \bar{W}_{pq} &= \bar{W}_{FIFO} - \sum_{i=1}^P \sum_{j=1}^{q-1} \rho_{ij} \bar{W}_{ij} \left(1 - \frac{b_j}{b_q}\right) \\ &- \sum_{i=1}^{p-1} \sum_{j=q}^Q \rho_{ij} \int_0^{\frac{r_p - r_i}{b_j}} P(w_{ij} > t) dt \\ &+ \sum_{i=p+1}^P \sum_{j=1}^Q \rho_{ij} \int_0^{t_{ij}} P(w_{pq} > t) dt \\ &+ \sum_{i=1}^P \sum_{j=q+1}^Q \rho_{ij} \int_0^{t_{ij}} P(w_{pq} > t) dt \quad (65) \end{aligned}$$

The unknown probabilities have to be substituted by the proposed approximations to get a recursive equation system, that can be solved numerically.

3.5 Dynamic Priorities with Class Dependent Deadlines

Another strategy for scheduling packets is based on deadlines which are assigned to each packet⁴. R packet classes are defined with a parameter G_i for each class, with:

$$G_1 > G_2 > \dots > G_R$$

The parameters G_i define the time period, which may maximally be elapsed to serve a packet ‘‘in time’’. Packets with the lowest deadlines have the highest priority and vice versa. According to the priority function:

$$q_r(t) = \begin{cases} (t - t_0)/(G_r - t + t_0) & \text{if } t_0 < t \leq G_r + t_0 \\ \infty & \text{if } G_r + t_0 \leq t < \infty \end{cases}$$

the priorities increase faster, whenever the deadline of a packet nears. If the deadline is reached, the packet gets an infinite priority, which forces the system to serve the packet (see Figure 8). Packets with a positive infinite priority are served in a FIFO order.

The approach to calculate the mean waiting times is not very different from the methods used for the other strategies. To determine the set \bar{N}_{ir} we divide the packets into two categories, which are considered separately:

⁴this strategy is also known as Earliest Deadline First (EDF)

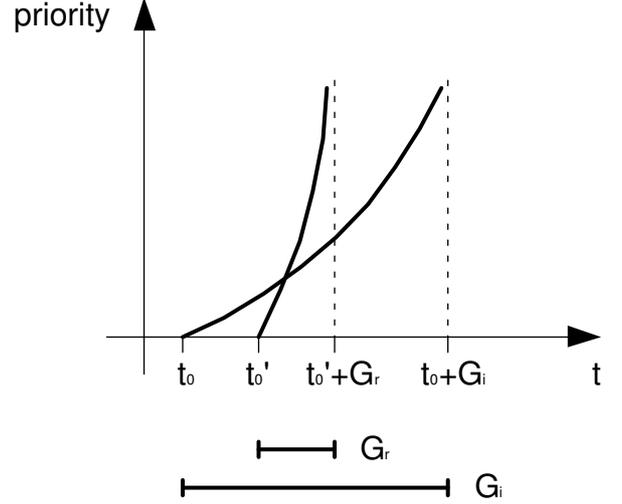


Figure 8: Class dependent deadlines

1. packets, which get an infinite priority after the considered packet C_r does. If $-t_1$ is their arrival time, so

$$G_i - t_1 > G_r$$

2. packets which get an infinite priority at the latest until G_r (at $t_0 = 0$, arrival of C_r), which means

$$G_i - t_1 \leq G_r$$

Using the priority functions we can deduce the sets $\bar{N}_{ir}(1)$ and $\bar{N}_{ir}(2)$. \bar{M}_{ip} can also be determined according to the known methods we used for the other strategies. Substituting these sets in Eq. (1) and using the conservation law gives:

$$\begin{aligned} \bar{W}_r &= \bar{W}_{FIFO} + \sum_{i=r+1}^R \rho_i \int_0^{T_i} 1 - P(w_r \geq G_r) dt \\ &- \sum_{i=1}^{r-1} \rho_i \frac{G_i - G_r}{G_i} \int_0^{G_i - G_r} P(w_i > x) dx \quad (66) \end{aligned}$$

with

$$T_i = \bar{W}_r \frac{G_r - G_i}{G_r} \quad (67)$$

For the unknown probabilities we may again use one of the two proposed approximations.

4 Conclusion

In this paper we investigated two classes of priority based scheduling mechanisms: static priorities and time dependent priorities. In all cases we showed how the mean waiting times of packets of different classes can be computed analytically. We presented results for different arrival and service time distributions. Mono- and multi-processor routers were considered. These results have been successfully applied

for the characterization and analysis of proportional differentiated services [12, 13].

It is still important to investigate how the scheduler parameters have to be chosen, if quality of service profiles for the different traffic classes are given (design problem). It is also interesting to investigate the behavior of the priority based schedulers, when self similar traffic and correlated inter-arrival times are considered.

References

- [1] *C.Dovrolis, D.Stiliadis, P.Ramanathan*: "Proportional Differentiated Services: Delay Differentiation and Packet Scheduling", ACM SIGCOMM '99, Cambridge, MA, Sep. 1999
- [2] *P.Ferguson, G.Huston*: "Quality of Service: Delivering QoS on the Internet and in Corporate Networks", Wiley Computer Publishing, 1998
- [3] *D.Black, S.Blake, M.Carlson, E.Davies, Z.Wang and W.Weiss*: "An Architecture for Differentiated Services", IETF RFC 2475, Dec. 1998
- [4] *K.Nichols, S.Blake, F.Baker, D.Black*: "Definition of the Differentiated Services Field (DS Field) in IPv4 and IPv6 Headers", IETF RFC 2474, Dec. 1998
- [5] *G.Bolch, W.Bruchner*: Analytische Modelle symmetrischer Mehrprozessoranlagen mit dynamischen Prioritäten. Elektronische Rechenanlagen, 26(1), page 12-19, 1984
- [6] *G.Bolch, S.Greiner, H.de Meer, K.S.Trivedi*: Queueing Networks and Markov Chains : Modeling and Performance Evaluation with Computer Science Applications, Wiley, New York, 1998
- [7] *B.Walke*: Realzeitrechnermodelle; Theorie und Anwendungen. Oldenburg Verlag, München 1978
- [8] *L.Kleinrock*: Queueing Systems - Volume I , Wiley, 1975
- [9] *L.Kleinrock*: Queueing Systems - Volume II , Wiley, 1976
- [10] *J.Kulbatzski, J.Liebeherr*: Entwicklung und Implementierung neuer Algorithmen für das Programmpaket PRIORI zur Leistungsbewertung von Mehrprozessorsystemen mit Prioritäten. Studienarbeit, University of Erlangen, 1987
- [11] *J.Kulbatzski*: Das Programmsystem PRIORI - Erweiterung und Validierung mit Simulationen. Diplomarbeit, University of Erlangen, 1989
- [12] *L.Essaft, G.Bolch, A.Andres*: "An Adaptive Waiting Time Priority Scheduler for the Proportional Differentiation Model", ASTC HPC'01, Seattle, Apr. 2001
- [13] *L.Essaft, G.Bolch, H.de Meer*: "Dynamic Priority Scheduling for Proportional Delay Differentiated Services", MMB, Aachen, Sep. 2001

SOME REMINISCENCES ON THE HISTORY OF HARDWARE AND SOFTWARE FOR SIMULATION 1963-2003

RICHARD N ZOBEL

Department of Computer Science
The University of Manchester
Oxford Road, Manchester M13 9PL, United Kingdom
rzobel@cs.man.ac.uk, r.zobel@ntlworld.com

Abstract: During the past fifty years the world has changed and simulation has advanced out of all recognition. The author has progressed from very large analogue computers in the early 1960s, through hybrid analogue/digital systems in the late 60s, to design and construction of digital signal processing computers in the 70s, onto early microcomputers in the 80s, used for simulation, and more recently to distributed simulation in the 90s and 00s, always up with the leading edge. This paper summarises some of these developments and considers the current situation and future prospects.

Keywords: History, Hardware, Software, Languages, Applications, Parallel and Distributed, Virtual Environments.

INTRODUCTION

I still have some vacuum tubes (valves) and some early metal can transistors with serial numbers! How times have moved on. My introduction to simulation began with a simulation of the Mark II Seaslug surface to air missile which of course required the parallel solution of many simultaneous non-linear, time-varying, ordinary differential equations. My PhD concerned the design of a new hybrid analogue multiplier and some fast (for those days) analogue to digital converters. At this point, I had my first experience of simulation societies (UKSC) and organised a conference at Manchester in 1970. A successful excursion into digital signal processing and design of signal processing computers was terminated by the arrival of the microprocessor. Fortunately, John Stephenson (Bradford) steered me back to simulation and in 1985 I attended my first SCS conference in Reno, Nevada, USA [1]. Since then simulation has enabled me to take part in many organised events in over 30 countries, making in the process, many friends both nationally and internationally, and in the process enjoying myself immensely.

THE EARLY DAYS OF SIMULATION USING ANALOG COMPUTING

Lord Kelvin first outlined the idea of a differential analyser, using mechanical ball and disc integrators, in 1876. However, the lack of contemporary technology prevented the idea from being realised. The earliest implementation was by Prof. Vannevar Bush of M.I.T. in 1930. Prof.

Douglas Hartree, FRS, of St Johns College, Cambridge, with his assistant Arthur Porter, a Manchester undergraduate, produced a Meccano machine in 1934. This was followed up in 1935 by a full scale machine by Hartree and Porter with engineering support by Metropolitan Vickers and funding by Sir Robert McDougall. It was for many years recognised as the best and most used machine for gunnery prediction and other applications [2,3].

By the end of the Second World War, vacuum tube based operational amplifiers were beginning to be used to construct an electronic equivalent for solving differential equations. That technology led to the development of the general purpose analogue computer, used mostly at that stage for research, development and performance assessment of military systems.

Fundamentals

Systems of ordinary differential equations, representing the mathematical models of real dynamic systems, may be solved using a small subset of mathematical operations directly or by means of combinations of these with some additional electronic circuits.

Figure 1 illustrates the basic uses of operational amplifiers for addition and subtraction of voltages and of the integration of the sum of voltages. For the examples given, the adder has an output voltage given by:

$$\begin{aligned} -V_0 &= V1.Rf/R1 + V2.Rf/R2 \\ &= k1V1 + k2V2 \end{aligned}$$

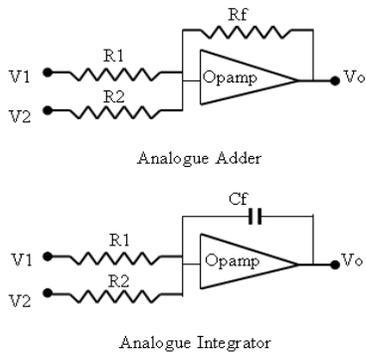


Fig. 1 Basic analogue computing elements

The integrator has an output voltage given by:

$$-V_0 = \int_0^t (V1/CfR1 + V2/CfR2)dt$$

$$-V_0 = \int_0^t (k1V_1 + k2V_2)dt$$

Voltages represent scaled equation variables.

Variables may be negated using an inverter (an adder with only one input and $R_f = R_1$).

Other required mathematical operation are:

$$V_0 = kV_1.V_2 \text{ (Multiplication)}$$

$$V_0 = f(V_1) \text{ (Function Generation)}$$

Multiplication may be achieved using a precision (10 turn) potentiometer (for fixed coefficients) or a time division multiplier, or quarter squares multiplier ($A.B = 0.25 \{(A+B)^2 - (A-B)^2\}$), the squaring being achieved by a fixed diode/resistor network. The latter was also used for non-linear function generation, but with adjustable segments.

Figure 2 is a reproduction of a photograph of the Solartron 247 system at Sperry Gyroscope Company, Bracknell, U.K., circa 1964.

There are 15 racks of mainly valve based operational amplifiers and other functional units. Clearly shown are 5 patch panels, where the units were interconnected to form the analogue network for solving the current problem. Each rack consumed around 1kW of power. The system cooling can clearly be seen above the units. The addition of a second machine of 12 racks yielded a total capability for solving problems requiring the resulting 1000 operational amplifiers. The operator (the author) is sitting in front of one of the control desks, setting coefficients on one of the potentiometers.

Analogue computers are parallel computers, solving sets of simultaneous differential algebraic equations (DAEs) in real time (or scaled

faster/slower than real time). However, unlike modern digital computers, analogue hardware cannot be used to quasi-simultaneously compute more than one function as with a digital processor, because it is not possible to share variables on a single analogue hardware unit due to lack of an equivalent storage mechanism and the near impossibility of remembering and then restoring the correct charges on the integrator capacitors with such large values of the order of .001 μ F to 1 μ F, required for precision integration with low drift.

Accuracy, Repeatability and Correctness

There are limits to the accuracy of analogue systems, due to the physical and economic costs of producing precision resistors and capacitors. There is also the problem of variation with temperature and time. Further, as the resolution is increased, noise arising from many sources limits further increases in accuracy. The repeatability of results for a particular simulation run is an important issue and is determined by the combined accuracy and stability of the parallel analogue computing units. It was difficult to achieve better than around 0.1% in practice, due to the combination of errors due to the 0.01% tolerance on many components.

Test runs, duplicated on a KDF9 digital computer, exhibited excellent resolution and repeatability. However, the correctness claimed was somewhat misguided. The problems associated with integration algorithms, sample rates and sparse matrices had yet to be appreciated. The latter appeared as a very large ratio between long time constants associated with the physical dimensions of the vehicle and the short time constants associated with the required accelerations. Over a period of time the analogue and digital test runs got closer and we all learned a lot about our respective disciplines and to respect each others professional expertise and experience!

Valves, Transistors, Hybrids and Logic

The amplifiers were constructed using valves running on supplies of ± 300 volts, with reference voltages of ± 100 volts used for constants and initial conditions. Mechanical choppers, used for d.c. offset stabilisation, were replaced by semiconductor devices with greater reliability. The time division multipliers employed germanium transistors for switching, which limited the ambient temperature for accurate operation (germanium melts at around 55 Celsius, hence the air conditioning requirement for both machine and operators. A few voltage comparators and simple logic gates provided for some useful logical operations. Examples of the use of these included the detection of range coincidence between missile and target, transfer from boost phase to guidance phase, and of out of scale variables.

THE EMERGENCE OF DIGITAL SIMULATION

As digital computers became faster and cheaper their advantages became obvious, and the demise of the traditional analogue computer became inevitable. However, there was still much to learn about digital simulation. Perhaps the most obvious problem was the loss of “feel” and the immediacy of the analogue system. The real-time interaction was much better than a 24 hour turn-around and the numerical teletype printout for the digital simulation runs.

Processing Power and Memory

Initially, the processing power and memory limitations of the digital computers of the 1960s and 70s severely limited their use. However, the appearance of the microprocessor and advances in both random and serial access memories began to change this. Progress has of course been and continues to be rapid, leading to the current substantial power of both PC and Workstation.

Digital Equivalents of Analogue Functions

Initially, most attempts concerned the reproduction of analogue techniques using numerical

equivalents. Some were quite successful, especially in the area of digital equivalents of analogue filters. A particular concern was the appearance of problems associated with word length and recursion convergence. Those of us with knowledge of both continuous and discrete (sampled) systems equated word length with gain, recursion with feedback, and convergence with stability.

Problems of Integration

The analogue integrator is a true integrator, admittedly with imperfections at both low and high frequency. However, sampled data approximations are fraught with other problems. It has taken many years to recognise that one integrator algorithm is not the answer to solving all differential equation systems. There is a black art (aka experience) in knowing which to use in a particular circumstance.

Speed

In the early days of analogue computers, digital computers were very slow compared to their analogue counterparts. This was partly due to the parallel nature of the latter. Thus the concept of real-time simulation was restricted to the analogue machine. The requirement for real-time simulation arose from two different requirements. The first was



Fig. 2 Solartron 247 Analogue Computer System

the need for including either humans or real hardware or both in the loop (time constants are fixed) and the second was the need for large numbers of simulation runs to be completed in a reasonable time (not just a lot too late). The first requirement remains in respect of training simulators. Modern digital systems employ multiple processors for different activities (such as input/output, graphics, sound, etc), parallel processors for partial differential equations or large numbers of ordinary differential equations, and networked systems for large training simulation exercises.

LANGUAGES

Those of us using simulation are accustomed to using simulation packages for speed and convenience. Newcomers however are often ignorant of the existence of such packages. In the early days only Fortran and machine code were available.

General Purpose Languages

Simulations have been and continue to be written, often inappropriately, in general purpose programming languages. Concerns over appropriateness, convenience, speed of execution and compatibility with other software tools such as databases and graphics are often ignored to the disadvantage of such inexperienced users. For simple simulations this probably does not matter, but for real, larger problems it does, as they are usually inappropriate. However, until simulation is taught as a general purpose tool, this is likely to be a problem that will persist for some time to come.

Simulation Languages

As the use of simulation has developed, increasingly more sophisticated packages have been developed, taking advantage of the increasing power and utility of computers. ACSL, ESL, Simplex 3 and Arena are examples of the many systems currently on offer supporting not only model development and simulation development, but also all of the tools necessary to support simulation studies for continuous system, discrete event or mixed applications.

Equation Solving

Historically, model equations had to be manipulated by hand before being programmed into a simulation language. Modern simulation systems do this automatically, using functional programming techniques.

Library Functions

Simulation specific techniques and models are now provided via libraries including mathematical functions such as integration, function generation, axis transformation, dead space, logic functions, queues, stacks, etc., supported with graphical model building using icons and interconnections.

GRAPHICS

Graphics did not arrive until the 1970s. Before this, pen recorders, numerical results tables and character based graphs printed on teletypes were the order of the day. The arrival of graphical

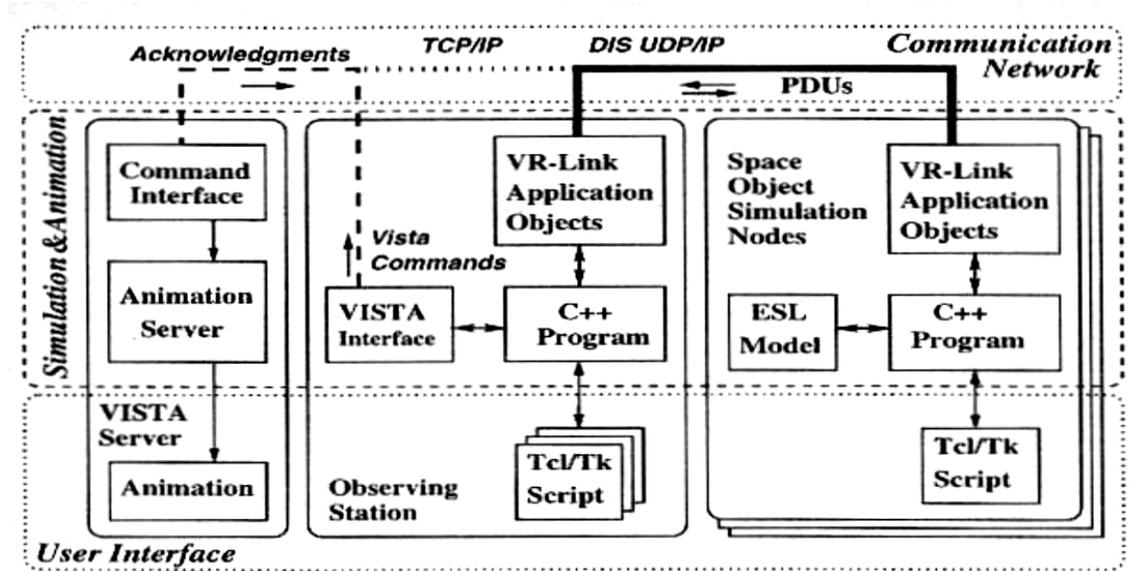


Fig. 3 DIS System Linking Distributed Simulations and Animator

screens, used almost universally now, allowed block diagrams of models and systems, full multi-channel graph systems, animation and virtual reality to develop.

COMPUTERS

Early digital computers were used for simulations, but they were slow, and integration and algebraic loops caused problems. However, the arrival of the microprocessor, led to a revolution in simulation applications of increasing variety and complexity. The earliest personal computers were used experimentally for simulation and their potential was obvious in terms of the re-acquired feel and fast turn around for the systems being simulated.

Main-Frames, Minicomputers and Microcomputers

Mainframes were difficult to use because they were distant and any resemblance of the hands-on approach was lost, particularly in relation to the delay between submitting a program to the computer centre and receiving a print out (or list of programming or run time errors) of at least 24 hours. Graphical printout was inconveniently character based.

Minicomputers, such as the PDP11, were a significant advance at lower cost and allowed a more hands on approach. The performance limited applications to those which fitted the RAM and disk parameters and fell short of real-time for many applications.

Early microcomputers, based on the Intel 8088, were too slow and too small, but gave an insight to their potential as personal computers (PC). However, the rapid advances in speed, power and memory gave rise to desktop machines of significant use for many simulation applications. Of course, this still left the big problems to the much larger mainframes and computer centres.

PARALLEL COMPUTERS AND NETWORKED COMPUTERS

With the increasing reduction in the costs of processors and memory, and the improving application of memory management systems, the rapid development of parallel computers came about, leading to the possibility of tackling large scale simulations of such problems as weather forecasting and stress analysis in addition to the larger ordinary differential equation problems with large numbers of equations. The rapid advances in networking also led to the development of

distributed simulations, interconnected over private or public networks especially for training.

DISTRIBUTED SIMULATIONS AND VIRTUAL/ SYNTHETIC ENVIRONMENTS

Training simulators have been an important part of military systems provision for many years. Distributed interactive simulation came into being first with SIMNET an attempt to connect military simulators together. Following this the DIS system for interconnecting training simulators together successfully gave rise to the IEEE 1278 standard. The author and one of his PhD students, was involved in a project concerning space missions. The systems comprised networked Unix workstations, each running a spacecraft simulation, interconnected by VR-Link to an animator (VISTA) shown in figure 3 [4]. In this case, the space objects were a space shuttle and a space station with an observing station on the ground. The motions of these objects were simulated using ESL, and visualised and animated using VISTA, the latter two packages being provided by iSIM and ESTEC respectively. The background of the Earth, Sun, Moon, Planets and Stars were also provided by ESTEC.

Following this, the high level architecture (HLA) provided standards and a run time interface (RTI) for management of a federation of a network of simulation federates. A consequence of this for non-military applications is the need for provision of security for both obtaining access to an HLA federation and for authentication and passing of data and messages. The latter is discussed in detail in a technical and tutorial paper by the author [5].

SUPERCOMPUTERS AND GRIDS

In recent years, supercomputers, comprising of upwards of a 1000 processors with massive local and global memory have been developed and used for solving large problems. Perhaps the best known of these are the weather forecasting machines in Bracknell, UK and in Washington, USA. These machines basically solve the Navier-Stokes partial differential equation (PDE) for the global atmosphere for weather prediction. Other PDEs are solved for dynamic stress analysis, fluid dynamics problems and atomic reactions.

Recently, projects have evolved to link such supercomputers together to solve even larger problems as parallel/distributed architectures, or to obtain finer grain solutions, by using grids of supercomputers [6,7]. Although an excellent concept, much work is being done on algorithms and optimisation as was done for parallel computing.

THE FUTURE

The increasing power of individual and networked computers, coupled with the universal availability of powerful graphics and virtual reality is making simulation a universally useful tool set. The recognition of the validity of discrete event, continuous and mixed simulations as a normal part of project planning, design, implementation, testing and training in terms of engineering, management and commercial activities is of great significance. Consequently, new areas of application of simulation are emerging. Of particular importance, are those connected with software agents [8], distance learning and web-based applications [5].

CONCLUSIONS

It is clear that simulation technology and applications now pervade almost every human activity and endeavour. As the tools become more widely used and integrated, and the use of simulation is taught more widely as a valuable addition to applied mathematics, the acceptance and future of simulation seems bright. The history of simulation is richly sprinkled with the names of stars in every subject. It probably starts centuries ago, perhaps with Leonardo da Vinci or earlier, but certainly the work of Lord Kelvin, and of Hartree and Porter gave simulation a major kick start. The foundation of the Society for Computer Simulation (SCS) in 1952, only 4 years after the first stored program digital computer was demonstrated at Manchester was an event of major significance. The establishment of the European Simulation Societies in Europe around 35 years ago has also contributed to its successful development. The author is grateful to have had the opportunity of taking part in this activity and of the consequences for his international travel experiences and his many friends gathered over the years.

ACKNOWLEDGEMENTS

The author would like to acknowledge all of the contributions of his colleagues and postgraduate students in the Dept. of Computer Science at Manchester, in SCS International, the Board and Member Societies of EUROSIM, in iSIM and ESTEC, and all of his colleagues and friends worldwide who have in various ways contributed to a fascinating career mainly concerned with simulation and its areas of application.

REFERENCES

[1] Adib D. and Zobel R.N. 'Simulation of uHARP, A Multi-Microprocessor Based Hierarchical Array

Processor for Signal Processing.' In *Proc. SCSC 86, Summer Computer Simulation Conf.*, Reno, Nevada, USA, July 1986, pp 965-971. SCS, San Diego.

[2] Hartree D.R. and Porter A. 'The Construction and Operation of a model Differential Analyser' In *Memoirs and Proc. of the Manchester Literary and Philosophical Soc.* 1934-45 Vol. LXXIX pp.51-81

[3] Hartree D.R. and Porter A. 'The Application of the Differential Analyser to Transients on a Distortionless Transmission Line'. *Jnl. Institution of Electrical Engineers.* Vol. 83, No. 503, Nov. 1938

[4] Tandayya P. PhD thesis, 2000, Manchester University. p151.

[5] Zobel R.N. 'Security for Internet and Web-based Applications' Invited paper in 4th Int. Conf. on Information Integration and Web-Based Applications and Services (iiWAS2002), Sept. 2002, Bandung, West Java, Indonesia. ISBN 3-936150-18-4. pp. 6-16.

[6] Foster I., Kesselman C., Tuecke S. 'The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration'. Open Grid Service Infrastructure WG, Global Grid Forum, June 22, 2002.

[7] Foster I., Kesselman C., Tuecke S. 'The Anatomy of the Grid: Enabling Scalable Virtual Organizations'. *Int. Jnl. Supercomputer Applications*, 15(3), 2001.

[8] Iwu O.F. and Zobel R.N. 'Network Attack Profiling: Using Agent Based Simulation to Gather Forensic Information'. In *Proc. 3rd Workshop: Agent-Based Simulation*, Passau, Germany. Apr. 2002. SCS Erlangen., ISBN 3-936150-17-6, pp180-186.

ABOUT THE AUTHOR



RICHARD ZOBEL graduated in Electrical Engineering from London University in 1963. His first experience of simulation was obtained during 1962-66 at Sperry Gyroscope whilst working on military projects, using valve analog computers. His Ph.D., obtained in 1970 at Manchester, concerned hybrid analog-digital computing. As

Lecturer and Senior Lecturer he became involved in digital signal processing, instrumentation and design environments with special emphasis on the simulation aspects of real-time embedded systems. He is a former Chairman of the United Kingdom Simulation Society (UKSim), Former Secretary of the European Federation of Simulation Societies (EUROSIM), and was a European Director of SCS, the Society for Computer Simulation International. His current research work concerns distributed simulation for non-military applications, model re-use, distributed simulation model databases, issues of verification and validation of re-useable simulation models and security for distributed simulation under commercial network protocols. He is now semi-retired, but still very active, both in teaching and research.

INTELLIGENT SYSTEMS

An Improved Self-Tuning Mechanism of Fuzzy Control by Gradient Descent Method

Ahcène Habbi*, Mimoun Zelmat

Laboratoire d'Automatique Appliquée, University of Boumerdès, FHC
35000 Boumerdès, Algeria.

*Corresponding author. Tel./Fax: +213-24-816905, E-mail: habbi_hacene@hotmail.com

Abstract – An improved self-tuning mechanism of fuzzy control by gradient descent method is presented. The membership function parameters are tuned by minimizing some criterion defined on the control output using the steepest gradient descent algorithm. The factor which controls how much the fuzzy controller parameters are altered is adjusted continuously using a set of fuzzy rules. This varying factor is determined with respect to the values of the objective function and its change. An application to the control output optimization of a PI-type fuzzy controller shows the superiority of the proposed self-tuning mechanism over a previously published approach in terms of both precision and convergence rate.

Keywords: adaptive fuzzy control; fuzzy reasoning; gradient descent algorithm; self-tuning.

I. INTRODUCTION

Recently, the "fuzzy logic wave" has reached the community of automatic control. Fuzzy logic controllers (FLC's) have been successfully used for a number of complex, ill-structured industrial processes [KIR, 98; FIS, 99]. Since most of the real-world processes that require automatic control are nonlinear in nature, FLC's can be designed to cope with a certain amount of process nonlinearity and parameter variations. Therefore, more attention has been paid to the problem of how to design a suitable adaptive fuzzy controller for a given process. Different types of adaptive FLC's have been developed and implemented for various practical applications [DRI, 96]. Adaptation mechanisms for FLC's are classified according to which the controller parameters are adjusted. Adaptive controllers that adjust the fuzzy set definitions or scaling factors are called self-tuning controllers (STC). However, when the fuzzy rule base is altered, the controller is called self-organizing controller (SOC).

Many works have centered on the use of mathematical optimization techniques (see, [BOR 90]) to tune the set definitions so that the output from the FLC matches a suitable set of reference data as closely as possible [DRI, 96; WON, 98; HE, 93]. The basic example of this is given by Nomura et al. in [NOM, 91], where they use gradient descent algorithm to tune simple membership functions. The controller is tuned iteratively by minimizing the

square error between the FLC output and the desired output given by the training data. This tuning method may be very good for control systems, but its applicability is closely related to the convergence rate of the adaptation algorithm, especially when it is used on-line as Gloennec did in the control of a mixer tap [GLO, 91]. The main problem is how to adequately choose the constant which controls how much the controller parameters are altered at each iteration in the gradient descent algorithm. This, however, puts an unnecessary and often inappropriate constraint on the design. A suitable choice of the gradient step may accelerate the convergence of the algorithm (see, for example [SAD, 75]) and then enhance the performance of the fuzzy control loop.

This limitation of the self-tuning mechanism of fuzzy control suggested by Nomura et al. motivated us to investigate techniques of improving the original algorithm by using experts' knowledge rather than mathematical models or heuristics methods. In the modified version the gradient step is adjusted continuously with the help of IF-THEN fuzzy rules. The amount of variation of this factor is determined with respect to the current values of the objective function to be minimized and its change. To check the effectiveness of the proposed approach, we consider the problem of minimizing the matching error affecting the input information of a fuzzy controller.

II. THE PROPOSED ADAPTATION MECHANISM

A. The fuzzy controller from Nomura et al.

The self-tuning method of fuzzy controllers developed by Nomura et al. is a well-known gradient descent technique to optimize both the fuzzy antecedent and crisp consequent parts. Our objective here is to improve the performance of the gradient descent tuning algorithm by adapting the gradient step iteratively using a set of fuzzy rules to achieve better precision and better convergence rate. This method relies on having a set of input-output data against which the controller is tuned. The FLC consists of a set of n fuzzy rules of the form

Rule i :

IF x_1 is $X_1^{(i)}$ and.....and x_m is $X_m^{(i)}$ THEN u is $U^{(i)}$

where x_1, \dots, x_m are the controller inputs, u is the control output variable, i is the rule number, $X_1^{(i)}, \dots, X_m^{(i)}$ are linguistic values of the rule-antecedent, $U^{(i)}$ is the linguistic value of the rule-consequent.

The membership functions, $\mu_{X_j}^{(i)}$, of the antecedent part are triangles described by a peak value a_{ij} , and a support b_{ij} , in the defined universe of discourse. The membership function is thus given by

$$\mu_{X_j}^{(i)}(x) = 1 - \frac{2|x - a_{ij}|}{b_{ij}} \quad (1)$$

The control output membership function is a fuzzy singleton set defined on the real number u_i . Using the max-dot composition and the Center-of-Area defuzzification method, the global control output from the fuzzy rule set is given by

$$u = \frac{\sum_{i=1}^n \mu_i u_i}{\sum_{i=1}^n \mu_i} \quad (2)$$

where

$$\mu_i = \prod_{j=1}^m \mu_{X_j}^{(i)}(x_j) \quad (3)$$

B. The modified gradient descent algorithm

If a reliable set of training data is available that describes the desired control output, u^r , for various values of the process state, $x_1^r, x_2^r, \dots, x_m^r$, the fuzzy controller can be tuned by minimizing the square error between the FLC output and the desired output given by the reference data. Nomura et al. have chosen to minimize the following objective function

$$J = \frac{1}{2} (u - u^r)^2 \quad (4)$$

Substituting (2) and (3) into (4) gives the following objective function

$$J = \frac{1}{2} \left(\frac{\sum_{i=1}^n \left(\prod_{j=1}^m \mu_{X_j}^{(i)}(x_j^r) \right) u_i}{\sum_{i=1}^n \left(\prod_{j=1}^m \mu_{X_j}^{(i)}(x_j^r) \right)} - u^r \right)^2 \quad (5)$$

The steepest descent algorithm seeks to decrease the value of the objective function (5) with each iteration t . In this case, the objective function parameters we wish to alter are the membership function parameters a_{ij} , b_{ij} and u_i . Solving this optimization problem gives the following iterative equations of the parameter values

$$a_{ij}(t+1) = a_{ij}(t) - \lambda_1(t) \frac{\partial J}{\partial a_{ij}}, \quad i = 1, \dots, n; \quad j = 1, \dots, m, \quad (6)$$

$$b_{ij}(t+1) = b_{ij}(t) - \lambda_2(t) \frac{\partial J}{\partial b_{ij}}, \quad i = 1, \dots, n; \quad j = 1, \dots, m, \quad (7)$$

$$u_i(t+1) = u_i(t) - \lambda_3(t) \frac{\partial J}{\partial u_i}, \quad i = 1, \dots, n. \quad (8)$$

The gradient step updating factor λ_l ($l=1,2,3$) is calculated using fuzzy rules of the form

Rule k :

IF $J(t)$ is $F_1^{(k)}$ and $\Delta J(t)$ is $F_2^{(k)}$ THEN $\lambda_l(t)$ is $G_l^{(k)}$

where $J(t)$ and $\Delta J(t)$ are values of the performance criterion and its variation at the iteration t , respectively. $F_1^{(k)}, F_2^{(k)}$ are the linguistic values of the rule-antecedent, and $G_l^{(k)}$ is the linguistic value of the rule-consequent.

The functional relationship of λ_l can be viewed as

$$\lambda_l(t) = f(J(t), \Delta J(t)) \quad (9)$$

where f is a nonlinear function (computational algorithm) of J and ΔJ , which is described by a fuzzy rule base.

For determining the fuzzy rule base for computation of λ_l we have taken into account some important considerations related to the optimization problem, i.e., the current values of the objective function to be minimized, the change-of-error and the direction of the gradient vector of the membership function parameters. We attempted to extract IF-THEN fuzzy rules from a linguistic description of a general optimization procedure, that is:

"we have to look for decreasing the value of the objective function most rapidly in the direction of the negative gradient vector when the current point seems to be significantly far from the desired solution; to slow down the procedure if the current point is close to the solution. If the optimum point is reached the optimization is complete".

It is very important to note that the rule base for computation of the gradient step will not be dependent on the choice of the rule base for the controller.

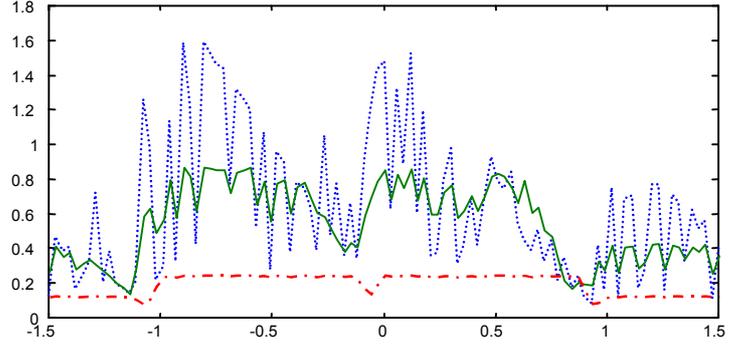


Fig. 3. The matching error for selected values of σ :
 $\sigma = 0.25$ (dash) ; $\sigma = 1$ (solid) ; $\sigma = 3$ (dash-dot)

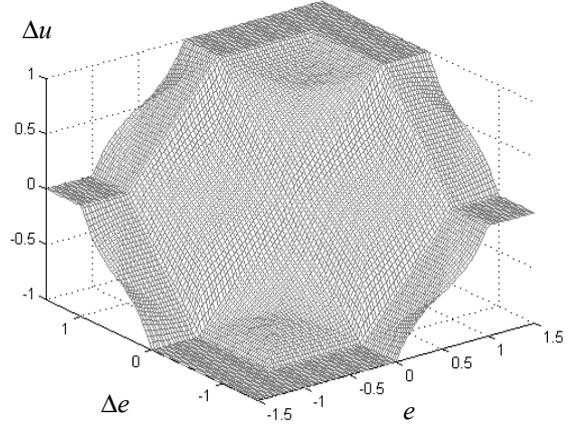


fig. 2. The control surface of the PI-type fuzzy controller

C. The Tuning procedure

Once a set of reliable controller input-output data has been collected, a possible optimization procedure is as follows:

Step 1: the rules are fired on the input data $(x_1^r, x_2^r, \dots, x_m^r)$ to obtain the antecedent value μ_i for each rule and the real-valued control output u .

Step 2: gradient step values λ_i are updated using (9).

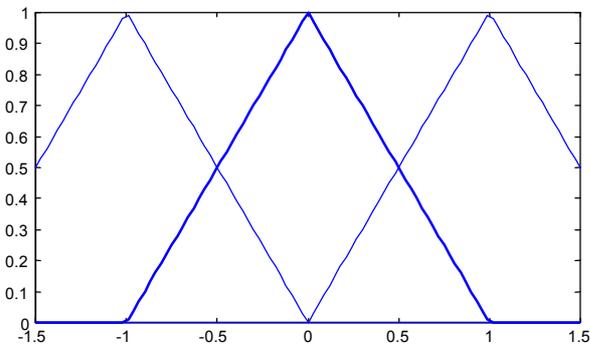


Fig 1. Fuzzy set definition for the input/output variables of the PI-type fuzzy controller

Step 3: parameters u_i are updated using (8).

Step 4: rule firing is repeated using the new values of u_i .

Step 5: parameters a_{ij} and b_{ij} are updated by (6) and (7), using the values of u_i , μ_i and u .

Step 6: inference error $D(t) = \frac{1}{2}(u(t) - u^r)^2$ is calculated.

Step 7: if the change-of-error $|D(t) - D(t-1)|$ is suitably small, the optimization is complete; otherwise it is repeated from step 1.

III. APPLICATION TO THE CONTROL OUTPUT OPTIMIZATION OF THE PI-TYPE FUZZY CONTROLLER

In order to demonstrate the performances of the proposed tuning algorithm, we consider here the optimization problem of a PI-type fuzzy controller. This control problem is solved using both the tuning mechanism suggested by Nomura et al. and the modified version proposed in this paper. The control output of the PI-type FLC is given by

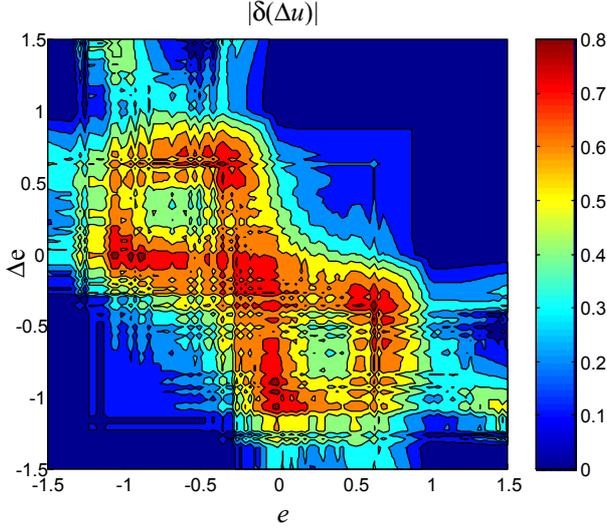


Fig. 4. Contour-plot of the change in the control-output of the PI-type fuzzy controller for $\sigma = 1$

$$u(k) = u(k-1) + \Delta u(k) \quad (10)$$

where k is the sampling instance and $\Delta u(k)$ is the incremental change in controller output. We emphasize here that this accumulation (10) of controller output takes place outside the FLC and is not reflected in the rules themselves. All membership functions for controller inputs, i.e., error (e) and change of error (Δe) and incremental change in controller output (Δu) are triangular-shape partitions uniformly distributed on the common interval $[-1.5, 1.5]$ with three fuzzy set terms: N (negative); Z (zero); P (positive) as depicted in Fig. 1. The PI-type control surface is shown in Fig. 2.

The noise affecting the controller inputs is modeled as an additive random Gaussian variable with a zero mean and standard deviation σ , namely $N(0, \sigma)$. Fuzzy partitions are exposed to controller inputs (e) and (Δe) as well as their noisy versions (e') and ($\Delta e'$). Thus, in fact, the noisy version of (e) induces:

$$\mu_N(e'), \mu_Z(e'), \mu_P(e') \quad (9)$$

instead of the original one:

$$\mu_N(e), \mu_Z(e), \mu_P(e). \quad (11)$$

The matching error is expressed in terms of the overall sum of the absolute differences and is given by

$$r(e) = |\mu_N(e) - \mu_N(e')| + |\mu_Z(e) - \mu_Z(e')| + |\mu_P(e) - \mu_P(e')| \quad (12)$$

The results in terms of $r(e)$ are plotted in Fig. 3. for selected values of σ . The robustness of the control algorithm for which the input fuzzy partition plays the role of an interface is closely related to the input information. Then, any change of the input error, if not

absorbed by the fuzzy partition, may have a meaningful effect on the processing error. The difference between the control value (u) obtained for exact input information (e and Δe) and that (u') generated by the controller for the noisy version of the input is illustrated by the contour-plot of the change in the control-output in Fig. 4. In general, this error is viewed as a suitable indicator of fault-tolerance for the fuzzy controller [PED, 93]. In order to optimize the dynamic behavior of the FLC, we propose to use the modified self-tuning mechanism. We choose to only tune the rule-consequent membership functions, via equation (8). The factor which controls how much the crisp consequent values are altered is updated iteratively using equation (9). The centers and the widths of the triangular input fuzzy sets are maintained constant.

A. Performance analysis of the proposed adaptation mechanism

The performances of the proposed tuning mechanism are compared with those obtained by using the original version suggested by Nomura et al. For a clear comparison, we have used some performance measures such as, the final value of the objective function J , and the number of iterations I . This may give a good idea on the precision and the convergence rate of the algorithm with respect to the initial parameters. It can be noticed from table (1) that the proposed tuning mechanism gives the best results in all the cases considered in the simulation study. For example, with a required precision, η , of 10^{-10} , the problem is solved in only 7 iterations using the proposed mechanism; hence it makes 79 iterations with the Nomura's algorithm. In this case, our modified version is some 70 times faster than the simple gradient descent method. In the last simulation case (for $\lambda_0 = 5$), the Nomura's method seems to be inappropriate. The optimization procedure proceeds out of the searching space leading to the divergence of the algorithm. However, the proposed tuning algorithm is much more effective because of the constraints imposed linguistically on the evolution of the gradient step. Using an appropriate architecture (see [GLO, 91]), it will be interesting to implement this mechanism on-line to form an adaptive fuzzy knowledge-based controller.

TABLE I
PERFORMANCE ANALYSIS OF THE MODIFIED TUNING ALGORITHM

Initial parameters		Nomura's method		The proposed method	
λ_0	η	Criterion (J)	Iterations	Criterion (J)	Iterations
0.25	10^{-4}	$9,8079 \cdot 10^{-5}$	25	$4,3429 \cdot 10^{-5}$	02
	10^{-6}	$9,3436 \cdot 10^{-7}$	43	$3,8836 \cdot 10^{-8}$	03
	10^{-10}	$8,4799 \cdot 10^{-11}$	79	$6,6947 \cdot 10^{-11}$	07
0.75	10^{-4}	$4,5316 \cdot 10^{-5}$	08	$2,2765 \cdot 10^{-5}$	02
	10^{-6}	$4,9237 \cdot 10^{-7}$	13	$2,0357 \cdot 10^{-8}$	03
	10^{-10}	$5,8124 \cdot 10^{-11}$	23	$6,2062 \cdot 10^{-11}$	07
1.00	10^{-4}	$8,2465 \cdot 10^{-5}$	05	$1,4914 \cdot 10^{-5}$	02
	10^{-6}	$4,0776 \cdot 10^{-6}$	09	$1,3337 \cdot 10^{-8}$	03
	10^{-10}	$3,7596 \cdot 10^{-11}$	17	$4,8688 \cdot 10^{-11}$	07
5.00	10^{-4}	Divergence		$1,0216 \cdot 10^{-7}$	03
	10^{-6}	Divergence		$1,0216 \cdot 10^{-7}$	03
	10^{-10}	Divergence		$9,1359 \cdot 10^{-11}$	04

IV. CONCLUSION

The problem of designing an adaptive fuzzy controller using gradient descent method has been tackled by proposing an improved self-tuning mechanism. The main contribution of this paper consists of using the approximate reasoning for modeling the optimization strategy with the help of IF-THEN fuzzy rules. The proposed fuzzy rule base is used for the computation of the gradient step which is adjusted continuously with respect to the amount of variation of the performance criterion and the direction of the gradient vector of the fuzzy controller parameters. It has been demonstrated by simulation that the proposed self-tuning mechanism gives more interesting results than a previously published approach. The modified tuning procedure can be used on-line to form an adaptive FLC, if suitable reference data can be generated by considering an appropriate architecture.

REFERENCES

- [BOR 90] Borne P., G. Dauphin-Tanguy, J. P. Richard, F. Rotella et I. Zambettakis, *Commande et optimisation des processus*. Méthodes et techniques de l'ingénieur, Editions Hermès, 1990.
- [DRI 96] Driankov D., H. Hellendoorn, M. Reinfrank, *An introduction to fuzzy control*. Springer-Verlag, 1996.
- [FIS 99] Fischle K., D. Schröder, An improved stable adaptive fuzzy control method, *IEEE Transactions on Fuzzy Systems*, vol. 7, p. 27-40, Feb. 1999.
- [GLO 91] Glorennec P.Y., Adaptive fuzzy control," *Proc. of the IFSA '91*, p. 33-36, 1991.
- [HE 93] He S. Z., S. Tan, F. L. Xu, P. Z. Wang, Fuzzy self-tuning of PID controller, *Fuzzy Sets Syst.*, vol. 56, p. 37-46, 1993.
- [KIR 98] Kiriakidis K., Fuzzy model-based control of complex plants, *IEEE Transactions on Fuzzy Systems*, vol. 6, p. 517-529, Nov. 1998.
- [LEE 93] Lee J., On methods for improving performance of PI-type fuzzy logic controllers, *IEEE Transactions on Fuzzy Systems*, vol. 1, p. 298-301, Nov. 1993.
- [NOM 91] Nomura H., I. Hayashi, N. Wakami, A self-tuning method of fuzzy control by descent method, *Proc. of the IFSA '91*, p. 155-158, Brussels, 1991.
- [PED 93] Pedrycz W., *Fuzzy control and fuzzy systems*, Second extended edition, RSP Ltd. England, 1993.
- [RAJ 99] Rajani K. Mudi, Nikhil R. P., A robust self-tuning scheme for PI- and PD-type fuzzy controllers, *IEEE Transactions on Fuzzy Systems*, vol. 7, p. 2-16, Feb. 1999.
- [SAD 75] Sadler D. R., *Numerical methods for nonlinear regression*, St. Lucia, University of Queensland Press, 1975.
- [WON 98] Wong C., B. Huang, J. Chen, Rule regulation of indirect adaptive fuzzy controller design, *IEE Proc.- Control theory appl.*, vol. 145, p. 513-518, Nov. 1998.
- [YOS 90] Yoshida M., Y. Tsutsumi, and T. Ishida, Gain tuning method for design of fuzzy control systems, in *Proc. Int. Conf. Fuzzy Logic Neural Networks*, Japan, p. 405-408, 1990.

Inducing Parameters of a Decision Tree for Expert System Shell McESE by Genetic Algorithm

I. Bruha and F. Franek

Dept of Computing & Software, McMaster University
Hamilton, Ont., Canada, L8S4K1
Email: {bruha | franya}@mcmaster.ca

Abstract

There exist various tools for knowledge representation, modelling, and simulation in Artificial Intelligence. We have designed and built a software tool (expert system shell) called *McESE (McMaster Expert System Environment)* that processes a set of production (decision) rules of a very general form. Such a production (decision) set can be equivalently symbolized as a decision tree.

In real life, even if the logical structure of a production system (decision tree) is provided, the knowledge engineer may be faced with the lack of knowledge of other important parameters of the tree. For instance, in our system McESE, the weights, threshold, and the certainty propagation functions – all of these are a part of the machinery handling the certainty/uncertainty of decisions – have to be designed according to a set of training (representative) events, observations, and examples.

One possible way of deriving these parameters is to employ machine learning (ML) or data mining (DM) algorithms. However, ‘traditional’ algorithms in both fields select immediate (usually local) optimal values – in the context of a whole decision set such algorithms select optimal values for each rule without regard to optimal values for the whole knowledge base. Genetic algorithms comprise a long process of evolution of a large population of objects (chromosomes) before selecting (usually global) optimal values, and so giving a ‘chance’ to weaker, worse objects, that nevertheless may prove to be optimal in the context of the whole knowledge base.

In this methodology case study, we expect that a set of McESE decision rules (or more precisely, the topology of a decision tree) is given. The paper discusses a simulation of an application of genetic algorithms to generate parameters of the given tree that can be then used in the rule-based expert system shell McESE.

Keywords: expert system shell, genetic algorithms, rule-based systems, classification, data analysis

1. Introduction

A builder of an expert system usually employs an expert system shell to design and develop a decision-support expert system for a given problem. We have designed and built a software tool (expert system shell) called *McESE (McMaster Expert System Environment)* that processes a set of production (decision) rules of a very general form allowing several means of handling uncertainty ([7], [8]). Note that such a production (decision) set can be equivalently exhibited as a decision tree.

In this study, we expect that the logical structure or the topology of a set of decision rules (a decision tree) is given. Even if this logical structure is provided, particularly in real-world tasks, the

designer may be faced with the lack of knowledge of other parameters of the tree. These parameters are usually adjustable values (either discrete or numerical ones) of production rules or other knowledge representation formalisms such as frames. In particular, in our system McESE, these are represented by weights and thresholds for terms and the selection of the certainty value propagation functions (CVPF for short) from a predefined set. We use the traditional approach of machine learning (ML) and data mining (DM): we adjust the above parameters according to a set of training (representative) observations (examples). However, we use a different and relatively new approach for the inductive process based on the paradigm of genetic algorithms.

Genetic algorithm (GA) encompasses a long

process of evolution of a large population of chromosomes (individuals) before selecting optimal values that have a better chance of being globally optimal compared to the traditional methods. The fundamental idea is simple: chromosomes selected according to an ‘evaluation’ are allowed to crossover so as to produce a ‘slightly different’ new one – the offspring. It is clear that the algorithm performs according to how ‘slightly different’ and ‘evaluation’ are defined.

In this paper, we present a simulation of applying GAs to generate/adjust the parameter values of a McESE decision tree.

Section 2 of this paper briefly describes our rule-based expert system shell McESE with emphasis on the form of rules. Section 3 then surveys the structure of GAs. Afterwards, Section 4 introduces the methodology of this project including a case study.

2. Rule-Based Expert System Shell McESE

McESE (McMaster Expert System Environment) [7], [8] is an interactive environment for design, creation, and execution of backward-or forward-chaining rule-based expert systems. The main objectives of the project focused on two aspects: to provide extensions of regular languages to deal with McESE rule bases and inference with them, and a versatile machinery to deal with uncertainty.

The language extension is facilitated through a set of functions with the native syntax that provide the full functionality required (for instance, in the Common-Lisp extension these are Common-Lisp functions callable both in the interactive or compiled mode, in the C extension, these are C functions callable in any C program).

The versatility of the treatment of uncertainty is facilitated by the design of McESE rules utilizing weights, threshold directives, and CVPF's (*Certainty Value Propagation Function*). The McESE rule has the following syntax:

$$R: T_1 \& T_2 \& \dots \& T_n =F=> T$$

T_1, \dots, T_n are the left-hand side terms of the rule R and T is the right-hand side term of the rule R . A term has the form:

$$weight * predicate [op cvalue]$$

where *weight* is an explicit certainty value, *predicate* is a possibly negated (by \sim or $-$) predicate possibly with variables, and *op cvalue* is the threshold directive (*op* can either be $>$, $>=$, $<$, or $<=$, and *cvalue* is an explicit certainty value). If the weight is omitted it is assumed to be 1 by default. The threshold directive can also be omitted. The certainty values are reals in the range 0..1 .

The value of a term depends on the current value of the predicate for the particular instantiation of its variables; if the threshold directive is used, the value becomes 0 (if the current value of the predicate does not satisfy the directive), or 1 (if it does). The resulting value of the term is then the value of the predicate modified by the threshold directive and multiplied by the weight.

In McESE in the backward-chaining mode, each rule that has the predicate being evaluated as its right-hand side predicate is eligible to fire. The firing of a McESE rule consists of instantiating the variables of the left-hand side predicates by the instances of the variables of the right-hand side predicate, evaluating all the left-hand side terms and assigning the new certainty value to the predicate of the right-hand side term (for the given instantiation of variables). The value is computed by the CVPF F based on the values of the terms T_1, \dots, T_n . In simplified terms, the certainty of the evaluation of the left-hand side terms determines the certainty of the right-hand side predicate. There are several built-in CVPF's the user can use (*min*, *max*, *average*, *weighted average*), or the user can provide his/her own custom-made CVPF's. This approach allows, for instance, to create expert systems with fuzzy logic, or Bayesian logic, or many others (see [13]).

Any rule-based expert system must deal with the problem of which of the eligible rules should be ‘fired’. This is dealt with by what is commonly referred to as *conflict resolution*. In McESE the problem is slightly different; each rule is fired and it provides an evaluation of the right-hand predicate – and we face the problem which of the evaluation should be used. McESE provides the user with three predefined conflict resolution strategies: *min* (where one of the rules leading to the minimal certainty value is considered fired), *max* (where one of the

rules leading to the maximal certainty value is considered fired), and *rand* (a randomly chosen rule is considered fired). The user has the option to use his/her own conflict resolution strategy as well.

3. Genetic Algorithms: A Survey

The induction of concepts from databases consists of searching usually a large space of possible concept descriptions. There exist several paradigms how to control this search, for instance various statistical methods, logical/symbolic algorithms, neural nets, and the like. However, such traditional algorithms select immediate (usually local) optimal values.

On the other hand, the *genetic algorithms* (GAs) comprise a long process of evolution of a large population of individuals (objects, chromosomes) before selecting optimal values, thus giving a 'chance' to weaker, worse objects. They exhibit two important characteristics: the search is usually global and parallel in nature since a GA processes not just a single individual but a large set (population) of individuals.

Genetic algorithms emulate biological evolution and are utilized in optimization processes. The optimization is performed by processing a population of individuals (chromosomes). A designer of a GA has to provide an evaluation function, called *fitness*, that evaluates any individual. The fitter individual is given a greater chance to participate in forming of the new generation. Given an initial population of individuals, a genetic algorithm proceeds by choosing individuals to become parents and then replacing members of the current population by the new individuals (offsprings) that are modified copies of their parents. This process of reproduction and population replacement continues until a specified stop condition is satisfied or the predefined amount of time is exhausted.

Genetic algorithms exploit several so-called *genetic operators*:

- *Selection* operator chooses individuals (chromosomes) as parents depending on their fitness; the fitter individuals have on average more children (offsprings) than the less fit ones. Selecting the fittest individuals tends to improve the population.

- *Crossover* operator creates offsprings by combining the information involved in the parents.
- *Mutation* causes the offsprings to differ from their parents by introducing a localized change.

Details of the theory of genetic algorithms may be found in several books, e.g. [9], [11]. There are many papers and projects concerning genetic algorithms and their incorporation into data mining [1], [6], [10], [12], [14].

We now briefly describe the performance of the genetic algorithm we have designed and implemented for general purposes, including this project. The foundation for our algorithms is the CN4 learning algorithm [2], a significant extension of the well-known algorithm CN2 [4], [5]. For our new learning algorithm (*genetic learner*) *GA-CN4*, we removed the original search section from the inductive algorithm and replaced it by a domain-independent genetic algorithm working with fixed-length chromosomes.

The learning starts with an initial population of individuals (chromosomes) and lets them evolve by combining them by means of genetic operators. More precisely, its high-level logic can be described as follows:

procedure GA

Initialize randomly a new population

Until stop condition is satisfied **do**

1. Select individuals by the tournament selection operator
2. Generate offsprings by the two-point crossover operator
3. Perform the bit mutation
4. Check whether each new individual has the correct value (depending on the type of the task); if not the individual's fitness is set to 0 (i.e., to the worst value)

enddo

Select the fittest individual

If this individual is statistically significant **then**
return it

else return nil

The above algorithm mentions some particular operations used in our GA. Their detailed description can be found e.g. in [9], [11], or [3].

More specifically, the generation mode of replacing a population is used. The fitness function is derived from the Laplacian evaluation formula. The default parameter values in our genetic algorithm: size of population is 30, probability of mutation $P_{mut} = 0.002$. The genetic algorithm stops the search when the Laplacian criterion does not improve after 10000 generations.

Our GA also includes a check for statistical significance of the fittest individual. It has to comply with the statistical characteristics of a database which is used for training; the P^2 -statistics is used for this test of conformity. If no fittest individual can be found, or it does not comply with the P^2 -statistic, then *nil* is returned in order to stop further search; the details can be found in [2].

4. Case Study

In our project, an individual is formed by a fixed-length list (array) of the following parameters of the McESE system:

- the *weight* of each term of McESE rule,
- the threshold value *cvalue* of each term,
- the selection of the CVPF of each rule from a predefined set of CVPF's
- the conflict resolution for the entire decision tree.

Since our GA-CN4 is able to process numerical (continuous) attributes, the above parameters *weight* and *cvalue* can be properly handled. As for the CVPF, it is considered as a discrete attribute with these singular values (as mentioned above): *min*, *max*, *average*, and *weighed average*. Similarly, the conflict resolution is treated as a discrete attribute.

Since the list of the above parameters is of the fixed size, we can apply the GA-CN4 algorithm that can process the fixed-length chromosomes (objects) only.

The entire process of deriving the right values of the above parameters (*weights*, *cvalues*, CVPF's, conflict resolution) looks as follows:

1. A dataset of typical (representative) examples for a given task is selected (usually by a knowledge engineer that is to solve a given task).

2. The knowledge engineer (together with a domain expert) induces the set of decision rules, i.e. the topology of the decision tree, without specifying values of the above parameters.
3. The genetic learner GA-CN4 induces the right values of the above parameters by processing the training database.

To illustrate our new methodology of knowledge acquisition we introduce the following case study. We consider a very simple task of heating and mixing three liquids L_1 , L_2 , and L_3 . The first two have to be controlled by their flow and temperature; then they are mixed with L_3 . Thus, we can derive these four rules:

$$\begin{aligned}
 R_1: & w_{11} * F_1 [\geq c_{11}] \& w_{12} * T_1 [\geq c_{12}] = f_1 \Rightarrow H_1 \\
 R_2: & w_{21} * F_2 [\geq c_{21}] \& w_{22} * T_2 [\geq c_{22}] = f_2 \Rightarrow H_2 \\
 R_3: & w_{31} * H_1 [\geq c_{31}] \& w_{32} * F_1 [\geq c_{32}] \& \\
 & w_{33} * H_2 [\geq c_{33}] \& w_{34} * F_3 [\geq c_{34}] = f_3 \Rightarrow A_1 \\
 R_4: & w_{41} * H_2 [\geq c_{41}] \& w_{42} * F_2 [\geq c_{42}] \& \\
 & w_{43} * H_1 [\geq c_{43}] \& w_{44} * F_3 [\geq c_{44}] = f_4 \Rightarrow A_2
 \end{aligned}$$

Here F_i is the flow of L_i , T_i its temperature, H_i the resulting mix, A_i the adjusted mix, $i=1, 2$ (or 3). The corresponding decision tree is on Fig. 1.

We assume that the above topology of the decision tree (without the right values of its parameters) was derived by the knowledge engineer. The unknown parameters w_{ij} , c_{ij} , f_i , including the conflict resolution then form a chromosome (individual) of length 29 attributes. The global optimal value of this chromosome is then induced by the genetic algorithm GA-CN4.

5. Analysis and Future Research

The primary aim of this project was to design a new methodology for inducing parameters for an expert system under the condition that the topology (the decision tree) is known. We have selected domain-independent genetic algorithm that searches for a global optimizing parameters values.

Our analysis of the methodology indicates that it is quite a viable one. The traditional algorithms explore a small number of hypotheses at a time, whereas the genetic algorithm carries out a parallel search within a robust population. The only disadvantage our study found concerns the time complexity. Our genetic learner is about 20 times

slower than the traditional machine learning algorithms.

In the near future, we are going to implement the entire system discussed here and compare it with other inductive data mining tools. The McESE system will thus comprise another tool for rule-based knowledge processing (besides neural net and Petri nets) [8].

The algorithm GA-CN4 h is written in C and runs under both Unix and Windows. The McESE system has been implemented both in C and Lisp.

References

- [1] J. Bala et al., *Hybrid learning using genetic algorithms and decision trees for pattern classification*, Proc. IJCAI-95 (1995), 719-724.
- [2] I. Bruha, S. Kockova, *A support for decision making: Cost-sensitive learning system*, Artificial Intelligence in Medicine, 6 (1994), 67-82.
- [3] I. Bruha, P. Kralik, P. Berka, *Genetic learner: Discretization and fuzzification of numerical attributes*, Intelligent Data Analysis J., 4 (2000), 445-460
- [4] P. Clark, R. Boswell, *Rule induction with CN2: Some recent improvements*, EWSL-91, Porto, Springer-Verlag (1991), 151-163.
- [5] P. Clark, T. Niblett, *The CN2 induction algorithm*, Machine Learning, 3 (1989), 261-283.
- [6] K.A. De Jong, W.M. Spears, D.F. Gordon, *Using genetic algorithms for concept learning*. Machine Learning, 13, Kluwer Academic Publ. (1993), 161-188.
- [7] F. Franek, *McESE-FranzLISP: McMaster Expert System Extension of FranzLisp*, in: Computing and Information, North-Holland, 1989
- [8] F. Franek, I. Bruha, *An environment for extending conventional programming languages to build expert system applications*, Proc. IASTED Conf.

Expert Systems, Zurich, 1989

- [9] D.E. Goldberg, *Genetic algorithms in search, optimization, and machine learning*, Addison-Wesley (1989).
- [10] A. Giordana, L. Saitta, *REGAL: An integrated system for learning relations using genetic algorithms*, Proc. 2nd International Workshop Multistrategy Learning (1993), 234-249.
- [11] J. Holland, *Adaptation in natural and artificial systems*, Univ. of Michigan Press, Ann Arbor (1975).
- [12] C.Z. Janikow, *A knowledge-intensive genetic algorithm for supervised learning*, Machine Learning, 5, Kluwer Academic Publ. (1993), 189-228.
- [13] Z. Jaffer, *Different treatments of uncertainty in McESE*, MSc. Thesis, Dept Computer Science & Systems, McMaster University (1990)
- [14] P.D. Turney, *Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm*, J. Artificial Intelligence Research (1995).

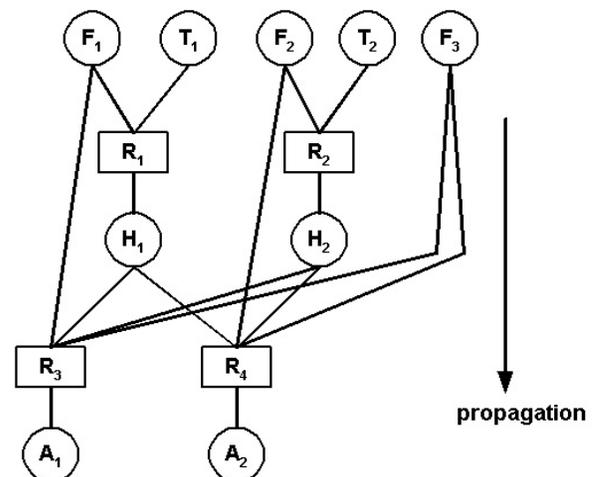


Fig. 1. The decision tree of our case study.

SOMA APPLIED TO OPTIMUM WORK ROLL PROFILE SELECTION IN THE HOT ROLLING OF WIDE STEEL

LARS NOLLE¹ and IVAN ZELINKA²

¹*School of Computing and Mathematics
The Nottingham Trent University
Burton Street, Nottingham, NG1 4BU, UK
lars.nolle@ntu.ac.uk*

²*Institute of Information Technologies, Faculty of Technology
Tomas Bata University in Zlin
Mostni 5139Zlin, CZ
zelinka@ft.utb.cy*

Abstract: The quality of steel strip produced in a wide strip rolling mill depends heavily on the careful selection of initial ground work roll profiles for each of the mill stands in the finishing train. In the past, these profiles were determined by human experts, based on their knowledge and experience. In this research, a Self-Organising Migration Algorithm (SOMA), a heuristic optimisation algorithm, has been used to find optimum profiles for a simulated rolling mill. The resulting strip quality produced using the profiles found by the optimisation algorithm and the quality produced using the original specifications are compared. The best set of profiles found by SOMA clearly outperformed the original set.

Keywords: hot strip, rolling, roll profiles, optimisation, SOMA

1 INTRODUCTION

There is a worldwide overcapacity for wide steel strip. In such a “buyers’ market”, producers need to offer a high quality product at a competitive price in order to retain existing customers and win new ones. Producers are under pressure to improve their productivity by automating as many tasks as possible and by optimising process parameters to maximise efficiency and quality. One of the most critical processes is the hot rolling of the steel strip [1].

2 HOT ROLLING OF WIDE STRIP

In a rolling mill a steel slab is reduced in thickness by rolling between two driven work rolls in a mill stand (Figure 1). To a first approximation, the mass flow and the width can be treated as constant. The velocity of the outgoing strip depends on the amount of reduction. A typical hot rolling mill finishing train might have as many as 7 or 8 close-coupled stands.

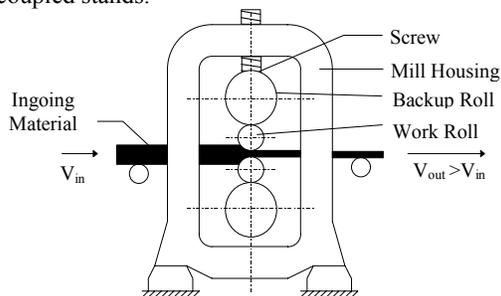


Figure 1 –Layout of a 4-high rolling stand.

2.1 Mill Train

A hot-rolling mill train transforms steel slabs into flat strip by reducing the thickness, from some 200 millimetres to some two millimetres. Figure 2 shows a typical hot strip mill train, consisting of a roughing mill (stands R1-R2) and finishing stands (F1-F7).

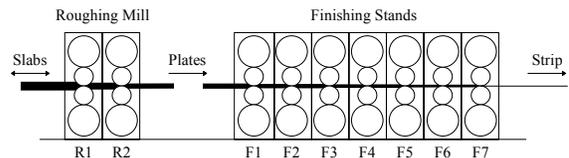


Figure 2 –Typical hot strip mill train.

The roughing mill usually comprises one or more stands which may operate in some plants as a reversing mill, i.e. the slabs are reduced in thickness in several passes by going through the stand(s) in both directions. When the slab or plate has reached the desired thickness of approximately 35 mm it is rolled by the “close-coupled” finishing stands in one pass. Strip dimensions, metallurgical composition, and the number of slabs to be rolled, together with other process dependent variables, are known as a *rolling program* or *rolling schedule*.

Within a rolling program, the width of the strip changes from wide at the beginning to narrow towards the end, because the edges of the strip damage the rolls. These damaged areas must not be

in contact with the strip and therefore, only strip with a reduced width can be rolled at that point.

2.2 Strip Quality

The main discriminator for steel strip from different manufacturers is quality, which has two aspects: *strip profile* and *strip flatness*.

Strip profile is defined as variation in thickness across the width of the strip. It is usually quantified by a single value, the *crown*, defined as the difference in thickness between the centre line and a line at least 40 mm away from the edge of the strip (European Standard EN 10 051). Positive values represent convex strip profiles and negative values concave profiles. For satisfactory tracking during subsequent cold rolling a convex strip camber of about 0.5% - 2.5% of the final strip thickness is required [2]. Flatness - or the degree of planarity - is quantified in *I-Units*, smaller values of I-Units representing better flatness.

Modern steelmaking techniques and the subsequent working and heat treatment of the rolled strip usually afford close control of the mechanical properties and geometrical dimensions. In selecting a supplier, customers rank profile and flatness as major quality discriminators. Tolerances on dimensions and profile of continuous hot-rolled uncoated steel plate, sheet and strip are also defined in European Standard EN 10 051.

Both the flatness and profile of outgoing strip depend crucially on the geometry of the loaded gap between top and bottom work rolls. As a consequence of the high forces employed, the work rolls bend during the rolling process, despite being supported by larger diameter back-up rolls [3]. Figure 3 shows a pair of cylindrical work rolls. In Figure 4 the effects of the loading can be seen. Due to contact with the strip at temperatures between 800°C and 1200°C the rolls expand, despite being continuously cooled during the rolling operation. Figure 5 shows the effect of thermal expansion of the unloaded work rolls on the roll gap.

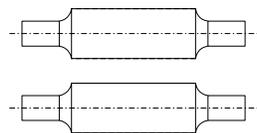


Figure 3 –Unloaded rolls.

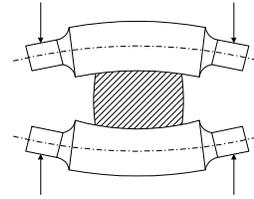


Figure 4 –Loaded cold rolls.

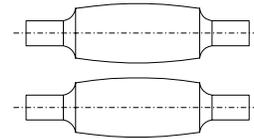


Figure 5 –Unloaded hot rolls.

If the geometry of the roll gap does not match that of the in-going strip, the extra material has to flow towards the sides (Figure 6). If the thickness becomes less than about 8mm, this flow across the width cannot take place any longer and will result in partial extra strip length, and therewith in a wavy surface (Figure 7).

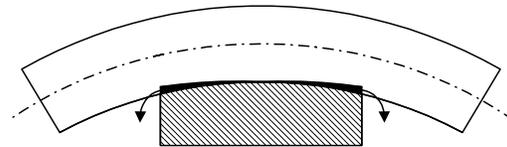


Figure 6 –Mismatch between roll gap and strip geometry.

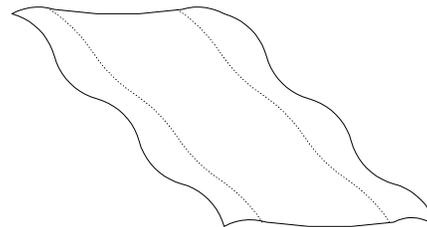


Figure 7 –Wavy strip surface.

The effects of bending and thermal expansion on the roll gaps, and the strip tension between adjacent mill stands, results in a non-uniform distribution of the internal stress over the width of the strip. This can produce either latent or manifest bad shape, depending on the magnitude of the applied tension and the strip thickness [4]. Bad shape, latent or manifest, is unacceptable to customers, because it can cause problems in further manufacturing processes.

2.3 Initially Ground Work Roll Profiles

To compensate for the predicted bending and thermal expansion, work rolls are ground to a convex or concave camber, which is usually sinusoidal in shape (Figure 8).

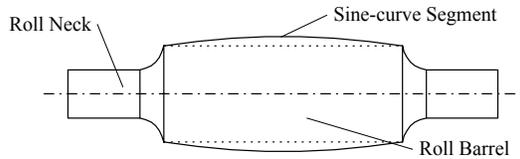


Figure 8 – *Cambered work roll.*

Figure 9 shows how the initially ground camber can compensate for the combined effects of bending and expansion.

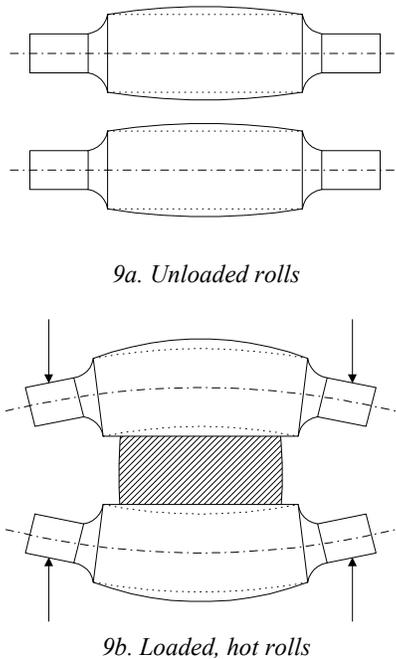


Figure 9 – *Compensating combined effects.*

Due to the abrasive nature of the oxide scale on the strip, the rolls also wear significantly. Due to this roll wear, the rolls need to be periodically reground after a specified duty cycle (normally about four hours), to re-establish the specified profile.

2.4 Roll Profile Specification

The challenge is to find suitable work roll profiles - for each rolling program - capable of producing strip flatness and profile to specified tolerances. In a new mill, these profiles are initially specified individually for every single roll program. These are often later changed, e.g. by the rolling mill technical personnel in an effort to establish optimum profiles! This fine-tuning of the roll profiles is nearly always carried out empirically.

Due to the lack of accurate model equations and auxiliary information, like derivatives of the transfer function of the mill train, traditional calculus-based optimisation methods cannot be

applied. If a new rolling program is to be introduced, it is a far from straightforward task to select the optimum work roll profiles for each of the stands involved.

3 OPTIMISATION OF PROFILES

The seemingly obvious solution of experimenting with different profiles in an empirical way is not acceptable because of financial reasons - the earning capacity of a modern hot strip mill is thousands of pounds per minute, and the mills are usually operated 24 hours a day. Any unscheduled interruption of strip production leads to considerable financial loss. The use of unsuitable roll profiles can seriously damage the mill train. The approach chosen in this research is to simulate the mill and then apply experimental optimisation algorithms. Figure 10 shows the closed optimisation loop, containing the mill model and an optimisation algorithm.

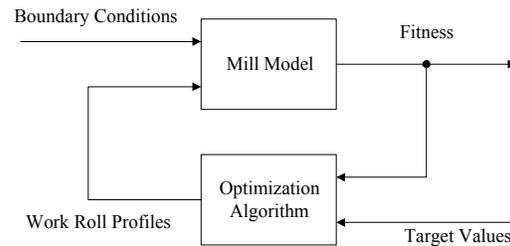


Figure 10 – *Optimisation loop.*

A finite constant volume elements model was used, which was developed in previous research. The accuracy of the model was increased by using real world data to train an Artificial Neural Network to compensate for the model error [5][6].

3.1 The Fitness Function

In the past, a number of optimisation algorithms were used to find optimum profiles for a single steel slab [5]. However, in the real world, a sequence of different slabs is rolled with the same set of profiles (see 2.1). Therefore, the profiles need to be suitable for each of the different slabs in the same rolling program. This has been taken into consideration in this research by adjusting the fitness function used to measure the fitness of a set of profiles.

The fitness (objective function) has been calculated by a combination of crown and flatness values of the centre-line, the edge, and the quarter-line (Equation 1). To avoid a division by zero, one been added to the denominator. The theoretical maximum value of this objective function is 1.0.

$$f(x, \alpha) = \frac{1}{n} \sum_{s=1}^n \frac{1}{1 + \frac{1}{\alpha} \sum_{i=1}^3 I_i(x) + |c_{aim} - c(x)|} \quad (1)$$

where:

- n : number of different slabs in rolling program
- $f(x)$: fitness of solution x ,
- $I_i(x)$: I-Units at line i for solution x ,
- c_{aim} : target crown,
- $c(x)$: achieved crown for solution x ,
- α : constant to select the relative contribution of flatness and camber, chosen to be 5000 for the experiments.

As it can be seen from Equation 1, the fitness for the rolling program is the average fitness for each of the different slabs rolled during the program.

3.2 Optimization Algorithm Used

In recent years, a broad class of optimisation algorithms has been developed for stochastic optimisation, i.e. for optimising systems where the functional relationship between the independent input variables x and output (objective function) y of a system S is not known. Using stochastic optimisation algorithms such as Genetic Algorithms (GA), Simulated Annealing (SA) and Differential Evolution (DE), a system is confronted with a random input vector and its response is measured. This response is then used by the optimisation algorithm to tune the input vector in such a way that the system produces the desired output or target value in an iterative process.

The following section describes the Self-Organising Migration Algorithm (SOMA). SOMA is a stochastic optimisation algorithm that is modelled after the social behaviour of co-operating individuals [7]. It was chosen because it was proven that the algorithm has the ability to converge towards the global optimum [8].

SOMA is a stochastic optimisation algorithm that works on a population of candidate solutions in loops - so called *migration loops*. The population is initialised randomly at the beginning of the search. In each loop, the population is evaluated and the solution with the highest fitness becomes the leader L (Figure 11). Apart from the leader, in one migration loop, all individuals will traverse over the input space into direction of the leader (Figure 12):

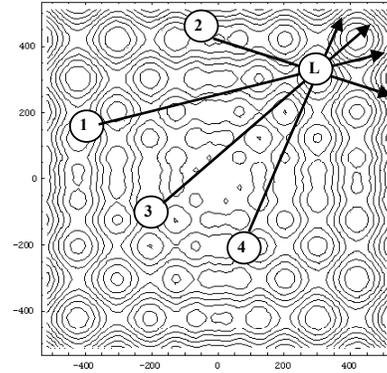


Figure 11 – 2D example: positions of individual before migrating.

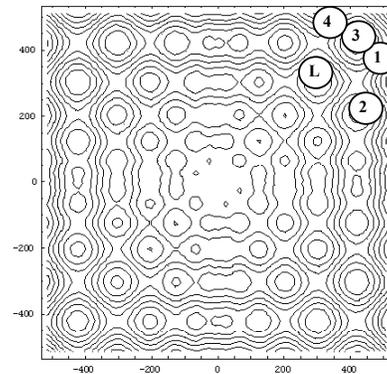


Figure 12 – 2D example: positions of individuals after migration loop.

An individual will travel a certain distance (called Path Length) towards the leader in n steps of defined length. If the path length is chosen to be greater than one, then the individual will actually over shot the leader. This path is perturbed randomly.

3.2.1 Perturbation

Mutation, the random perturbation of individuals, is an important operation for evolutionary strategies (ES). It ensures the diversity amongst the individuals and it also provides the means to restore lost information in a population. Mutation in SOMA is different compared to other ES strategies. SOMA uses a PRT parameter to achieve perturbation. This parameter has the same effect for SOMA as mutation has for GA. It is defined in the range $[0, 1]$ and is used to create a perturbation vector (PRTVector) as follows:

$$\text{if } \text{rnd}_j < \text{PRT} \text{ then } \text{PRTVector}_j = 1 \text{ else } 0, \quad j = 1, \dots, n_{\text{param}} \quad (2)$$

The novelty of this approach is that the PRTVector is created before an individual starts its journey

over the search space. The PRTVector defines the final movement of an active individual in search space.

The randomly generated binary perturbation vector controls the allowed dimensions for an individual. If an element of the perturbation vector is set to zero, then the individual is not allowed to change its position in the corresponding dimension.

Figure 13 shows an example of a candidate solution *Individual I* that would make a number of steps towards *Leader L* without perturbation. With the perturbation vector [0,1] it is only allowed to move in *y* direction.

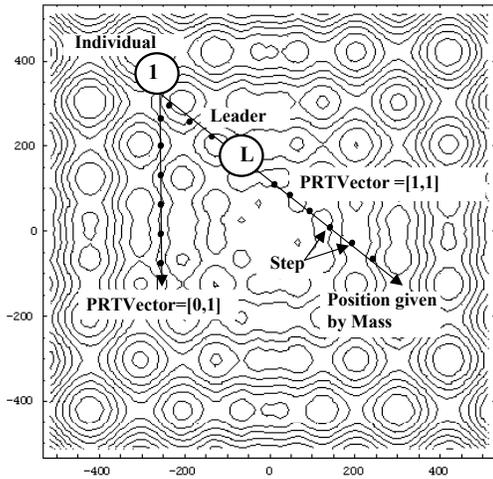


Figure 13 –Perturbation in SOMA.

3.2.2 Generating New Candidate Solutions

In standard ES the *Crossover* operator usually creates new individuals based on information from the previous generation. Geometrically speaking, new positions are selected from an *N* dimensional hyper-plane. In SOMA, which is based on the simulation of cooperative behaviour of intelligent beings, sequences of new positions in the *N* dimensional hyper-plane are generated. They can be thought of as a series of new individuals obtained by the special crossover operation. This crossover operation determines the behaviour of SOMA. The movement of an individual is thus given as follows:

$$\vec{r} = \vec{r}_0 + m t PRTVector \quad (3)$$

where:

- \vec{r} : new candidate solution
- \vec{r}_0 : original individual
- m : difference between leader and start position of individual
- t : $\in [0, \text{Path length}]$
- $PRTVector$: control vector for perturbation

It can be observed from Eq. (3) that the PRTVector causes an individual to move toward the leading individual (the one with the best fitness) in *N-k* dimensional space. If all *N* elements of the PRTVector are set to 1, then the search process is carried out in an *N* dimensional hyper-plane (i.e. on a *N+1* fitness landscape). If some elements of the PRTVector are set to 0 then the second terms on the right hand side of equation equal 0. This means those parameters of an individual that are related to 0 in the PRTVector are 'frozen', i.e. not changed during the search. The number of frozen parameters "*k*" is simply the number of dimensions which are not taking part in the actual search process. Therefore, the search process takes place in a *N-k* dimensional subspace.

4 EXPERIMENTAL RESULTS

SOMA has been applied 50 times in order to find the optimum set of profiles. In the rolling program there were 14 different slabs, therefore the average fitness for this 14 slabs had to be calculated.

The control parameter settings have been found empirically: 40 migration loops were carried out by 20 individuals. The path length was chosen to be 2.0, the step size was 0.31 and PRT was 0.1.

From Table 1 it can be seen that the average fitness achieved during the experiments was 0.96499526 out of 1.0. The small standard deviation indicates that in most of the searches the same optimum has been found, i.e. the algorithm has converged towards the global optimum. The algorithm needed on average 4418 fitness evaluations until it reached that optimum.

	Fitness	Fitness Evaluations
Average	0.96499526	4417.7
Standard Deviation	0.000304117	164.3509498

Table 1 – Search results.

Table 2 shows the strip quality achieved using the original specification, Table 3 shows the strip quality achieved using the best set of profiles found by SOMA during the experiments.

	Average	Standard Deviation
Crown error [mm]	0.06363165	0.03021786
Flatness edge [I-Units]	13.23412214	14.93880901
Flatness quarter [I-Units]	32.64022143	40.98805286
Flatness middle [I-Units]	22.2865	52.16940555

Table 2 – Strip quality with original profiles.

	Average	Standard Deviation
Crown error [mm]	0.023995157	0.026874573
Flatness edge [I-Units]	2.510596429	7.042907048
Flatness quarter [I-Units]	29.20729071	41.96476839
Flatness middle [I-Units]	26.86778571	53.90688005

Table 3 – Strip quality with best solution found by SOMA.

Table 4 shows the improvement achieved by using the optimised set of profiles. It can be seen that the average crown error was reduced dramatically by 62.3% and the corresponding standard deviation by 11.1%. The strip flatness at the edges was improved by 81.0 %, the flatness in the quarter line by 10.5%. Only the average flatness in the middle of the slabs has decreased by 20.6%.

	Average [%]	Standard Deviation [%]
Crown error	62.3	11.1
Flatness edge	81.0	52.9
Flatness quarter	10.5	-2.4
Flatness middle	-20.6	-3.3

Table 4 –Improvement of strip quality.

5 CONCLUSIONS

In this research, a Self-Organising Migration Algorithm (SOMA), a heuristic optimisation algorithm, has been used to find optimum profiles for a simulated rolling mill. The profiles were not only optimised for one particular slab, but for a whole rolling program, which is required for a real rolling mill.

The resulting strip quality produced using the profiles found by the optimisation algorithm and the quality produced using the original specifications were compared. It was shown that the best set of profiles found by SOMA clearly outperformed that of the original set. The average percentage improvement for crown error and fitness values is 33.3% compared to the original values. Therefore, SOMA has been applied successfully to the optimisation problem described in the paper.

In future work, the performance of other optimisation algorithms will be compared with that of SOMA in this problem domain.

Biography



Lars Nolle graduated from the University of Applied Science and Arts in Hanover in 1995 with a degree in Computer Science and Electronics. After receiving his PhD in Applied Computational Intelligence from The Open University, he worked as a System Engineer for EDS. He returned to The Open University as a Research Fellow in 2000. He joined The Nottingham Trent University as a Senior Lecturer in Computing in February 2002. His research interests include: applied computational intelligence, distributed systems, expert systems, optimisation and control of technical processes.

References

- [1] Larke, E. C.: The Rolling of Strip Sheet and Plate, Chapman and Hall Ltd, London, 1963
- [2] Winkler, W.: *Grundlagen des Breitbandwalzens*, Stahl u. Eisen 63 (1943) Nr. 40, pp 731-735
- [3] Emicke, Lucas: *Einflüsse auf die Walzgenauigkeit beim Warmwalzen von Blechen und Bändern*, Neue Hütte, 1. Jg. Heft 5, 1956, pp 257-274
- [4] Wilms, Vogtmann, Klöckner, Beisemann, Rohde: *Steuerung von Profil und Planheit in Warmbreitbandstraßen*, Stahl u. Eisen 105 (1985) Nr. 22, pp 1181-1190
- [5] Nolle, L., Armstrong, D.A., Hopgood, A.A., Ware, J.A.: *Optimum Work Roll Profile Selection in the Hot Rolling of Wide Steel Strip Using Computational Intelligence*, Lecture Notes in Computer Science, Vol. 1625, Springer, 1999, pp 435-452
- [6] Nolle, L., Armstrong, D.A., Hopgood, A.A., Ware, J.A.: *Simulated Annealing and Genetic Algorithms applied to Finishing Mill Optimisation for Hot Rolling of Wide Steel Strip*, International Journal of Knowledge-Based Intelligent Engineering Systems, Volume 6, Number 2, April 2002, pp 104-111
- [7] Zelinka, I., Lampinen, J., Nolle, L.: *SOMA - Self-Organizing Migration Algorithm*, Folia Fac. Sci. Nat. Univ. Mathematica, 11 (2002), pp 301-332
- [8] Zelinka, I., Lampinen, J., Nolle, L.: *On the Theoretical Proof of Convergence for a Class of SOMA Search Algorithms*, Proceedings of the 7th International MENDEL Conference on Soft Computing, Brno, CZ, 6-8 June 2001, pp 103-110

ON SOME PROPERTIES OF ARTIFICIAL FORAGING ANT COMMUNITIES

JUAN DE LARA, MANUEL ALFONSECA

*Dept. Ingeniería Informática, Universidad Autónoma de Madrid
Ctra. De Colmenar, km. 15, 28049 Madrid, Spain
e-mail: {Juan.Lara, Manuel.Alfonseca}@ii.uam.es*

Abstract: This paper describes the modelling, analysis and simulation of artificial foraging ant communities. Each virtual ant (*vant*) taking part in the simulation is modelled as an agent. The objective of each *vant* is to collect food for their community. Our aim is to study the communication flow in the community (and not to be biologically realistic). In this way, exchange of information can occur between colliding *vants*. We have used simulation and mathematical analysis to study different situations, such as low memory *vants*, forgetful *vants* and dying *vants*. Several statistical properties of these systems are characterized and some emergent phenomena are observed.

keywords: Artificial Societies, Agent-Based Simulation, Random Walks, Emergence.

1. INTRODUCTION

Computer modellization of large individual communities is an active area of research. Several objectives can be pursued with this kind of simulations, such as the resolution and optimization of problems [Dorigo and Maniezzo, 1996], and the study of emergent global behaviour and social interactions [Alfonseca and de Lara 2002], [Epstein and Axtell 1996]. The phenomenon of emergence occurs when interactions between large populations of objects at one level give rise to different types of phenomena at another level.

The most common techniques for the simulation of these systems are cellular automata [Wolfram, 1994], and multi agent systems (*MAS*) [Jennings et al. 1998]. In this last methodology, the key abstraction is the autonomous agent. According to [Jennings et al. 1998], an agent is “*a computer system, situated in some environment, that is capable of flexible autonomous action in order to meet its design objectives*”. *MAS* have been used extensively in very different applications such as industrial (manufacturing, process control, etc), commercial (information management, electronic commerce, etc), and so forth. In this paper, we will focus in the use of *MAS* for the simulation of a community of virtual agents with characteristics similar to an ant ecosystem.

We call *vants* (virtual ants) to the agents in our simulation because our aim is not to simulate realistic ants, but to study different aspects of knowledge flow in the community and the relationship of such knowledge to the nest’s ability to survive. In this simulation, each *vant* will be modelled as an object. Our approach differs from others, such as:

- [Guérin et al. 1998] where agents communicate using the environment, by dropping pheromones, and very realistic simulations have been carried out. In our simulations, agents communicate directly. This is done in order to study the flow of knowledge in the community of agents.

- [Anderson et al. 1997] where the population is low (100 ants) and uses a modification of the Ollason model [Ollason 1987] of hunting by expectation. Our agents have simpler foraging behaviour, but we work with more agents and experiment with different cognitive behaviour.

The purpose of our model is to study different situations in communication exchange, such as low memory *vants*, forgetful *vants*, etc. In previous publications [Alfonseca and de Lara 2002a] [Alfonseca and de Lara 2002b] we have presented a model in which *vants* are provided with a genotype to control their behaviour (activity, talkativity, lying, etc.) In this paper, we are interested in characterizing properties of the underlying simplified model of basic agents (without genotypes or evolution) to better understand the dynamics of the more complex model.

2. THE BASIC MODEL

A *vant* community is composed of a large number of agents. In the basic model, *vants* know the position of their nest, and are able to remember the position of one food position. When two *vants* meet, they can exchange information if one of them knows where to find food. When a *vant* finds food, it takes some portion of it to its nest and returns again until the food comes to an end. All the *vants* start at the nest, located at (0,0) coordinates.

Figure 1 shows a Statechart representing the vant behaviour.

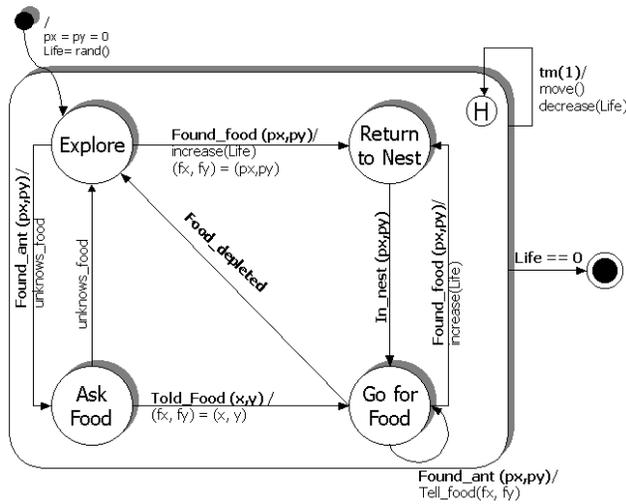


Figure 1: Behaviour of a vant.

In the first approximation to this problem, we try to characterize some of the properties of the system, such as:

- How much time does it take for a colony of vants to find food, in average? What is the minimum time?
- Does this time depend on the number of vants? how?
- How does the food knowledge propagate between vants? How does the communication between vants affect the knowledge of the community?

We will answer these questions in the following sections.

3. MINIMUM AND MEAN TIME TO REACH FOOD

When the simulation begins, all the vants act as random walkers [Berg 1983] that at each step can move to North, East, West or South. Several models have been proposed to simulate different kind of animal movements [Blackwell 1997]; we have chosen this model due to its simplicity.

Since all the vants start at the nest, at coordinates (0,0), with this kind of movement, it is not possible to reach an even cell (whose coordinates add to an even number) in an odd number of time steps. For the same reason, it is not possible for two vants to be adjacent vertically or horizontally. This situation disappears in section 2.3, where we allow let vants to be born at any time step.

Suppose the position of the food is (fx, fy), a vant needs at least |fx|+|fy| steps to reach the food. The probability for a random walker of reaching that point at time $t=|fx|+|fy|$ is $p(|fx|+|fy|, fx, fy) = t! / (4^t |fx|! |fy|!)$. In general (for $t > |fx|+|fy|$), the probability for a random walker to be at a certain position at a given time step is described by a diffusion equation, with coefficients 0.25, giving $p(t,x,y)_t - 0.25p(t,x,y)_{xx} - 0.25p(t,x,y)_{yy} = 0$. This equation can be simulated with a real-valued cellular automaton. In the automata, the probability splits in equal parts to the 4 nearest neighbours cells. The behaviour of this automaton and a finite differences [Stri89] scheme (classical one, forward differences in time) is exactly the same, as can be seen in the following equation:

$$\frac{p(t+1,r,s) - p(t,r,s)}{\Delta} = 0.25 \frac{p(t,r,s+1) - 2p(t,r,s) + p(t,r,s-1)}{\Delta x \Delta y} + 0.25 \frac{p(t,r+1,s) - 2p(t,r,s) + p(t,r-1,s)}{\Delta x \Delta y}$$

$$p(t+1,r,s) = 0.25(p(t,r,s+1) + p(t,r,s-1) + p(t,r+1,s) + p(t,r-1,s))$$

i.e. the value of the cell in the next time step is the average of the four neighbours. For the second step in the derivation, we have taken $\Delta = \Delta x = \Delta y = 1$. To solve this equation, we take as initial conditions (assuming the nest is located at (0,0)):

$$p(0,x,y) = \begin{cases} 1 & \text{if } x=0 \text{ and } y=0 \\ 0 & \text{elsewhere} \end{cases}$$

The boundaries are at infinity. The exact solution of the previous equation for $t > 0$ is:

$$p(t, x, y) = \frac{1}{\pi t} e^{-\frac{x^2+y^2}{t}}$$

But in our case, it is not useful to use this solution, because in the problem we want to simulate, each position has four neighbours (the “ground” is discretized); whereas in the exact solution the contribution to each point is done by integrating in the surrounding circle; time is also supposed to be continuous. This is not the case in our simulation, in which the time advances in a discrete manner. From now on, $p(t,x,y)$ will be solved using the solution given by the discrete approach (the cellular automaton or the finite differences scheme).

If we have N vants, the probability for at least one of them to reach (fx,fy) at time t is $1 - (1 - p(t,fx,fy))^N$. For example, if the food is located at position (10,10), we need at least 20 time steps to reach the food, the probability for one vant to find the food in 20 steps is of the order of 10^{-13} . When $t > 20$, the probability for position (10,10) to be occupied by one of the 100 vants

follows the curve in figure 2. Observe that since the sum of the x and y coordinates is even, the probability is zero for odd steps of time. It reaches its maximum at $t=200$ ($p(200,10,10)=0.11055$).

The probability of a cell being discovered for the first time exactly at $time=t$ by one or more vants is:

$$p_{full}(t, x, y) = (1 - (1 - p(t, x, y))^N) * \prod_{m=0}^{t-1} (1 - p(m, x, y))^N = \prod_{m=0}^{t-1} (1 - p(m, x, y))^N - \prod_{m=0}^t (1 - p(m, x, y))^N$$

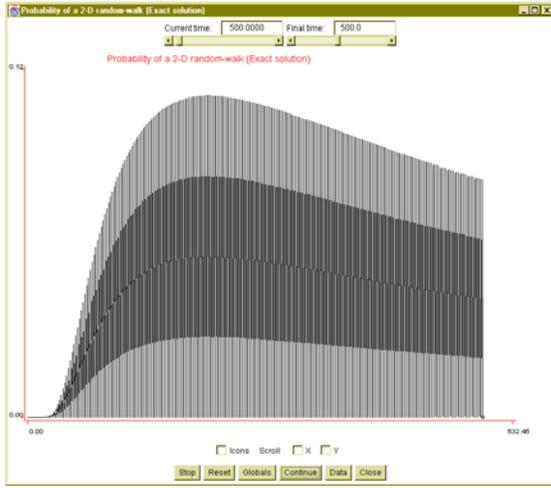


Figure 2: Probability that cell (10,10) is occupied in a random-walk of 100 walkers starting at (0,0)

The expected time for the cell (10,10) to be occupied for the first time is: $E[t] = \int_{t=0}^{\infty} t \cdot p_{full}(t,10,10)dt$

Care must be taken when calculating the previous integral, because of the sawtooth shape of function p . Experimentally, with 125 simulations and 1000 vants, the mean time to reach the food has been found to be around 48.48, with a standard deviation of 11.3. The theoretical value was 44.2. As the number of vants increases, the time to reach the food tends logarithmically to the minimum time necessary to reach the food (20), and the standard deviation decreases in the same way. This can be seen in figure 3.

4. KNOWLEDGE PROPAGATION WITHOUT INFORMATION EXCHANGE

Once food is reached, the vant remembers the position where it has been found, returns to the nest, and comes back for more food. If it collides with another vant that

does not know the food position, this vant is told the position. The question that we may ask is: how many vants will know where the food is? Before we tackle this problem, we have tried a simpler case, with no information exchange. This model will help us to prepare the more complex model in section 5.

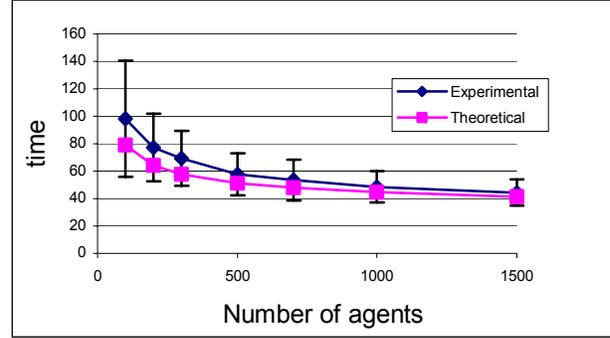


Figure 3: Average time to reach the food at (10,10) (theoretical and experimental) as the number of vants increases.

Suppose that the function returning the number of vants that know where the food is $K(t)$. Obviously $K(0) = 0$. The increment at each time step of this function is given by:

$$\Delta K(t) = (N - K(t)) \left[p(t, fx, fy) \prod_{a=0}^{t-\Delta} (1 - p(a, fx, fy)) \right]$$

Where N is the total number of vants. (fx, fy) are the coordinates of the food position. The term in square brackets represents the probability of a given vant to reach the food exactly at time t . The curve for $(fx, fy)=(0, 10)$ and $N=100$ vants can be seen in figure 4. It exhibits a fast growing at the beginning, because $N - K(t)$ is greater. When $p(t,0,10)$ decreases, the slope of $K(t)$ also decreases. The curve shows our experimental results (the average of 10 experiments, with the standard deviation).

5. KNOWLEDGE PROPAGATION WITH INFORMATION EXCHANGE

Next we have considered the case with information exchange. In this case an analytical model is too complex and we only show experimental results (see figure 5).

The results show simulations with 100 vants, with the food at a distance of 10. Note how the maximum final number of vants that know the food position is found to be around 65. The curve exhibits a logarithmic behaviour which approaches to this value. This happens because, as time increases, the vants are more dispersed, and it is more difficult for them to reach the food location and to

collide with one another. For this reason, the slope of the curve decreases.

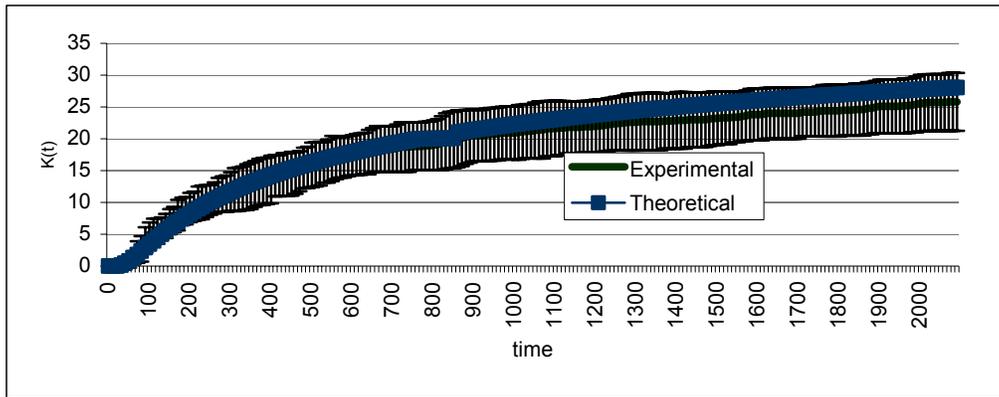


Figure 4: Vants that know the food position, without information exchange in collisions.

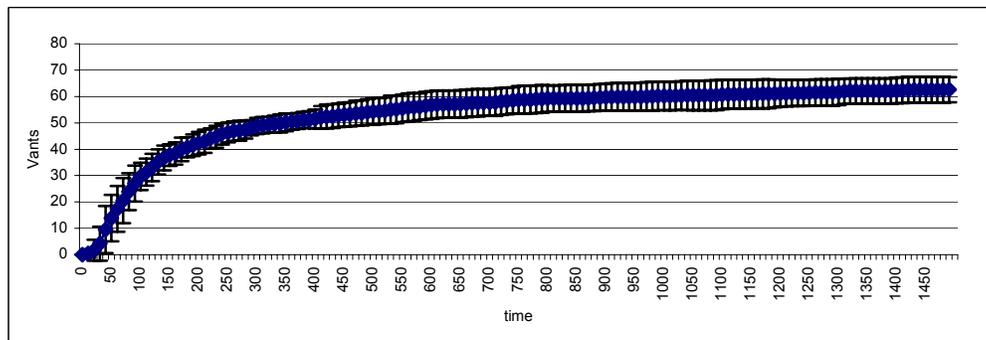


Figure 5: Vants that know the food position, with information exchange in collisions.

For distant food locations, the curve is shifted to the right, with less slope. Communication thus increases the knowledge especially in the first stages of the simulation. It can be observed that the curve in figure 5 is steeper at the beginning than the one in figure 4.

For the implementation of this simulation in OOCSSMP [Alfonseca and de Lara 2002b], each vant was represented as an OOCSSMP object. The territory, with information about the food sources is represented as another OOCSSMP object. In the main simulation loop we compute the number of vants that know where to find food. An applet with this simulation is accessible from: www.ii.uam.es/~jlara/investigacion/ecommm/otros/canti.html.

In the next sections, we will describe some situations that happen when the basic model is modified.

6. FORGETFUL AGENTS

The first variation in the previous models is the inclusion of forgetful vants. We allow vants to remember the food location only for a number of time steps. Figure 6 shows a simulation for the cases where vants can remember for

3 time steps (including the current time step), with a population of 200 vants. The figure shows the number of vants that know where to find food (to the right) together with each vant position (to the left). It can be noticed that the knowledge about the food position is not spread throughout the population. For the food distances chosen, four time steps seems to be the lower limit for a significant amount of vants to learn where the food is. For longer distances a longer memory is needed. For bigger communities, the required memory time is reduced. For example, with 500 vants, in the same conditions, three time steps are enough.

In this situation an interesting phenomenon emerges: the vants remember the position of the food by moving in groups, when a vant of the group forgets the food location, it immediately collides with another vant of the same group, that communicates it the food location. For example, in a simulation with 75 vants, with the food at (15,0), two vant groups were formed, one with 12 vants and the other with 8. The number of vants in the groups never decreased, and increased gradually. It is clear that this phenomenon emerges because vants cannot manage in another way to remember the food location.

Another interesting situation arises when we model knowledge reinforcement. Each vant encounter where both individuals know the same food position will result in a reinforcement of their knowledge (they will remember for one time step more). In this situation, a memory of two time steps (the current time step and the next) is enough to spread the knowledge of a food position to a population of 500 vants.

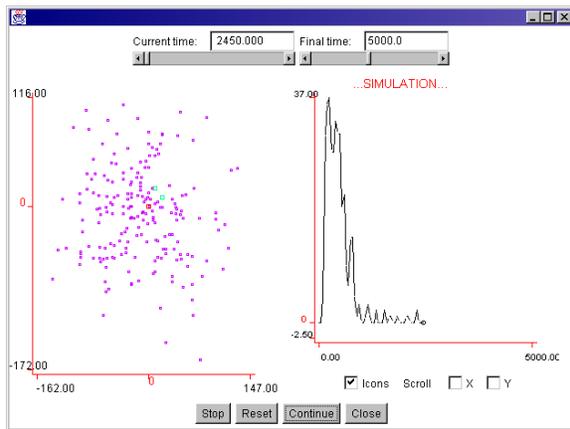


Figure 6: 200 vants with memory of 3 time steps

7. FINITE AMOUNT OF FOOD

In this simulation, food runs out, but can appear randomly somewhere else. The amount of food is also random. Agents in this simulation are not forgetful. In this situation, the knowledge curve grows more in the moments when a place with a greater amount of food is located. An interesting phenomenon emerges, similar to the spread of rumours: when the food disappears from one place, there are still vants that believe that the food is there, and this knowledge can be propagated (although it is false). When the vants realize that the food is not there, the belief curve decreases quickly.

8. VANTS THAT DIE AND ARE BORN

In these simulations, we have introduced an extension to the previous situation: vants grow old, and when they reach a certain age (predefined individually when each vant is born, and chosen randomly, between certain limits), they die. Death can be postponed when the vant eats. When a vant finds food, it takes a portion, and carries it to the nest. Once there, it leaves half of the food in the nest and eats the other. The food in the nest is used to produce new vants. In this scenario, we can control several parameters, to investigate if the community will survive or not:

- The number of food locations and the maximum and minimum amount of food per location. If these parameters are low the community always dies, if they are very high, the community always survives.

- How many time steps a vant increases its life when it eats.
- The maximum vant age.
- A plane infinite world, or a torus world (with the upper and lower borders connected, as well as the sides).
- The rate of vant birth. There are several strategies:
 - Employing all the available food to create new vants.
 - Creating vants if a food location has been found.
 - Employ a fixed percentage of food to create new vants at each time step.
- Whether the “new” vants are born knowing the last food location found or not. In the first case, we are promoting the appearance of rumours. This strategy seems a little worse than letting the new vants explore randomly: in 100 experiments with the other control parameters at the same value, the average time for extinction was about 5450, while the random exploring strategy lifetime was about 6700.

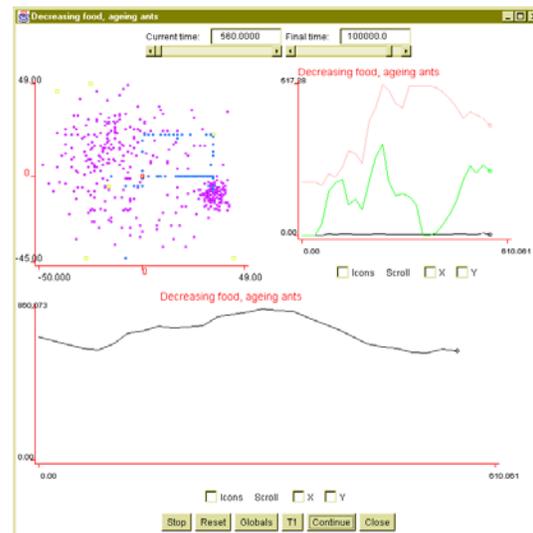


Figure 7: Simulating finite food places and dying vants

Figure 7 shows a moment in one of the simulations, with the following parameters: 700 vants in the nest initially; 6 food locations; the maximum amount of food per location is 150; a vant can live up to 650 time steps initially and 50 more time steps when it eats; Plane torus world with 2500 cells; 1/3 of the food in the nest is used to create new vants; the new vants are told where to find food. The upper left panel shows a large concentration of vants to the right. This is due to a rumour; there is a large line of vants that believe that there is food to the right, but that food location has been depleted. In the upper right panel, the dark grey line at the middle represents the community belief, the light grey at the top the amount of food in the environment, and the black at the bottom the amount of food in the nest. The lower panel shows the

number of vants. In all the simulations with these parameters, the population finally decreased to zero. In the case in figure 7, this happened at time = 5638.

9. CONCLUSIONS AND FUTURE WORK

This paper presents several simple models that simulate the behaviour of virtual ant (vant) communities. Different situations have been simulated or analyzed, such as forgetful vants, finite food, dying vants, etc. Some characteristics of the systems have been established in an analytic way, such as the minimum time to reach food, average time to reach food, knowledge propagation, etc. Using simulations, some emergent behaviour has been identified: rumours and grouping. In the forgetful vants scenario, vants form groups to be able to reach the food, this is necessary as otherwise they forget where the food is. Propagation of rumours has been observed in the situation where food is depleted from one place, but some vants are still propagating the information. In the last experiment (dying vants), comparisons between several strategies are done, and emergent behaviour due to rumours is observed. It seems that the strategy that allows the population to survive for the longest time is the one that minimizes the rumours, because, in our context, following a rumour means a waste of energy for the vants.

These simple models have helped us to better understand the more complex models presented in [Alfonseca and de Lara 2002a] and [Alfonseca and de Lara 2002b]. In those models we allow evolution and different vants' parameters are inherited (such as parameters for being communicative, sceptical, fast, liar...). Natural selection is used to determine the better combination of individual parameters to confront different situations.

We are working on an analytical model of knowledge propagation where the vants can communicate, also comparing and enriching our system with results obtained by means of other formalisms, such as cellular automata and L-Systems, although they have some limitations. For instance, it is difficult to represent individual memory. We may also use other forms of vant movement (such as the one proposed in [Blackwell 1997]), because unbiased random walks provide a very inefficient way of displacement over long distances.

Finally, this paper has an electronic and interactive version, where it is possible to experiment with the simulations, changing the number of vants, food positions, the memory length, etc., accessible from:
www.ii.uam.es/~jlara/investigacion/ecommm/otros/canti.html

ACKNOWLEDGEMENT: This paper has been sponsored by the Spanish Ministry of Science and Technology, project number TIC2002-01948.

REFERENCES

Alfonseca, M., de Lara, J. 2002. "Two level evolution of foraging agent communities". *BioSystems* Vol 66, Issues 1-2, pp.: 21-30.

Alfonseca, M., de Lara, J. 2002. "Simulating evolutionary agent colonies with OOC SMP". Proceedings of the 17th ACM Symposium on Applied Computing (SAC'2002), "AI and Computational Logic" track. pp.: 11-15. Madrid.

Anderson, C., Blackwell, P.G., Cannings, C. 1997. "Stochastic simulation of ants that forage by expectation". Fourth European Conference on Artificial Life (1997), ed. P. Husbands and I. Harvey, 531-538. MIT Press, London.

Berg, H.C. 1983. "Random walks in biology". Princeton U.P.

Blackwell, P.G. 1997. "Random Diffusion models for Animal Movement". *Ecological Modelling*, (1997) 100, 87-102

Dorigo, M., Maniezzo, V. 1996. "The Ant System: Optimization by a colony of cooperating agents". *IEEE Transactions on Systems, Man, and Cybernetics, Part-B*, Vol.26, No.1, pp. 1-13.

Guérin, S., Snyers, D., Fourcassier, V., Théraulaz, G. 1998. "Modeling ant foraging strategies by computer simulation". *Ants'98*. Brussels, Belgium.

Jennings, N.R., Sycara, K., Wooldridge, M. 1998. "A Roadmap of Agent Research and Development". *Autonomous Agents and Multi-Agent Systems*, 1, 7-38 (1998). Kluwer Academic Publishers.

Ollason, J.G. 1987. "Learning to forage in a regenerating patchy environment: can it fail to be optimal?". *Theoretical Population Biology* 31: 13-32.

Strikwerda, J.C. 1989. "Finite difference schemes and partial differential equations". Chapman & Hall; New York.

Wolfram, S. 1994. "Cellular Automata and Complexity: Collected Papers". Addison-Wesley Longman.

BIOGRAPHIES



Juan de Lara is an assistant professor at the Computer Science Department of the Universidad Autónoma in Madrid, where he teaches Software Engineering. He is a PhD in Computer Science, and works in areas such as Web based Simulation, Agent based Simulation and Multi-Paradigm

Modelling.



Manuel Alfonseca is director of the Higher Polytechnical School in the Universidad Autónoma of Madrid. From 1972 to 1994 he was Senior Technical Staff Member at the IBM Madrid Scientific Center. He works on simulation, complex systems and theoretical computer science.

MODELLING AGENTS WITH UML: AN EXAMPLE IN BUILDING SECURITY EVALUTION

JUAN DE LARA

*Dept. Ingeniería Informática, Universidad Autónoma de Madrid
Ctra. De Colmenar, km. 15, 28049 Madrid, Spain
e-mail: Juan.Lara@ii.uam.es*

Abstract: This paper proposes some extensions to UML for agent-based modelling and simulation, where the agents follow either a purely reactive or a hybrid layered approach (reactivity combined with proactivity). The extensions include notations for sensors, effectors and agent's capabilities and their specification in a formal way. In this work, agent-based simulation is also proposed as a method to help in the evaluation of security in buildings by simulating their evacuation. These simulations allow us to measure evacuation time of different scenarios, to play with different structural properties of the building and test their influence in the building security, which can be useful during its design. In these simulations, people are represented by means of agents using a hybrid layered approach. The lower layer deals with collision avoidance and doors' visualization, while the higher layer builds models of the environment (rooms' connectivity) to help the agent find the exit. Buildings are conceptually modelled as graphs where vertices and edges represent rooms and doors respectively. Rooms are discretized and represented as two-dimensional grids, in which agents can move.

keywords: Agent-Based Simulation, Hybrid Agents, UML, Building Safety, Crowd Simulation.

1. INTRODUCTION

Computer simulation is a valuable tool in situations in which experimentation with the real system is dangerous, expensive, or non-ethical. It is useful for decision-making as it enables the experimentation in multiple different scenarios in an inexpensive way. The increasing speed of today's computers is making possible the simulation of systems described by a large amount of interacting entities. For the modelling of such systems, one usually uses Cellular Automata or Agent-Based techniques [Jennings et al. 1998]. The former is more appropriate when individuals are simple and alike. The latter is more natural when individuals have more complex capabilities, such as sensors, effectors, internal complex states or reasoning abilities. Multi-Agent Systems (MAS) have been used in very different areas such as manufacturing, process control, information management and electronic commerce. In this paper, the concept of MAS is used for the modelling and simulation of a large number of individuals trying to escape from a building. Agents are modelled using a reactive layered architecture. In a purely reactive approach there is no symbolic reasoning, the agent behaviour is expressed as finite state machines [Brooks 1995]. In the approach of the present work, the behaviour is decomposed in layers dealing with reactive and pro-active behaviour.

Although there are many languages for agent-based simulation programming (such as Swarm [Swarm 2003], or OOC SMP [Alfonseca and de Lara 2002]), there is a need for higher-level, graphical, intuitive notations to

help in the modelling phase of such systems. Some emerging approaches use UML [Booch et al. 2002] extensions especially devised for the modelling of MAS [aUML, 2003]. These are notations suitable for the design of applications composed of agents, and mainly to specify interaction protocols for the application agents [Bauer et al. 2000]. In opposition, the extensions proposed in this paper are mainly useful for the modelling of *Agent-Based Simulations*. In particular, we propose extensions to model the agent sensors and effectors (to express the agent's interaction with the environment) and the agent capabilities. The extensions proposed for sensors and effectors follow the line of [aUML, 2003], for example, in [Bauer, 2001] interfaces are also used for expressing communication (in the case of an application "*interacting protocols*") between agents. In this paper we also propose the inclusion of the concept of *agent* and *classes of agents* in some of the standard UML diagrams, giving rise to *agent diagrams* (similar to *object diagrams*), *agent class diagrams* (similar to *class diagrams*) and *agent collaboration diagrams* (similar to *collaboration diagrams*). This last kind of diagram is proposed as a means to formalize some of the agent's capabilities, in a similar way as [Engels et al. 2000], but for a different purpose.

The extensions for agent-based modelling proposed in this work are illustrated by means of an example: the simulation of building evacuations. This paper proposes such agent-based simulations as an inexpensive means to help in the evaluation of building security. While

designing the building, simulations can help in architectural decisions which may affect building security, such as the placement of regular and emergency doors, and room's connectivity. In this case, simulation is the only choice, as direct experimentation (real evacuations) cannot be performed. For already existing buildings, simulations can complement evacuation simulations with real people, as they are less expensive and less annoying for the building inhabitants. Simulations make possible to experiment with different scenarios of people density in each room, with different environmental factors (fire, smoke, different degrees of visibility, etc.) or as a means to evaluate the impact of changing some security features in the building; for example, adding new emergency doors, indicators, etc.

2. EXTENDING UML FOR AGENT MODELLING

UML is becoming ever more used in the software community. For that reason, in this work the standard UML is used as much as possible, although extensions have been added to some of the diagrams and are explained in the next subsections.

2.1 Agent Class Diagrams

A class diagram is a graphical view of the static structural model. Here we propose to include *Agent Classes* in this kind of diagrams. These describe the kind of agents that exist in the system (in this paper we only consider reactive or hybrid agents). In standard UML, there is a notation for active objects (with their own thread of execution). Autonomous agents must have their own thread of execution, but they are not mere objects: they have a (partial) knowledge of their environment (by means of sensors) and may act upon it (by means of effectors). They have abilities and can be requested to perform a certain action. This is different from invoking a method on them, as the agent may refuse to perform the action. Figure 2 (Agent class *Runner*) shows the symbol we use for *Agent Classes*. There is a separate box for capabilities and another for actions. Capabilities are arranged in layers, separated by a dotted line. If capabilities or actions are not specified, the corresponding box can be omitted. We also provide symbols for the agent's sensors and effectors (similar to the ones used for interfaces, but filled in black). It is possible to connect other (Agent) classes to these black dots to mean that the agent can sense or act upon that other class. An example of the use of these symbols is given in Figures 2 and 3. Most of the times, *Agent Classes* have one or more associated Statechart diagrams specifying the agent behaviour (see Figure 1).

2.2 Agent Diagrams

A static object diagram is an instance of a class diagram, where objects and their relationships may appear. It shows a snapshot of the state of the system at a point in

time. Here we include *Agents* in this kind of diagrams. These are instances of *Agent Classes*, and are represented in a similar way (see Figure 3).

2.3 Agent Collaboration Diagrams

These diagrams show graphs of objects linked to each other, together with their communication patterns. Here we use these diagrams to specify some of the agent capabilities in a formal way. In order to specify a capability, one has to provide a number of collaboration diagrams; each one of them can be applied under different circumstances (in a similar way as graph-grammars [Engels et al. 2000] and rule-based programming). Each collaboration diagram is assigned a priority that specifies the order in which the collaboration diagram will be tried. A collaboration diagram is applicable if it is consistent with the state of the system at that moment. That is, if an homomorphism between the system's state and the collaboration diagram can be found. When the collaboration diagram is applied, the nodes and links tagged as *new* and *destroyed* are created and destroyed. Collaboration diagrams are also extended with the capacity to return values (as here they are used as a means to specify functions). An example of a capability specified in this way can be found in Figure 6.

This approach has similarities with graph grammars. These are composed by rules, each having graphs in their left and right hand sides (LHS and RHS). If a rule makes a match with a certain part of an input graph (called host graph) the rule can be applied and the zone of the graph that was matched is replaced by the RHS of the rule. In the extensions to collaboration diagrams that we propose, both LHS and RHS are collapsed into a single graph, and the nodes and links to be created and destroyed are tagged with *new* and *destroyed*. We also use *negative application conditions* in collaboration diagrams, which is a standard notation in graph grammars in the form of crossed-out elements (see Figure 6). It expresses the fact that, in order for the rule to be applied, the crossed-out elements must not be present in the matched graph. A similar idea but applied in another context, and without the possibility to return a value was proposed in [Engels et al. 2000].

An example of the use of these extensions is presented in the following section.

3. EVALUATING BUILDING SECURITY WITH AGENT-BASED SIMULATION

3.1. Single room model

This section deals with the simpler case, in which we consider evacuations of single rooms. In our model, rooms are discretized and represented as two-dimensional, rectangular grids. Each cell of the grid can

be empty, contain up to three agents, a wall or a door. Doors can be made wider by concatenating several of them. Time is discretized, in such a way that agents can move to one of the eight neighbour cells at each time step. Agents do not communicate. Once an agent sees a door, its objective is to move towards it in a straight line. If at some moment the shortest path cannot be followed – because there are many agents blocking the way – the agent moves to the least populated neighbour cell. If the agent has not seen a door before, then it moves to the most populated neighbour cell containing at most two agents (that is, he will try to “follow the crowd”). The simulation finishes when all the agents have reached a door. Figure 1 is a Statechart representing this behaviour. Some transitions in the model invoke methods (*lookAround()* and *move(when)*), which should be considered as the agent capabilities. These capabilities make use of the agent’s sensors and effectors (simulated, as we are implementing agents in software). In the case of *lookAround*, the agent is interested in visualizing either a door (which sometimes may not be visible), or the most populated place. By means of *tm(1)* we specify that at each time step, the agent should perform a certain action, depending on its current state.

The agent’s structure is shown in Figure 2. It shows an agent class (*Runner*), which has a sensor (*iVisual*) and an effector (*iLocomotion*). Special relationships (*is visible* and *can move in*) come from the classes that the sensors or effectors can act or sense. In this case, the *iVisual* sensor can sense either Doors, Walls or other Runners. The *iLocomotion* effector can act on Rooms (that is, agents can walk in the room). Agent capabilities *move* and *lookAround* are specified in the Agent class lower box. Attributes *doorX* and *doorY* are used to store the position of the door the agent is moving towards (in the case he has seen a door before). A *Runner* is situated in a room, and this is expressed with the relationship class *Position*.

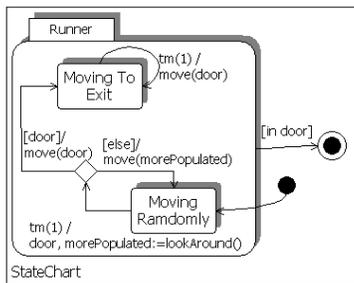


Figure 1: Behaviour of the Runner Agent.

A UML *Package* delimits the environment, which consists of a room made of several doors and walls. Doors are placed in walls (relationship “has”). The spatial dimensions of the room are stored in attributes *width* and *length*. The door coordinates are stored in

attributes *px* and *py*. The initial and final wall coordinates are stored in attributes *xinit*, *yinit* and *xend*, *yend* respectively. Only walls parallel to the X or Y axis are allowed in the model. The interaction between the environment and the agents is expressed by using the sensor/effector notation introduced in section 2.1.

Figure 3 shows an “agent diagram” that reflects the way in which an agent can sense the presence of doors or other agents. The figure shows a situation in which an agent (*r1*) is able to see another agent (*r2*) and a door. The condition for this to happen is that no other visible object must be between *r1* and *r2* or the door.

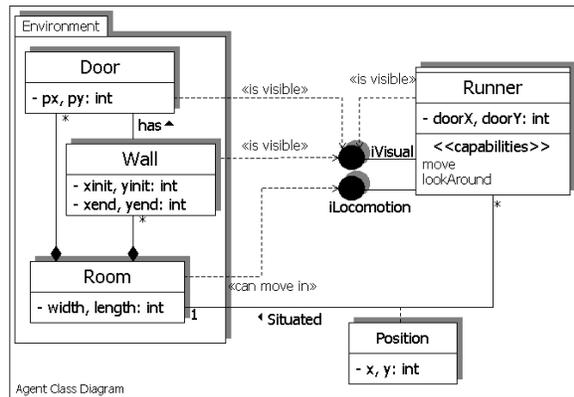


Figure 2: An Agent Class Diagram.

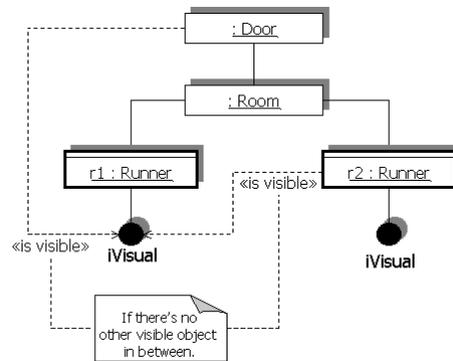


Figure 3: An Agent Diagram

The model has been programmed in C++, as efficiency in time and memory is needed, because thousands of agents will be created in the simulation. Agent programming languages are less efficient than programming directly in C++, because it allows for optimisation of the code by hand. On the contrary, agent languages provide higher-level constructs that make the programming easier. The implementation of the movement in straight line was done using the Bresenham’s line drawing algorithm. To illustrate the usefulness of this model the following subsection shows some of the experiments performed.

3.2. Experiments

Two different sets of experiments were performed to evaluate the effect of door placement in the time it takes the agents to escape from the room (of size 42x42). In the first set, four doors were placed in the room, in different configurations, each one tested with different density of agents, from 0.125 to 3 (the maximum, as in each cell at most three agents can be present at the same time). Forty experiments were performed with each room configuration and for each agent density.

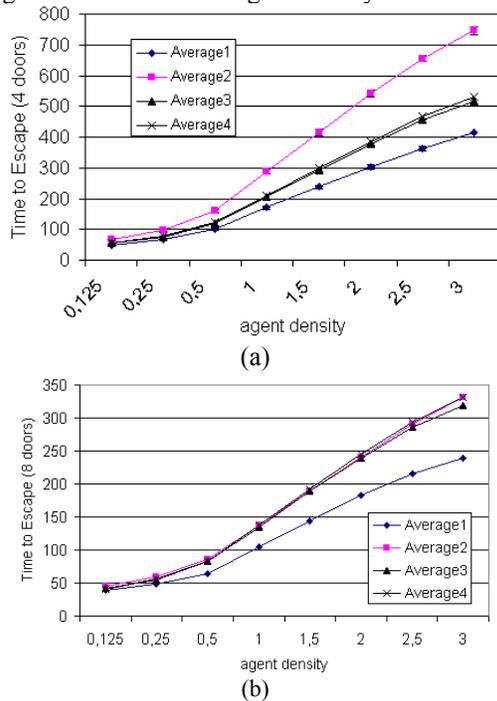


Figure 4: Time to escape with respect to agent density for different door configurations: 4(a) and 8 doors(b).

Figure 4(a) shows the results of each configuration tested for different agent densities. The X-axis is the agent density; the Y-axis is the time it took the agents to escape (average of the 40 experiments). The first configuration has a door in the middle of each wall. Setting the origin of coordinates at the upper left corner of the room, the second configuration has doors at (5, 0), (35, 41), (41, 35) and (0, 5). The third room has doors at (20, 0), (20, 41), (41, 35) and (0, 6). The fourth room has doors at (20, 0), (20, 41), (41, 19), (0, 19). The first configuration gives the better time, as agglomerations tend to form near the doors, making the escape process more difficult. If two doors are “too near” these agglomerations are even bigger. An example of this is configuration 2, which gives the worst results, as it has very close pairs of doors in the room corners. The advantage of configuration 1 is bigger as the agent density goes up, because the effect of the agglomerations as the density of agents increases is bigger.

Figure 4(b) shows the results of the second set of experiments, with eight doors. One of the objectives was to test the efficiency of 8 doors against 4 bigger doors, which can be produced by joining two smaller doors. The first configuration has two doors in each wall, each one placed in an equidistant position to the other door and to the corner of the room. For the second configuration, a big door (composed of two smaller doors) has been placed in the middle of each wall. The third configuration is a room with one big door in the North and in the South, and two smaller ones in the East and West. These are placed at 5 units from the end of the walls. Finally, configuration has two doors in each wall, at 5 units from the end of each wall. The best results have been obtained with the first configuration, for the same reason: if two doors are too near, agglomerations are formed. To reduce this effect, the simulations show that (specially if the room is very crowded) it is better to have numerous small separated doors than a few big doors.

3.3 Extending the model for multiple rooms

In this section, we consider buildings with multiple rooms. The agent structure must be extended with a “mental” representation of the rooms’ connectivity to guide the agent in his navigation towards the exit. We can experiment with two situations: in the first one the agent does not have any *a priori* knowledge of the building connectivity, he builds his *mental map* while navigating through the building looking for the exit. In the second situation, we assume that the agents have partial or total information about the building. In both cases, the mental map is used by the agent to navigate through the building.

Left of Figure 5 reflects this situation. Class *Building* has been introduced, composed by a *number* of rooms. Class *Door* has been extended with the attribute *type* indicating if the door is an exit or leads to another room. While *inner* doors are connected to other inner doors leading to other rooms; exit doors are not connected to other doors, as they lead to the outside. The mental map of the environment the agent builds and uses for navigation is shown in a separate package. A relationship of type “represents” expresses the fact that the agent is able to recognize a real room if he has been in the room before. The same happens with doors inside rooms. The agent also remembers if he has explored the door before or not. As the mental map is a model of the environment (an abstraction), the agent does not memorize room or wall dimensions, as they are not needed for navigation. The agent capabilities have been extended with the possibility to *memorize* new rooms or doors as they are discovered. Capabilities have been arranged in two layers. The upper layer capability (*getDoor*) is higher-level than the lower layer ones and is used by the agent to decide the most appropriate door to go to, and accesses the mental map.

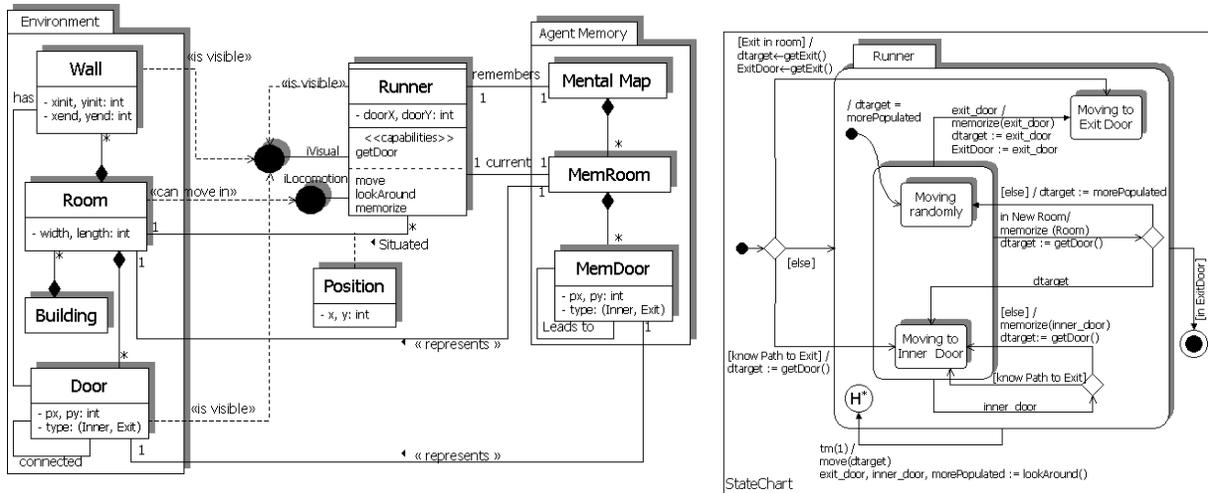


Figure 5: Agent Class diagram with the model for multiple rooms (left). Behaviour of the agent (right).

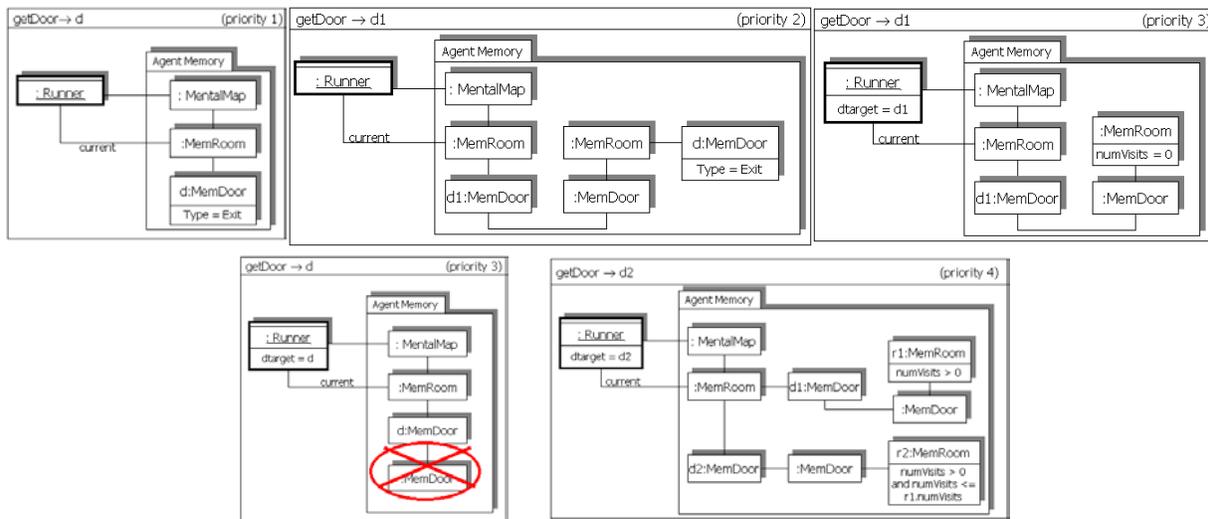


Figure 6: Collaboration Diagrams for Specifying the getDoor capability

If the agent has *a priori* knowledge of the building map, then this capability guides the agent through the shortest sequence of rooms towards the exit. If the agent does not have *a priori* knowledge, then his mental map may not be complete, and several situations can arise. In the easiest case, if he knows an exit door in the current room, this is the most appropriate door. If an exit door is not present in the current room, then the agent searches in his mental map to check if some of the neighbour rooms have an exit door. If this is the case, then the most appropriate door is the one leading to that room. In other case, the agent chooses the door that leads to a non-visited room, and if all rooms have been visited before, he chooses the least visited room. This complex behaviour can be formalized using a number of *agent collaboration diagrams* specifying the expected behaviour of the *getDoor* capability. Each collaboration diagram specifies a situation that, if present at run time, will cause the

capability specification to be executed. For example, the first diagram in figure 6 shows the situation in which an agent is in a room with an exit door. In this case, the capability returns this door as the most appropriate. This diagram does not state that the agent must only know one room, but that this is the minimum set of elements that must be present in order for this situation to be valid.

The second diagram specifies the situation in which the exit door is in a neighbour room. The third diagram applies when the agent does not know any exit door in the current or neighbour rooms (or there are not any). In this case the agent chooses a room not visited before. In the fourth diagram the situation is the same, but the agent does not have a complete knowledge of the environment: the map is not complete. If he finds a door which does not have any connection, the door is not explored. The *negative application condition* means that the agent must

be in a room with a door that has not been explored. Finally, the last diagram shows a situation in which the agent chooses the least visited neighbour room.

The Statechart showing the agent's behaviour has to be modified to consider the navigation between rooms and is shown to the right of Figure 5. If the agent does not have *a priori* knowledge of the environment, then from the initial state he moves to the "Moving Randomly" state. If the agent has *a priori* knowledge, two situations may arise. In the first one, the agent knows that the exit door is in the current room, so the agent moves to state "Moving to Exit Door". In the second one, the agent knows that the exit door is elsewhere, so he selects the most appropriate inner door to move to and moves to state "Moving to Inner Door".

4. CONCLUSIONS AND FUTURE WORK

This paper has proposed some extensions to the UML for the modelling of reactive or hybrid agent simulations. The extensions introduce elements similar to interfaces to express agent's sensors and effectors. Special relationships are introduced to express the fact that other agents or objects can be sensed or acted upon by these sensors and effectors. Instances of these relationships and symbols can be found in *agent diagrams* (a kind of diagram similar to object diagrams). Agent capabilities are declared in an extra box in the *agent class* box. Capabilities can be formally specified using a number of *agent collaboration diagrams*, in a similar way as graph grammars rules. Packages are used to separate the environment and the agent memory. The extensions continue the line of the ones proposed by the *aUML* community and have been used to model building evacuations. This kind of simulations is an inexpensive means to test building security, and can be a complement to real evacuation simulations.

We are extending the model with the possibility to evaluate exit signals placement, experimenting with situations of low visibility and communication between agents. We want to test the model with real buildings, validating the simulation results with data from real building evacuations. We are also implementing the proposed UML extensions in the meta-modelling tool *AToM³* [de Lara and Vangheluwe 2002], in such a way that code for some agent programming languages will be generated from the models. We are also constructing a meta-model to allow the users model different kinds of buildings. These models have to be translated into object diagrams for further processing.

Acknowledgement: This paper has been sponsored by the Spanish Ministry of Science and Technology (TIC2002-01948).

REFERENCES

- Agent UML (aUML) home page: <http://www.aUML.org>
- Alfonseca, M., de Lara, J. 2002. "Two level evolution of foraging agent communities". *BioSystems* Vol 66, Issues 1-2, pp.: 21-30.
- Bauer, B., Müller, J., Odell, J. 2000. "Agent UML: A Formalism for Specifying Multiagent Interaction". In Proc. of Agent-Oriented Software Engineering 2000, Springer. pp.: 91-103.
- Bauer, B. 2001. "UML Class Diagrams Revisited in the Context of Agent-Based Systems". In Proc. of Agent-Oriented Software Engineering (AOSE) 2001, Agents 2001, Montreal. pp.: 1-8.
- Booch, G., Rumbaugh, J., Jacobson, I. 1999. "The Unified Modeling Language User Guide". Addison-Wesley.
- Brooks, R. 1995. "Intelligence without reason". In *The Artificial Life Route to Artificial Intelligence. Building Embodied, Situated Agents*. Lawrence Erlbaum Associates.
- de Lara, J., Vangheluwe, H. 2002. "AToM³: A Tool for Multi-Formalism Modelling and Meta-Modelling". LNCS 2306, p.: 174-188. Springer. AToM³ home page: <http://atom3.cs.mcgill.ca>
- Engels, G., Hausmann, J. H., Heckel, R., Sauer, S. 2000. "Dynamic Meta Modeling: A graphical Approach to the Operational Semantics of Behavioral Diagrams in UML". LNCS 1930, pp.: 323-337. Springer.
- Ehrig, H., Engels, G., Kreowski, H.-J., and Rozenberg, G. 1999. "Handbook of Graph Grammars and Computing by Graph Transformation". Vols.1 and 2. World Scientific.
- Jennings, N.R., Sycara, K., Wooldridge, M. 1998. "A Roadmap of Agent Research and Development". *Autonomous Agents and Multi-Agent Systems*, 1, 7-38 (1998). Kluwer.
- Swarm development group home page: <http://www.swarm.org>

BIOGRAPHY



Juan de Lara is an assistant professor at the Computer Science Department of the Universidad Autónoma in Madrid, where he teaches Software Engineering. He is a PhD in Computer Science, and works in Web based Simulation, Agent based Simulation and Multi-Paradigm Modelling.

Face Detection in Grey Images Using Orientation Matching

Linlin Shen and Li Bai
School of Computer Science & IT
University of Nottingham
Nottingham NG8 1BB

1 Introduction

Face recognition is a major area of research with numerous potential commercial and industrial applications. The performance of face recognition techniques depends heavily on the accuracy of the detected face position within the input image - once the face position is determined, a rectangular box will be extracted from that area, normalized in both scale and orientation, and passed onto the face recogniser.

A natural approach to face detection is to use a colour model to locate skin-like areas in the image [1][2][3]. Image pixels can be labelled as skin or non-skin pixels according to the values of the pixels with respect to the colour model. The problem with this approach is that skin colour varies under different lighting conditions and other objects or even the background may have the same colour as the skin. A widely used approach is to model faces and non-faces as two separate classes. A typical such system is Rowley's neural network-based face detector [4]. The detector consists of a set of neural networks trained by a large training set of face images and non-face images. Pixel values of 20x20 subwindows are input to the neural networks, and outputs from these networks are put through an arbitrating process to arrive at the final decision as to whether a subwindow is a face image. Feraud's face location methods [5] are also based on neural networks and the size of the subwindows is 15x20.

Another popular approach to face detection is based on matching facial features. This approach aims to find the arrangement of certain features such as the eyes, nose, and the mouth in the image, which forms a face pattern. Since it is quite difficult to locate these facial features accurately in light of image variations in illumination and facial expression, some feature extraction methods perform wavelet analysis [7] on the image. For real time applications detection speed is very important. Viola et al [9] proposed a rapid object detection algorithm using a basic and over-complete set of Haar-like features and a cascade of classifiers. These classifiers were combined to produce a more powerful one. The multi-stage classification procedure reduces the processing time substantially and yet achieves almost the same accuracy as the single stage classifier. Rainer and Jochen [10] extended the basic set of Haar-like features by a set of 45^o rotated features. In addition, they performed a new post-optimisation procedure for the boosted classifier and improved the performance significantly. A hit rate of 82.3% on the CMU face set is reported. But the method is sensitive to head pose - only nearly frontal faces ($\pm 10^o$) can be detected. To address the over-fitting problem due to lighting conditions, poses and complex backgrounds, R.Y. Qiao and Y. Guo proposed a soft margin AdaBoost algorithm [11]. A regularisation term was introduced and the most effective weak classifiers are selected. Experimental results showed an improved performance over the original AdaBoost algorithm.

Instead of using the original image, experiments have shown that it is possible to locate face patterns in the edge orientation image. Fröba and Küblbeck [6] extract the edge orientation map from a face model and use this to match against the edge orientation map extracted of the input image. To locate faces larger or smaller than the face model a pyramid of edge orientation fields has to be built. Since this method uses only the edge orientation information, false detection usually occurs when image texture or edge frequency is high. In [13], a fast face detection algorithm using skin colour information and orientation map matching is proposed. A colour image is converted to a skin probability image using the Gaussian skin colour model, from which an orientation map is extracted. After that, the orientation map is matched with a pre-generated model. It is indicated in [13] that skin colour information can be used to suppress the background. As a result, false detection at high edge frequency areas can be reduced. Recently, an edge-based shape comparison method [8] was used for face detection. 2D Hausdorff distance (HD) was used as a similarity measure between a general face model and possible instances of the object within the image. After a coarse detection of the facial region, face location is refined in a second phase. Accuracy of 91.8% on BioID database [14] was reported.

This paper proposed an edge orientation based algorithm for face detection in grey images. Since colour information is not available in grey images, orientation histogram is included for detection. Experimental results show that 93.7% accuracy is achieved for the BioID database, which contains 1521 images with large a variety of lighting conditions and backgrounds. Since the histogram is used to filter the blocks before they are matched with the template, our algorithm is also faster in speed.

2 Orientation Extraction

As shown in [13], a block-based orientation extraction method is adopted in our algorithm. Two 3×3 Sobel operators, S_x for horizontal filtering and S_y for vertical filtering, were convolved with the image $I(x, y)$ to generate two gradient images $G_x(x, y)$ and $G_y(x, y)$.

$$G_x(x, y) = S_x * I(x, y) \quad (1)$$

$$G_y(x, y) = S_y * I(x, y) \quad (2)$$

Similar to the algorithm used in [12], the gradient images $G_x(x, y)$ and $G_y(x, y)$ are divided into a series of non overlap windows of size $w \times w$, each pixel (x, y) in the same window centred at pixel (i, j) is assigned the same orientation value $O(x, y)$ as below:

$$\begin{aligned} V_y(x, y) &= \sum_{u=i-\frac{w}{2}}^{u=i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{v=j+\frac{w}{2}} 2G_x(u, v)G_y(u, v) \\ V_x(x, y) &= \sum_{u=i-\frac{w}{2}}^{u=i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{v=j+\frac{w}{2}} (G_x^2(u, v) - G_y^2(u, v)) \\ O(x, y) &= \frac{1}{2} \tan^{-1} \left(\frac{V_y}{V_x} \right) \end{aligned} \quad (3)$$

After the orientation field of an input image is estimated, the certainty level of edge orientation at pixel (x, y) in the same window centred at pixel (i, j) is defined as below: [12]

$$C(x, y) = \sqrt{\frac{1}{w \times w} \frac{(V_x^2(x, y) + V_y^2(x, y))}{V_e(x, y)}} \quad (4)$$

where

$$V_e(x, y) = \sum_{u=i-\frac{w}{2}}^{u=i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{v=j+\frac{w}{2}} (G_x^2(u, v) + G_y^2(u, v)) \quad (5)$$

A 3×3 box filter (or averaging mask) is used to smooth the estimated orientations obtained from previous result so as to remove any abrupt changes in orientation that are caused by noise in the low image quality regions. The edge information on homogenous parts of the image where no grey value changes occur is often noisy and bears no useful information for the detection [6]. A threshold T_c is applied to the certainty level $C(x, y)$ to generate an edge certainty level field $C_t(x, y)$.

$$C_t(x, y) = \begin{cases} C(x, y) & \text{if } C(x, y) > T_s \\ 0 & \text{else} \end{cases} \quad (6)$$

Figure 3 (b) shows the orientation map extracted from Figure 3 (a). The edge orientation information can be rewritten as a vector field as below:

$$\mathbf{V}(x, y) = C_t(x, y)e^{jO(x, y)} \quad (7)$$

3 Orientation Histogram

In our experiment, we found that orientation histogram was also a very useful feature for face detection. To apply this kind of feature for detection, a face orientation model was generated at first as described in section 4.1. The histogram of this orientation model is shown in Figure 1. From this figure, we can see that the range of orientation is from 0 to π , the histogram is nearly symmetrical along the axis located at $\pi/2$ for a face with frontal view and upright position. Figure 2 (b), (d) show the orientation histogram from a typical face image and a typical background block from the test database respectively. The orientation histogram of the face image is similar to that of model except that the curve is less smooth. On the other hand, there are much difference between the histogram of background block and that of model.

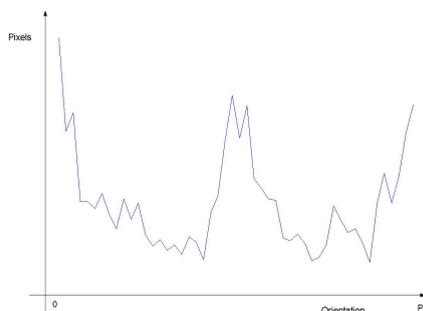


Figure 1 Orientation Histogram of Face Model

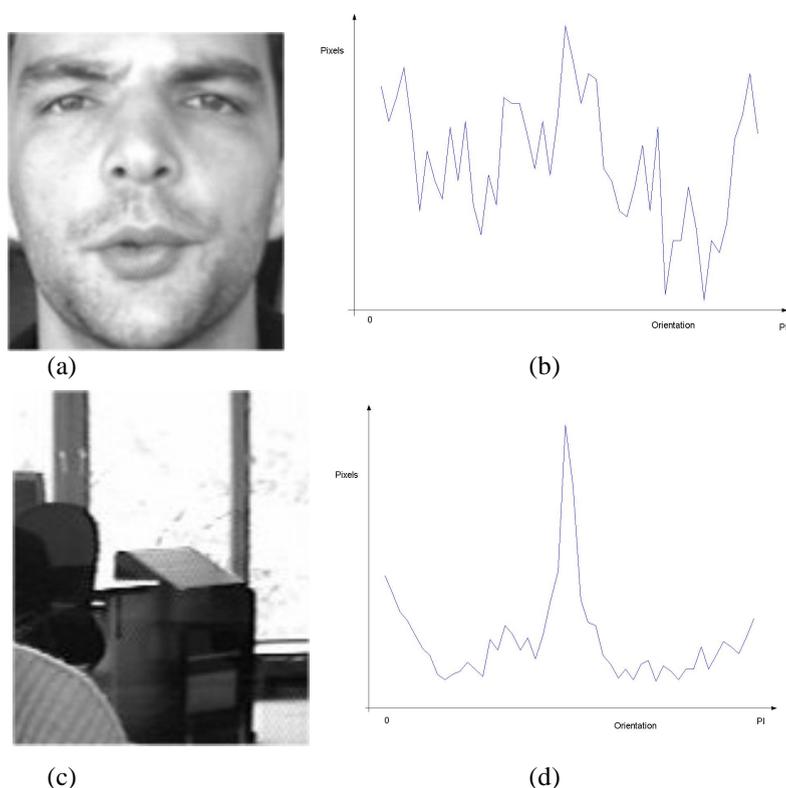


Figure 2 Orientation Histogram of Typical Face Image and Background Block

4 Matching

4.1 Orientation Map Matching

A face model was built from a sample of hand-labeled face images. Ten face images are cropped, aligned and scaled to the size 90×120 . Ten orientation maps were extracted from the face images and a model orientation map was calculated by averaging the ten maps. For face detection, the orientation model $\mathbf{V}_m(x, y)$ is slid over the input orientation image and the similarity between the model and the underlying orientation block centred at pixel (i, j) is calculated and normalized as below:

$$S_O(i, j) = \frac{\sum_{u=i-\frac{w}{2}}^{u=i+\frac{w}{2}} \sum_{v=j-\frac{h}{2}}^{v=j+\frac{h}{2}} \text{sim}(\mathbf{V}_m(u, v), \mathbf{V}_I(i+u, j+v))}{M} \quad (8)$$

where $\mathbf{V}_I(x, y)$ is the orientation map of the input image, w is the width of the model orientation map, h is the height of the model orientation map, M is the number of orientation vectors with strength > 0 in the model orientation map and

$$\text{sim}(\mathbf{V}_1, \mathbf{V}_2) = \begin{cases} \cos(|\arg(\mathbf{V}_1) - \arg(\mathbf{V}_2)|) & \text{if } |\mathbf{V}_1|, |\mathbf{V}_2| > 0 \\ 0 & \text{else} \end{cases} \quad (9)$$

4.2 Histogram Intersection

Histogram intersection is used in our algorithm to match two histograms q and v . The similarity score between the model histogram v and test histogram q is calculated as below:

$$S_H = \frac{\sum_{j=1}^N \min(q_j, v_j)}{\sum_{j=1}^N v_j} \quad (10)$$

where N is the number of bins used to quantize the orientations. q_j, v_j are the values corresponding to bin j in histogram q and v respectively.

4.3 Face Detection Algorithm

A resolution pyramid of orientation map is used to detect faces of different sizes. The size ratio between two resolution levels is set to be 1.25. At each resolution level, the orientation model \mathbf{V}_m is slid over the resized orientation image. When the model is located at pixel (i, j) in a resized image of resolution level l , the similarity score $S(i, j)$ between the underlying block and the model is calculated as below:

$$S(i, j) = \begin{cases} 0 & \text{if } S_H < T_H \\ w_o \times S_O(i, j) + w_h \times S_H(i, j) & \text{else} \end{cases} \quad (11)$$

where w_o, w_h are two weight values ($w_o + w_h = 1$), T_H is a preset threshold, $S_O(i, j)$ and $S_H(i, j)$ denote the orientation map matching score and the histogram similarity score calculated for the block centred at pixel (i, j) respectively. To reduce the effect of low resolution, histogram similarity score is calculated at a higher resolution level. The corresponding block B centred at (i, j) is retrieved from the image of resolution level 0 (original image) and the orientation histogram of block B is then calculated and matched with that of model, using equation (10). A similarity map was generated for each resolution level and from these similarity maps, the pixel with the maximum similarity score is found and a face is detected at location of this pixel.

5 Experimental Results

A test set, BioID database [14], is used in our experiments to evaluate the proposed algorithm. The set consists of 1521 images (384×288 pixel, grayscale) of 23 different persons and has been recorded during several sessions at different places. This set features a large variety of illumination, background and face size. Two algorithms, named as A and B, are evaluated in our experiments. Orientation template matching [6] is used in algorithm A, while both orientation map matching and orientation histogram intersection are applied in algorithm B. Blocks of size 3×3 are used to calculate the orientation map and orientation is quantized to 50 bins. Figure 3 shows the different results when the two algorithms are applied to an image from the database. Figure 3 (c), (d) shows the detection result when algorithm A and B is applied to orientation map (b) respectively. The image block that yields the maximum similarity score is bounded with a rectangle. From this figure, we observe that when only orientation information is used, false detection easily occur where edge frequency, e.g. texture, is high. After the histogram information is involved, false detection for this image is avoided and better detection accuracy can be achieved.

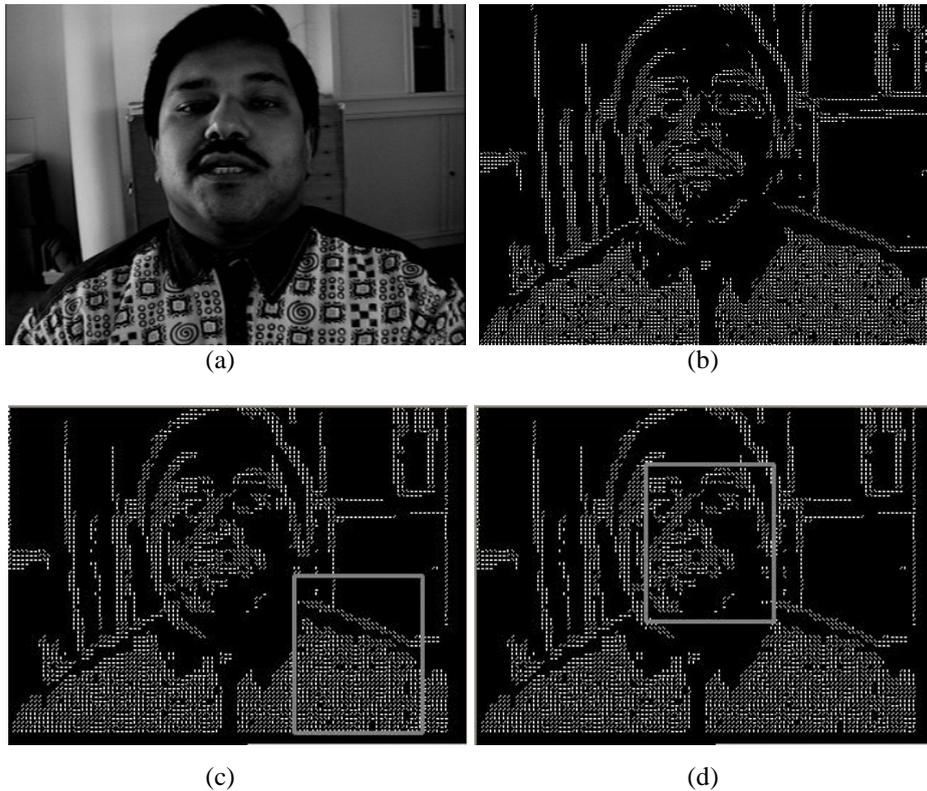


Figure 3 Detection Results for Algorithms A & B when applied to a Face Image

Table 1 shows the detection results for these two algorithms after they are applied to the whole database. Accuracy of 93.7% is achieved for the new algorithm B. Since the histogram is also used to filter the blocks before they are matched with the template, less processing time is achieved for algorithm B. The average processing time per image for algorithm B is 1.73 seconds, which is about 17% less than that of algorithm A. When this algorithm is applied to video sequence, the search space can be greatly reduced if the information from previous frame is used. As a result, our algorithm can be easily applied to real-time application.

Table 1 Comparative Results for Algorithm A and B

Algorithm	A	B
Accuracy	89.49%	93.7%
Average Processing Time (Sec)	2.01	1.73

6 Conclusions

In this paper, we have proposed an orientation based algorithm for face detection in grey images. Both orientation map matching and orientation histogram intersection are applied in our algorithm. Orientation histogram is firstly used to filter the blocks before they are matched with the template. After that, the histogram similarity score is weighted together with the orientation map matching score to yield a total matching score for the image block under processing. The test set of BioID database is used to evaluate our algorithm. Experimental results show that 93.7% accuracy is achieved for the database, which contains 1521 images with large variety of lighting and background.

7 References

- 1 BMenser and F.Muller, Face Detection in Color Images Using Principal Components Analysis. *Proceedings Seventh International Conference on Image Processing and its Applications*, vol. 2, pp.620–624 , July 1999.
- 2 Eli Saber, a.Murat Tekalp, Frontal-view Face Detection and Facial Feature Extraction Using Color, Shape and Symmetry Based Cost Function, *Pattern Recognition Letters*, 19(8):669-680, June, 1998.
- 3 Saber, E. Takalp, A.M., Eschbach, etc., Automatic Image Annotation Using Adaptive Color Classification. *Graphical Models and Image Process.* 58, 115-126, 1996.
- 4 Rowley, Shumeet Baluja, and Takeo Kanade, Neural Network-Based Face Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 1, January 1998.
- 5 Rapha el Feraud, Olivier J. Bernier, Jean-Emmanuel Viallet, and Michel Collobert, A Fast and Accurate Face Detector Based on Neural Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 1, January 2001.
- 6 Bernhard Froba and Christian Kublbeck. Real-Time Face Detection Using Edge-Orientation Matching. *3rd International Conference on Audio and Video Based Biometric Person Authentication*, p78-83, Sweden, June, 2001.
- 7 F.Smeraldi, O.Carmona, and J.Bigun. Saccadic Search with Gabor Features Applied to Eye Detection and Real-time Head Tracking. *Image and Vision Computing*, 18:323-329, 2000.
- 8 Oliver Jesorsky, Klaus J. Kirchberg and Robert W. Frischholz. Robust Face Detection Using the Hausdorff Distance. *3rd International Conference on Audio and Video Based Biometric Person Authentication*, p90-95, Sweden, June, 2001.
- 9 P.Viola and M.Jones, Rapid Object Detection using a Boosted Cascade of Simple Features. *IEEE Conf. on Computer Vision and Pattern Recognition*, Kauai, Hawaii, Dec. 2001.
- 10 Rainer Lienhart and Jochen Maydt. An Extended Set of Haar-like Features for Rapid Object Detection. *IEEE ICIP 2002*, Vol. 1, pp. 900-903, Sep. 2002.
- 11 R.Y. Qiao and Y.Guo. Face Detection Using Soft Margin Boosting. *Image and Vision Computing New Zealand Conference*, pp.157-161, New Zealand, Nov. 2002.
- 12 Lin Hong, Anil Jain, Sharath Pankanti, Ruud Bolle, An Identity Authentication System Using Fingerprints. *Proceedings of the IEEE*, Vol. 85, No. 9, Sept. 1997, pp. 1365-1388.
- 13 Li Bai and LinLin Shen. Face Detection by Orientation Map Matching. Accepted by *International Conference on Computational Intelligence for Modelling Control and Automation*, Austria, Feb. 2003.
- 14 <http://www.bioid.com/research/index.html>

ADAPTIVE OPTIMISTIC SYNCHRONISATION FOR MULTI-AGENT DISTRIBUTED SIMULATION

MICHAEL LEES

*School of Computer Science
University of Nottingham
Nottingham NG8 1BB, UK*

mhl@cs.nott.ac.uk

BRIAN LOGAN

*School of Computer Science
University of Nottingham
Nottingham NG8 1BB, UK*

bsl@cs.nott.ac.uk

GEORGIOS
THEODOROPOULOS

*School of Computer Science
University of Birmingham
Birmingham B15 2TT, UK*

gkt@cs.bham.ac.uk

ABSTRACT In this paper we describe an adaptive, optimistic synchronisation mechanism for the parallel discrete event simulation of agent-based systems. The mechanism uses the Sphere of Influence (SoI) of an event (the region of the shared simulation state read or written to by the event) to define an adaptive metric which can be used with a throttling mechanism such as moving time windows. We show how such a metric can be calculated by monitoring the common reads and writes made by the agents to the shared simulation state modelling the agent's environment, and present the results of our preliminary investigations into the relationships between agent read and write patterns and rollback frequency.

Keywords : Distributed Simulation, Agent-based Systems, Synchronisation.

1. INTRODUCTION

An *agent* can be viewed as a self-contained, concurrently executing thread of control that encapsulates some state and communicates with its environment (in which the agent is embedded) and possibly other agents via some sort of message passing. The *environment* of an agent is that part of the world or computational system 'inhabited' by the agent. The environment may contain other agents whose environments are disjoint with or only partially overlap with the environment of a given agent. Agent-based systems offer advantages when independently developed components must inter-operate in a heterogeneous environment and are increasingly being applied in a wide range of areas including telecommunications, business process modelling, computer games, control of mobile robots and military simulations.

While agent-based systems offer great promise, their adoption has been hampered by the limitations of current development tools and methodologies. Multi-agent systems are often extremely complex and it can be difficult to formally verify their properties. As a result, design and implementation remains largely experimental, and experimental approaches are likely to remain important for the foreseeable future. In this context, simulation has a key role to play in the development of agent-based systems, allowing the agent designer to learn more about the behaviour of a system or to investigate the implications of alternative agent architectures, and the agent researcher to probe the relationships between agent architectures, environments and behaviour.

In [Logan and Theodoropoulos, 2001] a parallel discrete event simulation framework for multi-agent systems is presented. Identifying the efficient distribution of the agents' environment (namely, the *shared state*) as a key problem in the simulation of agent-based systems, the framework models agents as Logical Processes and the environment as a tree-shaped network of processes (referred to as *Communication Logical Processes* or *CLPs*) which is dynamically

reconfigured to reflect the changing interaction patterns between the agents and their environment in the simulation.

The central concept of the framework is the notion of the *Sphere of Influence* (SoI). The SoI of an event is defined as the set of state variables read or written as a consequence of the event and depends on the type of event (e.g., sensor events or motion events), the state of the agent or environment logical process which generated the event and the state of the environment. The SoI of an event is limited to the immediate consequences of the event rather than its ultimate effects, which depend both on the current configuration of the environment and the (autonomous) actions of other agents in response to the event. The SoI of an agent process p_i over the time interval $[t_1, t_2]$, $s(p_i)$, is then defined as the union of the spheres of influence of the events generated by the process over the interval. In [Logan and Theodoropoulos, 2001], the SoI of the LPs are used to derive an idealised decomposition of the shared state into CLPs to facilitate dynamic load balancing and interest management. In this paper, we discuss how the SoI can be exploited in the design of an adaptive synchronisation mechanism.

The rest of the paper is organised as follows: In sections 2 and 3 we give a brief introduction to simulation and synchronisation and discuss the role of the shared state in an agent simulation. In section 4 we describe our adaptive synchronisation mechanism and present a metric based on Spheres of Influence. Although the metric could be used with any optimism limiting mechanism, for clarity and ease of explanation we assume a window based scheme where a smaller window implies lower optimism (e.g., moving time windows [Sokol and Stucky, 1990]). In section 5 we present our experimental results. The paper concludes with section 6 where we touch on possible future work.

2. SYNCHRONISATION

Every simulation model specifies the physical (real) system in terms of *events* and *states*. Executing a simulation therefore consists of 'processing' events, which correspond to real events in the physical system. Simulations can be classified

into two groups depending on the way events are processed and state updates occur: *continuous* and *discrete*. In a continuous simulation, state changes occur continuously, whereas in a discrete simulation events occur at fixed points in time and execute instantaneously. In an *event driven* simulation state variables are updated only when something interesting occurs, i.e., an *event*. Each event occurs at a particular instant in simulation time and the event has this time associated with it, this is known as the *time-stamp* of the event. A single processor (sequential) discrete event simulation consists of the following,

- *State Variables* – collectively describe the state of the system
- *Event list* – list of events to be processed
- *Global clock* – denotes the simulation time

If the discrete event simulation is split into multiple *Logical Processes* (LPs) and spread across multiple machines it becomes a *Parallel Discrete Event Simulation* (PDES).

A sequential discrete event simulation can easily ensure that events are processed in time stamp order as it processes the event with the smallest time stamp in the event list. Spreading the simulation over multiple processes (PDES) requires multiple event lists, one for each LP. A consequence of this is that ensuring the events are processed in time stamp order is less straightforward. In asynchronous, event-driven distributed simulation, each LP maintains its own local clock with the current value of the simulated time, Local Virtual Time (LVT). This value represents the process's local view of the global simulated time and denotes how far in simulated time the corresponding process has progressed. With each LP processing its event list independently and advancing its LVT at its own rate, it may be the case that events are processed out of time stamp order. Therefore a mechanism is required to ensure the parallel simulation faithfully implements the causal dependencies and partial ordering of events dictated by the causality principle in the modelled system.

It has been shown [Lamport, 1978] that a distributed system consisting of asynchronous concurrent processes will not violate the causality principle if each process consumes and processes event messages in non-decreasing timestamp order (the *local causality constraint* (LCC)). There are two main approaches to ensuring that the local causality constraint is not violated: *conservative* and *optimistic*. Conservative mechanisms strictly avoid violation of the LCC while optimistic mechanisms provide a means to undo computation which causes a violation. In more recent years hybrid mechanisms which take aspects of both have been developed, i.e., optimistic schemes with constrained optimism such as moving time window [Sokol and Stucky, 1990]. Other optimistic schemes have been developed so that the degree of optimism (how constrained they are) can be decided at run time. These are known as adaptive synchronisation mechanisms (e.g. [Ferscha, 1995]).

3. SHARED STATE AND SPHERES OF INFLUENCE

Consider an agent simulation with two agents, A1 and A2. The shared state of their environment can be modelled as a table (see Table 1). The table shows the *read* and *write*

patterns for each variable. We use the term *access* to indicate either a read or a write.

Variable	Access patterns
x_1	$(A_2, R, t = 1), (A_2, R, t = 3), (A_1, W, t = 2)$
x_2	$(A_1, R, t = 2), \dots$
.	
.	
x_n	\dots

Table 1: A global view of the shared state

Table 1 depicts the access patterns for two variables in the shared state. Each access is represented by a triple: A_n represents the agent performing the access, R/W represents whether the access was a read or write and t represents the virtual time at which the access occurred. The left to right ordering indicates when the access arrived in real time. So the access $(A_2, R, t = 1)$ arrived before $(A_2, R, t = 3)$ for variable x_1 .

The table also depicts a *rollback pattern* occurring on variable x_1 . A *rollback* pair consists of a read R made by LP_i with time stamp T_R and a write W made by LP_j with time stamp T_W ($j \neq i$). We say a rollback pair is a rollback pattern on variable x if $T_W < T_R$. A rollback pattern results in an actual rollback when the write W is realised after the read R in real time. For variable x_1 in Table 1 this occurs when A_1 performs the write at $t=2$. The read performed by A_2 with virtual time stamp $t=3$ arrived, in real time, before the write performed by A_1 . This means the read performed by A_2 won't reflect the correct value and so A_2 needs to rollback to before the read occurred. In terms of synchronisation a key observation is that the probability of rollback is increased when many different LPs read and write the same variables. We now extend and clarify the definition of SoI from [Logan and Theodoropoulos, 2001] by splitting the SoI into two distinct sets:

1. The *sphere of influence of Writes* (SoI_W), which contains the set of variables written to by the LP over the time period $[t_1, t_2]$.
2. The *sphere of influence of Reads* (SoI_R), which contains the set of variables read by the LP over the time period $[t_1, t_2]$.

Considering the example given above we can now say:

1. Any variable which appears only in SoI_W for all LPs (agents) over $[t_1, t_2]$ (i.e., no agent reads the variable) is not important in terms of rollback;
2. Any variable which appears only in SoI_R for all LPs (agents) over $[t_1, t_2]$ (i.e., no agent writes the variable) is not important in terms of rollback either; and
3. A variable which is present in the SoI_R of one LP (agent) and in SoI_W of another may cause a rollback.

A rollback will occur if the variable is in both sets and a late write arrives from one LP after a read was made by another LP (as described in the example above). The third point

above requires that a minimum of two LPs be involved; intuitively as the number of LPs involved increases so does the likelihood of rollback. We can therefore predict the likelihood of rollback using the access patterns of any particular variable.

	Many Writes	Few Writes
Many Reads	High	Medium
Few Reads	Medium	Low

Table 2: How probability of rollback due to a particular variable is affected by reads and writes

1. A variable which is in both SoI_R and SoI_W for many different LPs (agents) – High probability of rollback
2. A variable which is in the SoI_R of a single LP (agent) and in many LPs SoI_W – Medium probability of rollback
3. A variable which is in the SoI_W of a single LP (agent) and in many LPs SoI_R – Medium probability of rollback
4. A variable which only appears in the SoI_R of one LP and in the SoI_W of another – Low probability of rollback

4. AN ADAPTIVE MECHANISM

Table 2 uses the notion of SoI to give an indication of how likely rollback is due to different access patterns of a particular variable. This suggests a simple scheme where the optimism of an LP should reflect the access patterns of the variables in its SoI, the more common variables, the less optimistic. Figure 1 illustrates this idea. There are four agents (LPs) represented by small squares. The circles surrounding each of the agents are an abstraction of their spheres of influence which are defined in table 3¹. Circles overlapping indicates that the two agents read and write some common variables. With the assumption that each agent reads or writes a variable within its sphere of influence with equal probability over the time interval $[t_1, t_2]$, a larger overlap indicates that the two agents access more common variables.

Agent	Sphere of Influence
A1	x_1, x_2, x_3, x_4, x_8
A2	x_1, x_5, x_6, x_8
A3	$x_2, x_3, x_4, x_5, x_6, x_7, x_8$
A4	x_7

Table 3: Agents sphere of influence.

Agent A4 reads and writes the smallest number of common variables (smallest intersection) and under the suggested scheme would be given the largest time window and hence would execute with the highest degree of optimism. Agent A3, however, reads and writes the largest number of common variables (it's circle intersects with the other three) and would be given the smallest time window. Hence agent A3 would execute with the least degree of optimism (most conservative). If we assume a balanced load (or at least not a

¹For this example we make the valid simplifying assumption that $SoI_R=SoI_W$

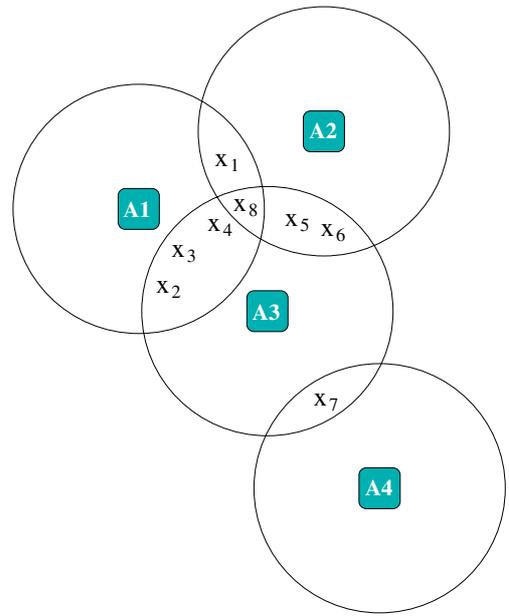


Figure 1: Four agents and the intersection of their spheres of influence

highly imbalanced load) then A4 will execute a faster rate than A3. As a result, the difference in LVT between A3 and A4 will increase with time as shown in Figure 2 (here A3 and A4 have a time window of 5 and 20 respectively).

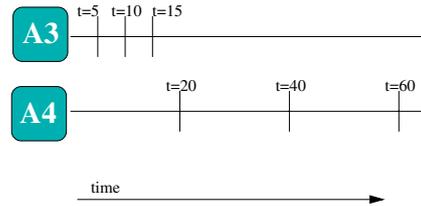


Figure 2: Progression of LVT with time windows

A4 and A3 read and write some common variables (x_7), so the fact that they execute with different degrees of optimism increases the chance of rollback, e.g., A4 would have to rollback if it reads x_7 at virtual time $t=40$ and A3 subsequently (in real time) writes x_7 at virtual time $t=15$. Clearly it is not desirable to have LPs which read and write common variables executing at extremely different rates. If the SoI of two agents A_i and A_j overlap then we can say the probability of A_i causing rollback on A_j is affected by the number of *critical accesses* made between A_i and A_j (CA_{ij}) defined as;

$$CA_{ij} = |SoI_{Ri} \cap SoI_{Wj}| + |SoI_{Wi} \cap SoI_{Rj}| \quad (1)$$

We can then say for any agent A_i (or LP_i , the terms can be used interchangeably here) the likelihood of rollback is affected by all critical accesses made between A_i (CA_i) and all other $n - 1$ agents,

$$CA_i = \sum_{j=1}^{n-1} (CA_{ij})_{j \neq i} \quad (2)$$

The size of the time window for a given agent A_i is therefore

inversely proportional to

1. the number of critical accesses in its sphere of influence, CA_i ; and
2. for each neighbouring agent A_j (i.e., $CA_{ij} \neq 0$) whose LVT differs from A_i by ΔLVT_{ij} , $\Delta LVT_{ij} \times CA_{ij}$

We can now state the form of the equation used to determine window size (optimism) of an agent A_i ,

$$WS_i = \frac{a}{k_1 CA_i \times \sum_{j=1}^{n-1} (k_2 CA_{ij} \times k_3 \Delta LVT_{ij})} \quad (3)$$

Where the total number of agents in the simulation is n and a , k_1 , k_2 , k_3 are appropriate constants.

To enable each LP to calculate its time window we need to have global information regarding the access patterns on shared state variables and the LVTs of the LPs in the simulation. We now propose a simple centralised scheme for collecting the relevant information. First we allocate a counter to each variable in the shared state. This counter indicates how many different LP's spheres of influence the variable lies in and so indicates how difficult the variable is to associate with a particular LP. A central LP is used to collect this information, with all reads and writes passed via this central LP². The LP would simply keep a list of all access made for each variable in the shared state (similar to Table 1) between a time period $[t_1, t_2]$. From this, the centralised LP can determine CA_i and CA_{ij} for all LPs in the simulation. It can also determine the current LVT of each LP (via the time stamp of the most recent access) and hence ΔLVT_{ij} for all LPs i and j .

At time t_2 a GVT computation would occur and the new window size would be calculated for all LPs. This relies on the fact that the access patterns from the time period $[t_1, t_2]$ will produce an appropriate window for the time period $[t_2, t_3]$. If the length of the time intervals are chosen appropriately, the change in the spheres of influence from one time period to the next should be small. It was shown in [Logan and Theodoropoulos, 2001] for some typical agent simulations the change in spheres of influence is limited.

Using a central controller LP in this way limits the message traffic. Without a central controller each LP would need to broadcast all information to the other LPs in the simulation. A protocol which uses global information in this way incurs extra overhead, and we envisage that the development of CLPs will offer a solution to this problem.

5. RESULTS

Investigation of the metric is still at a preliminary stage and our results to date relate to the spheres of influence rather than the performance of the metric itself. These experiments and results serve as feasibility study for later development of the metric. The experiments here are performed in the SIM_TILEWORLD [Lees, 2002] testbed implemented in the agent toolkit SIM_AGENT. Tileworld is a commonly used testbed in agent evaluation and experimentation. The Tileworld environment consists of tiles, holes and obstacles. A

²This LP behaves much the same way as the CLPs described in [Logan and Theodoropoulos, 2001]

Tileworld agent tries to score as many points as possible by filling holes with tiles (the agent receives a point for each tile placed in a hole). The Tileworld environment (see figure 3) is dynamic in that objects (holes, tiles and obstacles) are created at random with a predefined lifespan. The experimenter

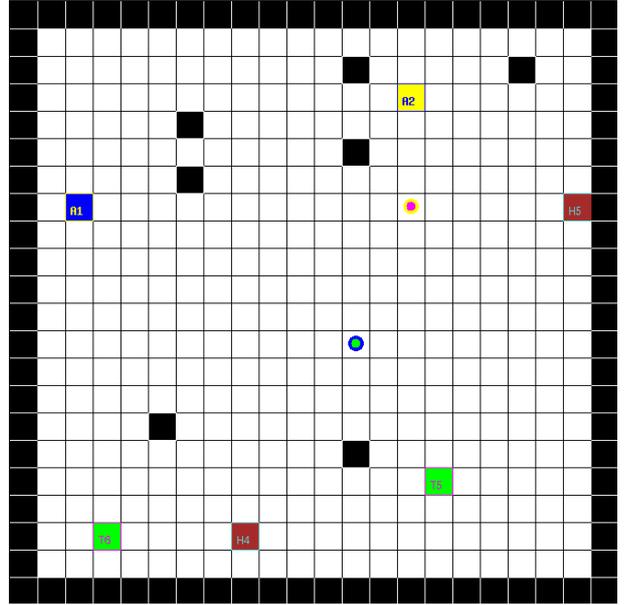


Figure 3: The SIM_TILEWORLD testbed

can control how dynamic the environment is by defining the probability of new objects being created and the lifespan of those objects. For example, an environment with high object creation probability and short lifespan would be very dynamic. Tileworld also allows the experimenter to vary the density of objects in the environment: if the object creation probability is high and the lifespan is high there will be a large number of objects in the environment at any one time.

The results presented below show how access patterns vary when the density of objects in the environment is changed. In particular the experiments look at how access patterns relate to the number of possible rollbacks occurring in a simulation. The initial hypothesis being that as the number of common accesses increases so does the number of possible rollbacks occurring. Possible rollbacks are identified by a particular access pattern. Firstly a ΔLVT value l is set, this defines the largest possible difference between the time stamp of the access and the LVT of the receiving LP³. A possible rollback pattern occurs if a variable is written to by one agent and then read by another agent up to l time periods (cycles) later. In these experiments we set l to be 3.

To explain the results we first introduce the notion of a common read and common write. A *common read* occurs when one agent reads to a variable and another agent also accesses the same variable (i.e., read or write). A *common write* occurs when one agent writes to a variable and another agent also accesses the same variable (i.e., read or write). The first graph (figure 4) shows the read patterns of two agents in a 20x20 Tileworld environment during a 20 cycle period. Each agent has a sensor range of 5 squares. The graph shows how

³This value should reflect typical differences in LVT of two LPs

the number of reads made by each agent and the common reads varies with the number of objects in the environment. With an object creation probability of 0.1 both agents made about 300 reads, a high percentage of these were common reads (200 or 66%). At 0.1 object creation probability there are few objects in the environment in this situation the agents tend to aim for the same tiles and holes. As the number of objects in the environment increases the proportion of common reads drops (to around 44%). Figure 5 shows how the number of common reads varies with the size of the environment. With the environment at size 80x80 the number of common reads has almost dropped to zero. This could be due to two things, firstly the agents are further apart and secondly the environment is less densely populated with tiles and holes.

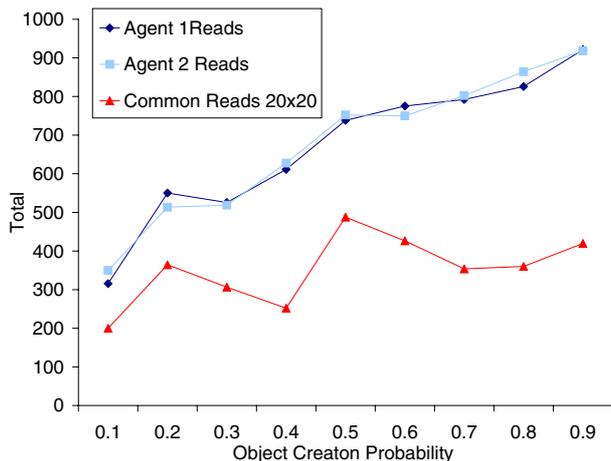


Figure 4: The reads made by two agent in a 20x20 Tileworld environment in a 20 cycle period

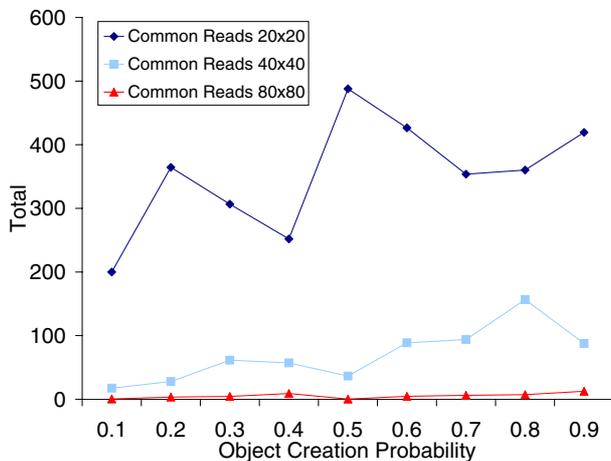


Figure 5: How the number of common reads made between two agents varies with object creation probability and environment size

The graphs in Figures 6-8 show the write patterns of two agents in environments of varying size. The graphs also show how closely related rollback patterns and common writes are.

From this we can say that rollbacks patterns will occur when

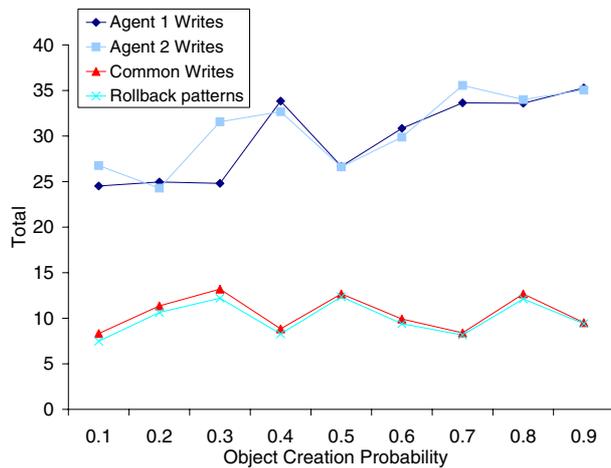


Figure 6: Writes and rollback patterns of two agents in a 20x20 Tileworld environment over a 20 cycle period

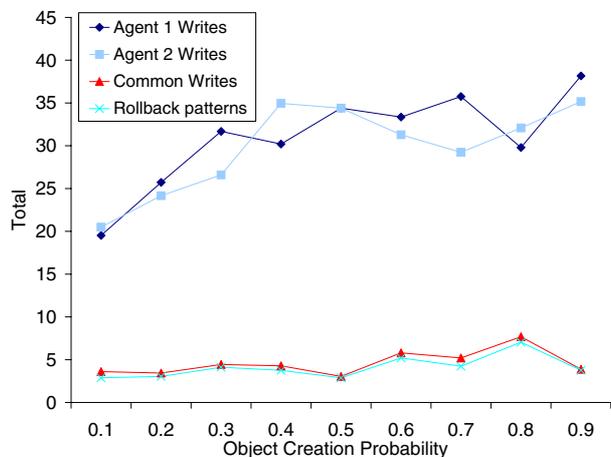


Figure 7: Writes and rollback patterns of two agents in a 40x40 Tileworld environment over a 20 cycle period

the activities of the agents result in a common write. For example, a rollback pattern will occur if agent A_1 is pushing a tile which is within the sensor range agent A_2 up to 3 cycles (moves⁴) later. This conclusion is reinforced upon comparison of the number of rollback patterns in different sized environments. As the environment size increases the number of writes made by each agent drops. The number of common writes drops even further and hence rollback patterns become much less common in larger, less dense environments.

6. DISCUSSION AND FUTURE WORK

In this paper we describe a novel adaptive optimistic synchronisation mechanism for the parallel discrete event simulation of agent-based systems. Our mechanism uses the Sphere of Influence of an event to define an adaptive metric which can be used with a throttling mechanism such as moving time windows. We show how such a metric can be calculated by monitoring the common reads and writes made by the

⁴The agents currently implemented in SIM_TILEWORLD are purely reactive and hence move one square every cycle

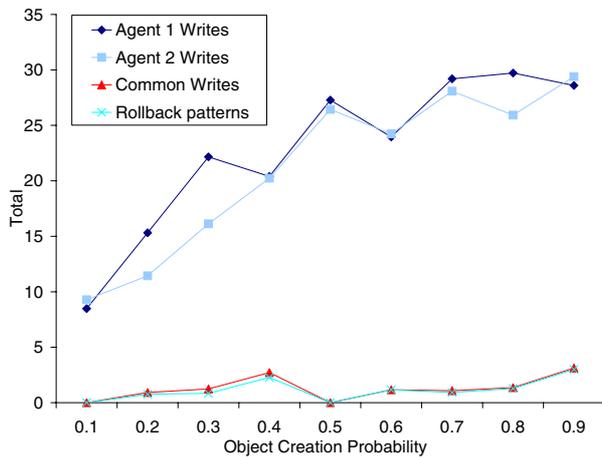


Figure 8: Writes and rollback patterns of two agents in a 80x80 Tileworld environment over a 20 cycle period

agents to the shared simulation state modelling the agent's environment, and present the results of our preliminary investigations into the relationships between agent read and write patterns and rollback frequency.

Our experimental results show, as expected, that the number of common accesses does affect the number of rollback patterns occurring in an agent simulation. Surprisingly, the results have shown that for this particular agent simulation (SIM_TILEWORLD) rollback patterns are very closely related to common writes. This relates to the ratio of writes to reads made by the agents. If, as in this case, the agents perform a large number of reads compared to writes (700/35), then almost all common writes will result in a rollback pattern. Our plans for further work in this area will be toward investigating this relationship. We plan to repeat the experiments with different types of agent simulation with highly different access patterns.

Our long term goal is to implement the adaptive metric in Georgia Time Warp (GTW). Initially a simple C/C++ program will simulate typical access patterns of agent simulation (mirroring the results obtained here). This allows us to test the metric without the need for integrating an agent toolkit with GTW beforehand. If the results are as expected the next step is to integrate an agent toolkit (e.g., SIM_AGENT) with GTW to investigate the performance of the mechanism.

Acknowledgements

We would like to thank Dr Bora Kumova and Tonworio Oguara, both based at the University of Birmingham, for their contribution to this paper. This work is part of the PDES-MAS project⁵ and is supported by EPSRC research grant No. GR/R45338/01.

7. REFERENCES

Ferscha, A. (1995). Probabilistic adaptive direct optimism control in time warp. In *Proceedings of the 9th Workshop on Parallel and Distributed Simulation (PADS '95)*, pages 120–129.

⁵<http://www.cs.bham.ac.uk/gkt/Research/PDES/pdes-mas.html>

Lamport, L. (1978). Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, pages 558–564.

Lees, M. (2002). A history of the tileworld agent testbed. Technical report, Nottingham University.

Logan, B. and Theodoropoulos, G. (2001). The distributed simulation of multi-agent systems. In *Proceedings of the IEEE*, pages 174–185.

Sokol, L. and Stucky, B. (1990). Mtw: Experimental results for a constrained optimistic scheduling paradigm. In *Proc. SCS Multiconf. Distributed Simulation*, volume 22, pages 169–173.

Author Biographies

MICHAEL LEES. is a PhD student studying in the School of Computer Science and IT at the University of Nottingham, UK. He received a joint Honours Computer Science and Artificial Intelligence degree from the University of Edinburgh, UK in 2001. His thesis will be centred of the areas of Multi-Agent systems and distributed simulation.

BRIAN LOGAN. is a lecturer in the School of Computer Science and IT at the University of Nottingham, UK. He received a PhD in design theory from the University of Strathclyde, UK in 1986. His research interests include the specification, design and implementation of agent-based systems, including logics and ontologies for agent-based systems and software tools for building agents.

GEORGIOS THEODOROPOULOS. received a Diploma degree in Computer Engineering from the University of Patras, Greece in 1989 and MSc and PhD degrees in Computer Science from the University of Manchester, U.K. in 1991 and 1995 respectively. He is currently a Lecturer in the School of Computer Science, University of Birmingham, U.K. His research interests include parallel and distributed systems, computer and network architectures and modelling and distributed simulation. He is a co-founder of the Midlands e-Science Center of Excellence in Modelling and Analysis of Large Complex Systems.

ON-LINE DESIGN OF ROBUST FUZZY-LOGIC CONTROL SYSTEMS BY MULTI-OBJECTIVE EVOLUTIONARY METHODS.

P. Stewart (corresponding author) * D.A. Stone *
P.J. Fleming **

** Electrical Machines and Drives Group, Department of
Electronic and Electrical Engineering, University of Sheffield.
Mappin St. Sheffield S1 3JD U.K.*

e-mail:p.stewart@shef.ac.uk, tel: +44 (0)114 2225841.

*** Department of Automatic Control and Systems Engineering,
University of Sheffield. Mappin St. Sheffield U.K.*

Abstract: Evolutionary development of a fuzzy-logic controller is described and is evaluated in the context of hardware in the loop. It had been found previously that a robust speed controller could be designed for a dc motor motion control platform via off-line fuzzy logic controller design. However to achieve the desired performance, the controller required manual tuning on-line. This paper investigates the automatic design of a fuzzy logic controller on-line. An optimiser which modifies the fuzzy membership functions, rule base and defuzzification algorithms is considered. A multi-objective evolutionary algorithm is applied to the task of controller development, while an objective function ranks the system response to find the Pareto-optimal set of controllers. Disturbances are introduced during each evaluation at run-time in order to produce robust performance. The performance of the controller is compared experimentally with the fuzzy logic controller which has been designed off-line. The on-line optimised fuzzy controller is shown to be both robust, possessing excellent steady-state and dynamic characteristics, demonstrating the performance possibilities of this type of approach to controller design.

Keywords: Fuzzy systems, Evolutionary/Genetic algorithms, Methodologies, Models and algorithms.

1. INTRODUCTION

This paper is investigates the potential of multiobjective control design with hardware in the loop. Tuning of PI parameters on-line has been achieved [17] with multiobjective genetic algorithms. Here, the potential of parameter and controller structure tuning on-line is considered. A DC motor dynamometer rig and a microcontroller is used as a platform to develop and assess the control algorithms. In particular, an off-line designed type (fuzzy logic) is considered for performance comparison. An automatic method for fuzzy logic con-

trol design is considered, utilising a multiobjective evolutionary algorithm for the optimisation process. Random disturbances with bounds which reflect realistic parameter variations are injected during each on-line assessment with the aim of producing a controller which is also robust to disturbances.

Fuzzy logic control, comprising a fuzzification interface, rule base and defuzzification algorithm [1,2], has been applied to a wide variety of motion control applications [3,4]. A vital region of interest concerns the implementation of the fuzzy

controller. Several different approaches have been postulated to extract the knowledge base from experts or training examples to construct the input-output membership functions and the fuzzy rule-base. These methods can be based on neural networks [5,6] or the application of fuzzy clustering techniques to construct a fuzzy controller from training data sets [7]. It has been observed that the major drawback of most fuzzy controllers and expert systems is the need to predefine membership functions and fuzzy rules. In [5], a method is proposed based on fuzzy clustering techniques and decision tables to derive membership functions and fuzzy rules from numerical data. A natural evolution of the technique was to integrate Genetic Algorithms (GAs) into the Fuzzy logic design process [8,9,10]. The robustness of the GA allows it to cover a multidimensional search space while ensuring an optimal or near-optimal solution, thus simultaneous design of membership functions and fuzzy control rules can be achieved [11]. The development of these techniques to design optimal fuzzy logic controllers has arisen to satisfy the need which exists when expert heuristic knowledge doesn't exist to translate into controller design.

The performance of a particular control design is fundamentally tied to the accuracy of the model upon which it is based. This is especially true for iterative control design and optimisation procedures. The substitution of hardware in the loop for the software model opens up new possibilities for design based on real world performance indices. In this paper the implementation of GA fuzzy design will be evaluated via an on-line experimental DC motor connected to a DC shunt load motor set to introduce dynamic disturbances. The performance of the resulting motion controller is compared with that of a manually tuned fuzzy controller. The results presented here demonstrate a convenient and practical method to produce a robust controller design on a prototype plant.

1.1 *Multiobjective optimisation by evolutionary algorithm*

Evolutionary algorithms are global parallel search and optimisation methods based around Darwinian principles, working on a population of potential solutions to a problem (in this case the on-line design of an optimal fuzzy logic controller via hardware in the loop). Every individual in the population represents a particular solution to the problem, often expressed in binary code. The population is evolved over a series of generations to produce better solutions to the problem. At every generational step, each individual of the population is run on the hardware, and its performance evaluated and ranked via a cost function.

Individual performance is indicated by a fitness value, an expression of the solution's suitability in the solution of the problem. The relative degree of the fitness value determines the level of propagation of the individual's genes to the next generation. Evolution is subsequently performed by a set of genetic operators which stochastically manipulate the genetic code. Most genetic algorithms include operators which select individuals for mating, and produce a new generation of individuals. *Crossover* and *Mutation* are two well-used operators. The crossover operator exchanges genetic material between parental chromosomes to produce offspring with new genetic code. The mutation operator makes small random changes to a chromosome. Further repetitions of this process are made in the search for the strongest genetic material. The genetic algorithm explores the multidimensional search-space to find good solutions to the problem. It is possible for the GA to find several dissimilar but equally valid solutions to a single problem due to its use of population, and the competing nature of multiple objectives, since real-world problems involve the simultaneous evaluation of multiple performance criteria. Trade-offs occur between competing objectives with the consequence that it is very rare to find a single solution to a particular problem. In reality a family of *non-dominated* solutions will exist. These *Pareto-optimal* [12,13] solutions are those for which no other solution can be found which improves on a particular objective without a detrimental effect on one or more competing objectives. The designer then has the opportunity to select an appropriate compromise solution from the trade-off family based on a subjective engineering knowledge of the required performance. For example, in this application, it would be expected that a tradeoff will exist between energy consumption and tracking performance. In this case, the designer may be willing to sacrifice a little energy efficiency to achieve a certain tracking metric. Individuals which represent candidate solutions to the optimisation problem (in this case fuzzy controller parameters such as membership functions, rule bases etc.) are encoded as either binary or real number strings, producing an initial population of chromosomes by randomly generating these strings. The population of individuals is evaluated using an objective function which characterises the individual's performance in the problem domain. The experimental system is run iteratively with each individual's set of controller parameters. The objective function determines how well each individual performs based on experimental data (in this case the current and velocity tracking performance and power consumption), and is used as the basis for selection via the assignment of a fitness value. Individuals which perform well are assigned a higher probability of

being selected for reproduction. Reproduction of individuals (usually in pairs) is achieved through the application of genetic operators, and the new individuals overwrite their parents in the population vector. The resulting new population contains material exchanged between the parents. Due to the stochastic nature of the GA as a search mechanism, a complete sweep of the global search space is achieved with more likelihood of finding the global minimum than conventional search methods. Whereas conventional methods require well-behaved objective functions, GAs tolerate noisy, discontinuous and even time-varying function evaluations. The motivation in this case for combining GAs with fuzzy logic for control is to investigate a number of factors. Firstly, the design potential which can be gained by removing the need for knowledge solicitation to enable the fuzzy logic design. Secondly to reduce the design time. Thirdly to examine a method for introducing robustness into the fuzzy design. Finally to investigate and define an method for multiobjective controller design where an accurate system model is either unavailable, or runs extremely slowly, a limiting factor in the process of iterative evolutionary design.

1.2 Hardware overview

The application consists of a brushed DC permanent magnet field motor fed by a four quadrant DC chopper drive operating at 5kHz. Figure 1 shows a schematic of the on-line control system and hardware setup. The objective is to perform robust closed loop speed control on this motor. The drive motor is connected via a flexible coupling to a field wound DC load motor which itself is fed directly by a 200V DC supply. The disturbance torque from this load motor is independently controllable, based on the applied armature voltage. Current control is embedded in the INTEL 80C196KC microcontroller as is the fuzzy logic velocity controller. The microcontroller also hosts the velocity and current feedback signals from the motor set and chopper drive respectively. The multiobjective optimisation programme runs under *Matlab* [18], and resides on a PC. Candidate controllers are downloaded from this host to the microcontroller via the serial link and on-line debug facility allowing direct access to programme memory. Assessment of the candidate controllers is performed on the PC according to a pre-programmed performance cost function. A National Instruments data acquisition board performs signal acquisition to bring feedback signals into the PC, to facilitate performance evaluation via the objective function.

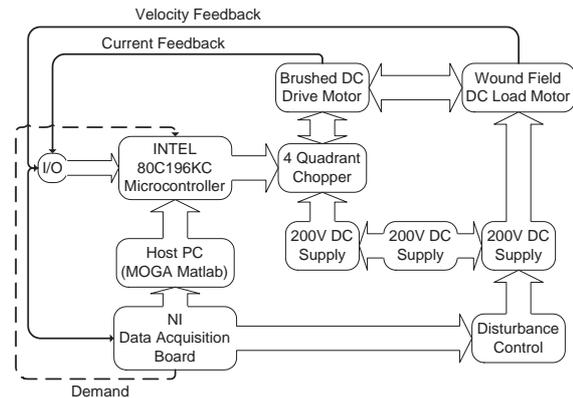


Fig. 1. Online optimisation hardware setup

2. OFF-LINE FUZZY LOGIC CONTROLLER DESIGN

A fuzzy logic velocity control scheme had been developed for this system previously in order to investigate the implementation issues involved with this type of control structure. Although claims are made concerning the reduction of development time [19], in fact the development time to produce the fuzzy controller off-line was significantly greater than the time required to manually produce and tune a robust PID tracking controller, a factor which is exacerbated by the complexity of the design procedure. The designer must choose input and output membership functions, a meaningful rule base, and an effective defuzzification strategy. In essence this requires the implementation of a controller with many degrees of freedom in the design, and consequently a complex implementation to achieve robust design.

An iterative design approach was utilised, to investigate the effects of the various degrees of design freedom in order to design the best controller. The most effective control structure was found to be input membership functions for error ($v(k)$) and change of error ($\Delta v(k)$) at time k , where

$$\Delta v(k) = v(k) - v(k-1) \quad (1)$$

The form of the membership function is shown in figure 2, The input functions are linked to the controller output by a rule base of the form;

- IF error is Positive Big THEN output is Positive Big
- IF error is Positive Small THEN output is Positive Small
- IF error is Zero THEN output is Zero
- IF error is Negative Small THEN output is Negative Small
- IF error is Negative Big THEN output is Negative Big

This rule base is repeated for change of error, and was implemented experimentally, the structure being shown in figure 3. The error and change

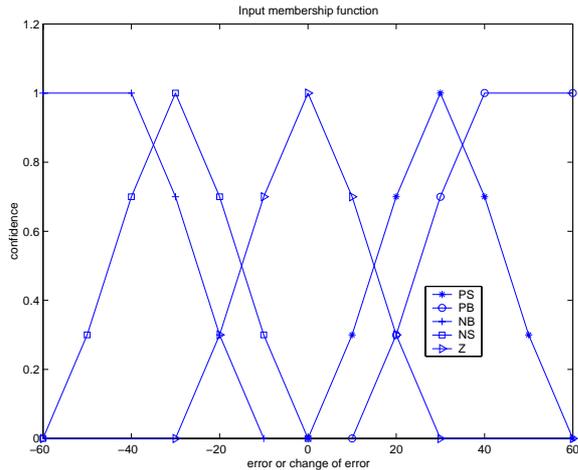


Fig. 2. Input membership functions for v and Δv

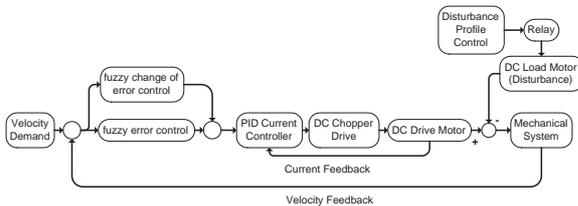


Fig. 3. Fuzzy controller implementation

of error controllers were constructed as follows. The fuzzy inference rule base is implemented using the intersection operator. A matrix of input and output sets included in each rule is constructed. Assuming for example, two classical sets A and B in a universe U , with membership functions μ_A and μ_B , then the minimum operator *intersection* can be defined as [19]

$$\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)) \quad (2)$$

The overall transfer surface for the controller was achieved by combining the matrix representation of all the individual rules into one overall matrix and applying the maximum operator *union*. This operation exemplifies the Cartesian cross product operator defined on n classical sets A_1, \dots, A_n as

$$\begin{aligned} X_{i=1}^n &= A_1 \times \dots \times A_n \\ &= ((x_1, \dots, x_n) | x_1 \in A_1, \dots, x_n \in A_n) \end{aligned} \quad (3)$$

The resulting transfer characteristic for velocity error is shown in figure 4. A corresponding surface consequently exists for change of velocity error. The utilisation of the centre of area defuzzification strategy [19] results in a controller structure shown in (figure 5). The surface provides a nonlinear relationship between velocity error, change of velocity error, and the controller output.

2.1 Results of off-line fuzzy logic controller design

The performance of the off-line designed fuzzy logic velocity controller is presented in figure 6

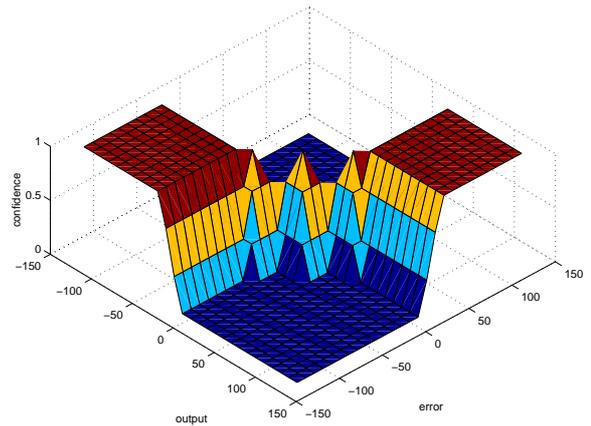


Fig. 4. Fuzzy transfer surface for velocity error

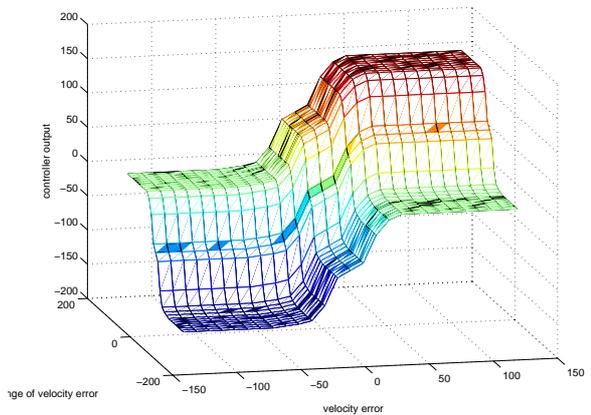


Fig. 5. Fuzzy controller output

for the non-disturbance case, and figure 7 for the case with external disturbance. In this case, a bi-directional velocity demand is supplied to the controller. In both the disturbed and undisturbed state, velocity tracking is comparable both in terms of rise time and steady state accuracy to a standard PID controller. Although it is beyond the central remit of this paper, a substantial amount of time was spent selecting an appropriate defuzzification strategy and the selection of the input-output sets in order to achieve this tracking performance. Consequently, the investigation of an online fuzzy logic design becomes an attractive proposition which is described in the next section. The development for an automatic design scheme with hardware in the loop will be considered and experimentally tested.

3. ON-LINE FUZZY LOGIC CONTROLLER DESIGN

Evolutionary algorithms have been used to optimize various aspects of intelligent control systems. In particular, the algorithm can generate the fuzzy rulebase, and tune the parameters of the associated membership functions. The application of evolutionary algorithms to fuzzy optimisation

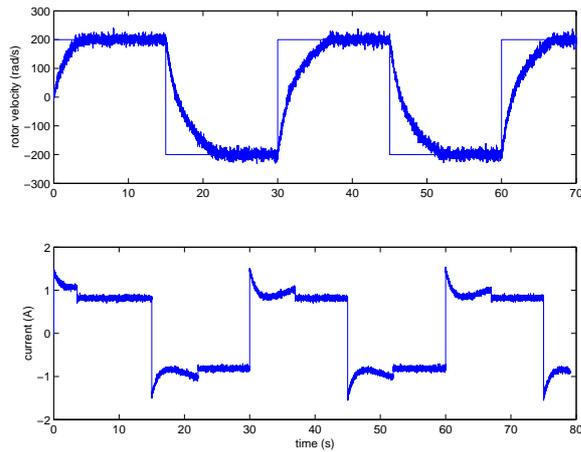


Fig. 6. Off-line designed fuzzy controller performance

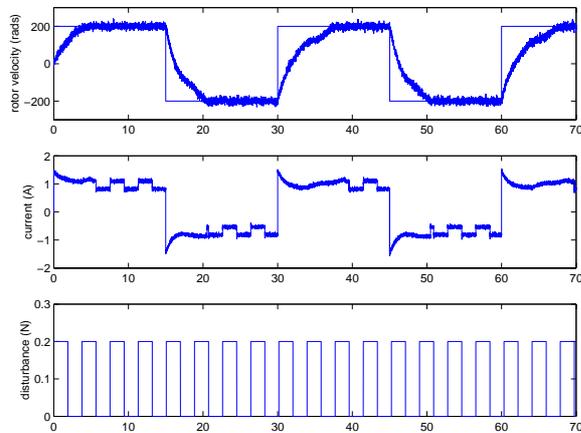


Fig. 7. Off-line designed fuzzy controller performance with external disturbance

is broadly split into two general areas; namely membership function tuning, and rulebase design with tuning. GA has been applied [14] to the off-line tuning of fuzzy membership functions, using a *fuzzy clustering* technique a fuzzy model was developed to describe the friction in a DC-motor system. In this case, the GA was seeded initially by the results obtained by fuzzy clustering. The results were greatly improved over those obtained by the non-tuned version. An *asynchronous* evolutionary algorithm has been used to generate membership functions to facilitate the rapid prototyping of fuzzy controllers [15]. This approach utilized parallel processing, being implemented on a 512 processor CM-5 Connection Machine. The application in question was a simulated space-based oxygen production system. Evolutionary methods have also been used where the derivation of an obvious set of fuzzy rules is not immediately apparent. In this case, the designer may either pre-specify a number of rules, or allow the number of rules to become an extra degree of freedom in the design. In all cases, the computational intensiveness of the designed optimisation technique must

be borne in mind, particularly in the case of on-line optimisation.

Due to the considerable computational and experimental considerations implicit in this method, certain constraints are included in the bounds of the decision variable vector in order to bring the automatic design time down to a reasonable level. A flowchart of the experimental setup is shown in

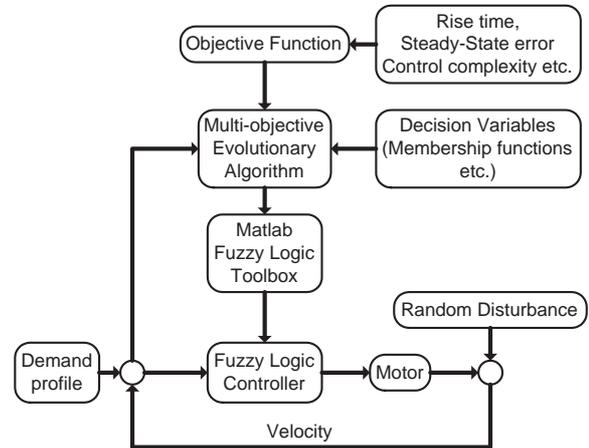


Fig. 8. On-line Fuzzy Logic design setup

figure 8 and contains a number of elements;

- Objective function

The objective function contains the elements of performance and design to be minimised, including rise-time, steady-state error, power utilisation and control complexity.

- Decision variables

The decision variable vector contains the elements of controller design which are implemented in each individual during the evolutionary process. The decision variables include the number of inputs, number of membership functions for each input and output, number of rules in the rule base, and-or-ignore conjugates in each rule, and finally the defuzzification algorithm. The selected values in the decision variables vector are passed to the Matlab Fuzzy Logic Toolbox to be constructed into a controller file. In order to reduce the necessary execution time to converge to a satisfactory conclusion the decision variable vector is bounded as follows

- number of inputs: 1-2
- number of membership functions for each input 3-5
- membership functions limited to triangular, with 2 base and one peak co-ordinate
- number of rules: 3-5
- conjugates: and, or, none
- defuzzification: centre of maximum

In addition, a random $\pm 0.2Nm$ disturbance is injected during each experimental run to in-

roduce an element of robustness into the design procedure. For each iteration of the design, the fuzzy controller was run on the motor rig and its performance ranked. It was found that the selected controller appeared early on in the procedure (generation 17 in a population of 10), in an initial run of 50 generations. The Pareto-Optimal set of solutions included several configurations and combinations of membership functions, including one which was markedly similar to the solution defined by the off-line fuzzy design with on-line tuning. The solution chosen for presentation here however, exhibits the required dynamic and steady-state performance but is coupled with a minimal set of membership functions (comprising an additional objective) and rules which presents computational advantages.

3.1 Results of on-line fuzzy logic controller design

The first results to present are those which show the dynamic and steady state performance of the velocity controller. The undisturbed case is shown in figure 9, and the disturbed case in figure 10 In

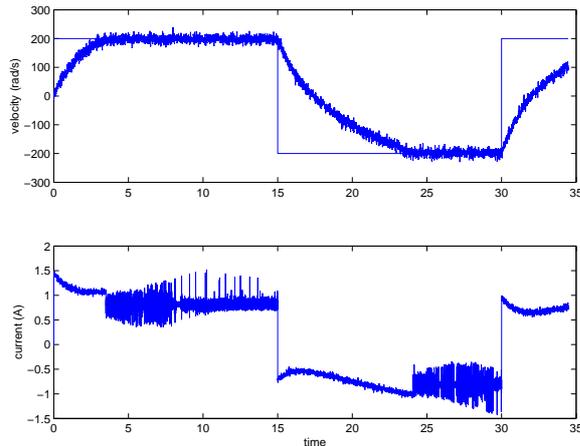


Fig. 9. On-line designed Fuzzy Logic velocity controller performance

both cases, the velocity tracking response of the system is comparable with earlier designs achieved by off-line fuzzy logic control design. One difference of particular interest is the current waveform in both cases which exhibits high frequency components. This effect has been commented upon [20] in the context of fuzzy logic control design, concluding that some off-line or on-line tuning is necessary to eliminate or effectively reduce the harmonics. In the case of the off-line fuzzy logic controller described earlier in this paper, the harmonics were reduced by on-line tuning. For future work in this case, the addition of frequency analysis to the objective function to minimise the unwanted harmonics would be a beneficial area of research. Hardware and computational constraints

precluded the implementation of this analysis on-line at this time, but it is intended that the investigation of this phenomenon on an upgraded rig be performed at some future time. Although

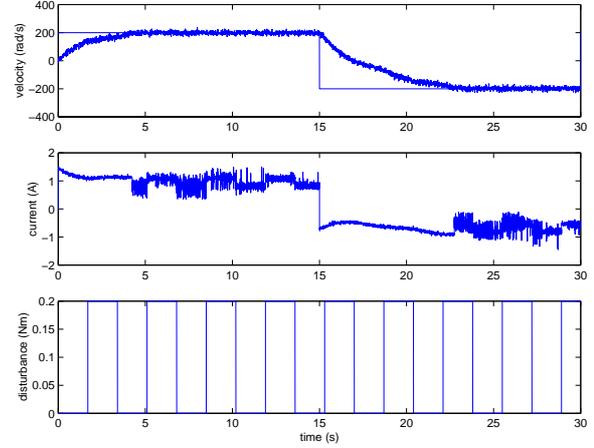


Fig. 10. On-line designed Fuzzy Logic velocity controller performance with disturbance

the performances of the various controllers are very similar, the structure of the on-line and off-line designed controllers are very different. Both have similar rule bases, but whereas the off-line design has inputs of both error and change-of-error, the automatically designed controller solely acts on error input. The membership functions

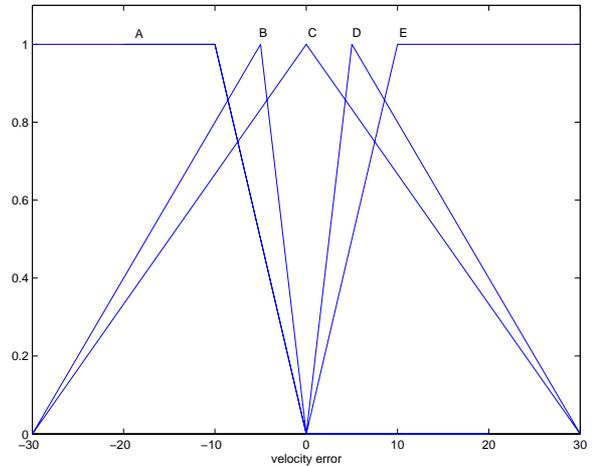


Fig. 11. On-line designed Fuzzy Logic velocity controller input membership functions. A:negbig, B:negsmall, C:zero, D:possmall, E:posbig.

which make up the input set are shown in figure 11, being the same number (5) as in the off-line designed case, but are far more closely clustered around the zero set. The membership functions which make up the output set are shown in figure 12 and are linked to the input set by the rule base;

- if velocity error is *negbig* THEN current demand is *negbig*
- if velocity error is *negsmall* THEN current demand is *negsmall*

- if velocity error is *zero* THEN current demand is *zero*
- if velocity error is *posbig* THEN current demand is *posbig*
- if velocity error is *possmall* THEN current demand is *possmall*

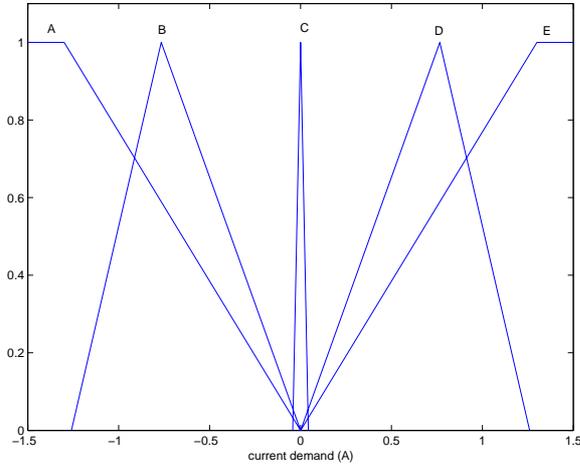


Fig. 12. On-line designed Fuzzy Logic velocity controller output membership functions. A:negbig, B:negsmall, C:zero, D:possmall, E:posbig.

The methods attached to the fuzzy logic controller were as follows;

- and:min
- or:max
- implication:min
- aggregation:max
- defuzzification:mom

4. CONCLUSIONS

The primary objective of this work, to assess the feasibility of automatically designing fuzzy logic controllers on-line with hardware in the loop has been demonstrated. A hardware platform previously intended for fuzzy logic design, formed the hardware in the loop since it was well characterised. The design of a fuzzy logic controller by traditional off-line methods had required manual tuning on line to maximise performance, and in particular, to reduce current harmonics introduced by the control action. It has been shown experimentally that on-line fuzzy logic controller design is feasible, and also that excellent dynamic and steady-state performance can be achieved. The design was optimised without the solicitation of knowledge because of the stochastic nature of the evolutionary optimisation algorithm which searches the multidimensional space of membership functions and rules for combinations which can achieve the performance specified in the objective function. Controller design based around models and simulation is often limited by the

veracity of the model under consideration. For example, electromagnetic actuators may be approximated by relatively simple expressions. However under certain circumstances, dynamic effects such as eddy currents, which are extremely difficult to model, need to be included in dynamic simulation. In this case, the differences between actual and simulated plant can make a significant difference to the controller performance. It appears that the on-line fuzzy controller design offers considerable advantages, and is worthy of serious consideration, also the possibility of injecting random disturbances during the design phase resulting in a controller capable of rejecting at least bounded disturbances shows particular promise. This topic together with consideration of the effects of controller dynamics on the harmonic content of the current waveforms will form part of a further investigation.

References

- [1] Mamdani E.H., Application of fuzzy algorithms for control of simple dynamic plant., *Proceedings of the IEE*, 121(12), (1974), 1585-1588.
- [2] Zadeh L.A., Outline of a new approach to the analysis of complex systems and decision processes, *IEEE Transactions on Systems, Man and Cybernetics*, 3(1973), 28-44.
- [3] Betin F., Pinchon D. and Capolino G.A., Control of electrical drives subject to large variations of load: a fuzzy logic approach., *Electromotion*, 8(3), July-September 2001, 155-168.
- [4] Guillemin P., Universal motor control with fuzzy logic, *Fuzzy Sets and Systems*, 63(3), May 1994, 339-348.
- [5] Hong T.P. and Lee C.Y., Induction of fuzzy rules and membership functions from training examples, *Fuzzy Sets and Systems*, 84(1), 1996, 33-47.
- [6] Ishibuchi H. and Tanaka H., Neural networks that learn from if-then fuzzy rules, *IEEE Transactions on Fuzzy Systems*, vol.1, 1993, 85-97.
- [7] Grauel A. and Mackenberg H., Mathematical analysis of the Sugeno controller leading to general design rules, *Fuzzy Sets and Systems*, 85(2), 1997, 165-175.
- [8] Chang C.H. and Wu Y.C., The genetic algorithm-based tuning method for symmetric membership functions of fuzzy logic control systems, *Proceedings of the International IEEE/IAS Conference on Industrial Automation and Control: Emerging Technologies.*, 1995, 421-428.
- [9] Homaifar A. and McCormick E., Simultaneous design of membership functions and rule sets for fuzzy controller using genetic algorithms, *IEEE Transactions on Fuzzy Systems*, 3(2), 1995, 129-139.

- [10] Thrift P., Fuzzy logic synthesis with genetic algorithms, *Proceedings of the 4th International Conference on Genetic Algorithms*, 1996, 279-283.
- [11] Wu C.J. and Liu G.Y. A genetic approach for simultaneous design of membership functions and fuzzy control rules, *Journal of Intelligent and Robotic Systems*, vol.28, 2000, 195-211.
- [12] Fonseca C.M. and Fleming P.J., An overview of evolutionary algorithms in multiobjective optimisation, *Evolutionary Computation*, 3(1), 1995, 1-16.
- [13] Fonseca C.M. and Fleming P.J., Multiobjective optimisation and multiple constraint handling with evolutionary algorithms - Part 1: A unified formulation and Part 2: Application example, *IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans*, 28(1), 1998, 26-37 and 38-47.
- [14] Tzes A., Peng P.Y. and Guthy J., Genetic-based fuzzy clustering for DC-motor friction identification and identification, *IEEE Transactions on Control Systems Technology*, 6(4), 1998, 462-472.
- [15] Kim J., Moon Y. and Ziegler B.P., Designing fuzzy net controllers using genetic algorithms, *IEEE Control Systems Magazine*, 15(3), 1995, 62-72.
- [16] Kuo B.C. and Hanselman D.C., Matlab tools for control system analysis and design, *Prentice-Hall International, New Jersey*, 1994, ISBN 0-13-099946-6.
- [17] Schroder P., Green B., Grum N. and Fleming P.J., On-line evolution of robust control systems: an industrial active magnetic bearing application., *IFAC Journal of Control Engineering Practice*, 9, 2001, 37-49.
- [18] Chipperfield A.J., Fleming P.J. and Pohlheim H.P., A genetic algorithm toolbox for MATLAB, *Proceedings of the International Conference on Systems Engineering*, 1994, pp.200-207, Coventry U.K.
- [19] Driankov D., Hellendoorn H. and Reinfrank M., An introduction to fuzzy control, *Springer-Verlag Berlin*, 1993, ISBN 3-540-56362-8.
- [20] Zhu Z.Q., Shen Z.X. and Howe D., Comparative study of alternative fuzzy logic control strategies of Permanent Magnet Brushless AC drive, *Proceedings of the 2002 International conference on Control Applications*, pp.42-47, September 18-20, 2002. Glasgow, Scotland, U.K.

Factorized Distribution Algorithms: Selection without selected population

Roberto Santana

Institute of Cybernetics, Mathematics, and Physics (ICIMAF)

Calle 15, e/ C y D, Vedado

CP-10400, La Habana, Cuba

rsantana@cidet.icmf.inf.cu

Abstract- In this paper we investigate the problem of an efficient implementation of the selection step in Factorized Distribution Algorithms. We demonstrate that while in Genetic Algorithms the selection operator needs the creation of a selected population, in Factorized Distribution Algorithms this is not always the case.

Keywords: Evolutionary algorithms, genetic algorithms, EDAs

1 Introduction

Selection is a corner stone of a number of population based search methods like Genetic Algorithms (GAs) [4, 3], and Estimation Distribution Algorithms (EDAs) [10]. These methods are non deterministic heuristic search strategies, commonly used for optimization. Their main characteristics are: They use a population of individuals or solutions, instead of a single point, to conduct the search. In every iteration (usually called generation) a subset of individuals is selected, and by applying some operators, a new population is created. In this way the algorithm iterates (evolves) until one stop condition is satisfied.

In GAs the recombination and mutation operators are applied to the selected set of individuals to obtain the new population. Other algorithms are characterized by the use of probabilistic modeling of the information contained in the selected set. In this paper we will use the term Estimation Distribution Algorithm to refer to any evolutionary algorithm that uses the estimation of probability distributions instead of the genetic operators. Although not all the proposals fit well in the EDAs scheme, this conceptual framework has been used before [5] as a model to study the algorithms under analysis. Algorithm 1 shows the steps of an EDA.

One efficient way of estimating a probability distribution is by means of factorizations. A probability distribution is factorized when it can be computed by a small number of factors. A subclass of EDAs will group the algorithms that use factorizations of the probability distribution. In this paper we call to this subclass

Algorithm 1: EDAs

```
1 Set  $t \leftarrow 0$ . Generate  $N \gg 0$  individuals randomly.
2 do {
3   Select a set  $S$  of  $k \leq N$  individuals according to a selection method.
4   Calculate a probabilistic model of  $S$ .
5   Generate  $N$  new individuals sampling from the distribution represented in the model.
6    $t \leftarrow t + 1$ 
7 } until Termination criteria are met
```

Factorized Distribution Algorithms (FDAs)¹ [9]. FDAs belong to the EDAs class as well as other evolutionary algorithms where the estimation of the distribution is achieved by other means. In this paper we show that the selection step in a FDA can be omitted when this step involves the calculation of the selection probabilities of all the individuals in the current population. The possibility of changing the way selection is accomplished represents an important difference with GAs.

Let $X = (X_1, \dots, X_n)$ be a tuple of random variables, and $X \in B^n$ where B^n is the finite n -dimensional binary space. Throughout this paper x will denote a value of the individual X , and x_i the value of X_i , the i -th component of X .

The paper is organized as follows: In the next section we review the main types of selections used by GAs. In section 3 we discuss the way selection is implemented in FDAs. We present a modification to the traditional way of doing selection. In section 4 we conduct a number of experiments to evaluate the impact of the proposed changes in the behavior of some FDAs. Finally, section 5 describes the relationship between selection and the other components of the FDAs, and presents the conclusions of our work.

¹In the literature the term FDA is frequently used to name a particular type of Factorized Distribution Algorithms. Our definition covers it, and other algorithms that use factorizations.

2 Selection in GAs

In our analysis we have used the classification of the selection methods proposed by Sastry and Goldberg [12]. Selection methods are classified in two classes: (1) Proportional schemes, and (2) Ordinal based schemes. Proportional schemes select an individual based on its relative fitness value compared to others. Ordinal schemes select an individual based on its ranking in the population.

In Proportional schemes selection is usually accomplished in two steps. First, the selection probabilities of the individuals in the current population are determined, then new individuals are sampled from these probabilities. These individuals form the selected set that will serve as a mating pool. Examples of these Proportional schemes are the Proportional [3] and Boltzmann [8] selection. In Ordinal based schemes, selection probabilities are not explicitly calculated. Instead, some procedure is used to select the individuals straight from the population. One example is the Tournament selection, where s individuals are randomly chosen from the population, and the best individual from this group is included in the selected set². This process is repeated until the selected set has been filled.

When Proportional schemes are applied, different sampling algorithms can be used to sample from the selection probabilities. The most known example is the Roulette Wheel (RW) selection [3], generally used for Proportional selection. In RW selection a biased roulette wheel is created where each current individual in the population has a roulette wheel slot sized in proportion to its fitness. Every time the wheel is spinned, a copy of the selected individual is included in the selected set. RW selection is also a clear example of a bad sampling method, for small populations it has a high variance.

To solve this problem other types of sampling methods, like the Stochastic Universal Sampling (SUS) [1], have been proposed. SUS consists of simultaneously selecting N individuals by locating N equally spaced pointers in the roulette. After only one spin, we select the individuals corresponding to the slots where the N pointers are located.

In table 1 an example of the application of the Proportional selection is shown. From a population of 7 individuals the selection probabilities are calculated based on the individuals' fitness. The expected count shown in column 4 is the expected number of copies of each individual (n^j), calculated as the product of the selection probabilities ($p(x^j)$) by the number of gener-

ated individuals.

$$n^j = p(x^j) \cdot N \quad (1)$$

Possible outcomes of the RW and SUS methods are presented in columns 5 and 6. Due to the finite size of the population none of the actual counts corresponds to the expected counts.

x	$f(x)$	$p^s(x^j)$	n^j	n^j_{RW}	n^j_{SUS}
10100	2	0.1	0.7	1	1
10110	2	0.1	0.7	0	1
11000	2	0.1	0.7	0	0
10011	3	0.15	1.05	1	1
10101	3	0.15	1.05	1	1
10111	4	0.2	1.4	2	1
11011	4	0.2	1.4	2	2

Table 1: Different steps of the Proportional selection method

3 Selection in FDAs

As there is no crossover operator in FDAs, the selected set is not used as a mating pool. These algorithms use the selected individuals to construct a probabilistic model that captures the interdependencies between the variables. Considering the complexity of the probabilistic models they use, FDAs can be classified in two classes: (1) Algorithms that make a parametric learning of the probabilities, and (2) Algorithms where a structural learning of the model is done. In parametric, as well as in structural learning, the only relevant information extracted from the data is initially represented in the marginal probabilities of the variables. Structural learning finds a graphic representation of the interactions among the variables using these marginals. Parametric and structural learning are also known as model fitting and model selection. The interested reader is referred to [5] for a review of probabilistic graphical modeling and EDAs.

3.1 Proportional Schemes

Traditional implementations of FDAs that use Proportional schemes first determine the set of selected solutions, and calculate then the marginal probabilities from this set. We analytically show that avoiding the creation of the selected set is possible. To present our case we use as an example of FDAs the Univariate Marginal Distribution Algorithm (UMDA) [10]. The UMDA uses a very simple probabilistic model that assumes all the variables are independent. The

²Tournament selection can be done with or without replacement of the selected individual.

probability of an individual x is estimated as $p(x) = \prod_{i=1}^n p_i^s(X_i = x_i, t)$, where $p_i^s(X_i = x_i, t)$ are the univariate probabilities calculated from the selected population. In the case of the binary problems we treat in this paper, we will use $p_i^s(x_i, t)$ as a shortcut for $p_i^s(X_i = 1, t)$.

Vars./Univ.	p_i	p_i^{sRW}	p_i^{sSUS}	\hat{p}_i^s
$x1$	1.0	1.0	1.0	1.0
$x2$	0.286	0.286	0.286	0.3
$x3$	0.571	0.571	0.571	0.55
$x4$	0.571	0.714	0.714	0.65
$x5$	0.571	0.857	0.714	0.70

Table 2: Univariate Probabilities

In table 2 we show the univariate probabilities corresponding to the selected populations presented in table 1. Column 2 shows the univariate marginals of the initial population, calculated assuming that there is only one copy of the 7 individuals. Columns 3 and 4 show the univariate marginals calculated from the populations obtained using RW and SUS (columns 5 and 6 in table 1). The last column corresponds to the univariate probabilities calculated from the expected counts, we call them expected univariate probabilities $\hat{p}_i^s(x_i, t)$.

$$\hat{p}_i^s(x_i, t) = \frac{\sum_{j=1}^N (n^j \cdot x_i^j)}{\sum_{j=1}^N n^j} \quad (2)$$

Substituting equation (1) in (2), and considering $\sum_j p^s(x^j) = 1$:

$$\begin{aligned} \hat{p}_i^s(x_i, t) &= \frac{N \sum_{j=1}^N (p^s(x^j) \cdot x_i^j)}{N \cdot \sum_{j=1}^N p^s(x^j)} \\ &= \sum_{j=1}^N (p^s(x^j) \cdot x_i^j) \\ &= p_i^s(x_i) \end{aligned} \quad (3)$$

Notice in table 2 that the univariate probabilities of the selected population obtained using SUS are a better approximation of the expected univariate probabilities, than those calculated from the selected population obtained using RW selection. Nevertheless, both approximations have a departure from $\hat{p}_i^s(x_i, t)$. This difference is due to the sample methods used, and to the fact of using a small population. Equation (2) shows that for calculating the expected univariate probabilities there is no need of applying any sample method, they can be calculated from the expected counts. Even more, as it is observed in equation (3), no even the expected counts are needed.

When Proportional schemes are applied to FDAs it is possible to avoid the step of selecting a set of individuals. The marginal probabilities can be calculated straight from the selection probability distribution determined by the selection method.

If parametric learning is used the learning process finishes once the marginals have been calculated. There is no need to create a selected population, but if a good sampling method like SUS were applied to create the selected population, the obtained marginals would be close to the expected ones, just as in the example described in tables 1 and 2. The only difference in the case of structural learning is that the marginals will be also used for learning the structure of the probabilistic model.

3.2 Ordinal Based Schemes

For Ordinal based schemes, for which selection probabilities are not calculated, the model learning process has necessarily to be done using a selected set of individuals, instead of the selection probabilities. This is the case of Tournament selection. However, in these cases it can still be convenient learning the model from a probability distribution. An algorithm for the case of Ordinal based schemes would have the following steps:

1. Create the selected population.
2. Create a 'compact' population from the selected population where all the individuals are different among each other.
3. Associate to each individual in the compact population its probability in the selected population.
4. Learn the probabilistic model from the probabilities associated to the individuals in the compact population.

In table 3 an example of how to construct a compact population is shown. It is more efficient to do the model learning step when there is only one copy of each individual. Furthermore, by calculating the frequency of each individual in the selected population we can control early convergence of the FDA, and stop the algorithm when a few numbers of individuals dominate the selected population.

4 Experiments

In section 3.1 we have analytically shown that the step of creating a selected population is not always needed by the FDAs. Thus, the goal of our experiments is not to investigate the validity of this result. Instead, we are

Selected Pop.	Compact Pop.	Prob.
10100	10100	0.3
10110	10110	0.1
10100	00100	0.2
00100	01001	0.1
01001	11000	0.2
11000	01010	0.1
01010		
11000		
10100		
00100		

Table 3: Construction of a compact population

interested to find out which is the influence of using a (as shown before redundant) selected population in the behavior of the FDAs that work with Proportional schemes. The goal is to measure the difference between the two selection procedures: when a selected population was used (SP), or when this was not the case (No SP). We will be concerned with the maximization of a function $f : X \rightarrow R^{\geq 0}$.

An important point is that the results shown below are by no means the best that can be achieved using the FDAs presented. Results with Truncation selection can be much better than those achieved with Proportional selection. On the other hand, the performance of Boltzmann selection can be considerably improved when an adaptive Boltzmann selection schedule is incorporated to the FDA [6]. Similarly, results of the comparison between the different FDAs must not be taken as conclusive.

4.1 Functions Used in the Experiments

Most of the functions used in our experiments belong to the class of Additive Decomposable Functions (ADFs), for which the value of the function is the sum of the evaluation of a number of sub-functions in subsets of variables. They represent an interesting class of functions where possible interactions among the variables are reduced to a subset of them, and thus are used to simulate problems that can be decomposed in smaller subproblems. The functions presented have served before as a test bed for evolutionary algorithms. The interested reader is referred to [5] for an account of the performance of other FDAs for these functions.

The simplest additive function is *OneMax* where each sub-function is evaluated in only one variable.

Function *OneMax*:

$$OneMax(x) = \sum_{i=1}^n x_i \quad (4)$$

Functions of unitation are used to define the sub-functions comprised by an ADF. A function of unitation is a function whose value depends only on the number of ones in an input string. The function values of the strings with the same number of ones are equal. Thus, functions of unitation can be defined in terms of the unitation of the individual (u). The $f_{3deceptive}$ function is defined as a sum of the more elementary unitation function f_{dec}^3 .

Function $f_{3deceptive}$:

$$f_{3deceptive}(x) = \sum_{i=1}^{\frac{n}{3}} f_{dec}^3(x_{3i-2}, x_{3i-1}, x_{3i})$$

where

Function f_{dec}^3 :

$$f_{dec}^3(u) = \begin{cases} 0.9 & \text{for } u = 0 \\ 0.8 & \text{for } u = 1 \\ 0.0 & \text{for } u = 2 \\ 1.0 & \text{for } u = 3 \end{cases} \quad (5)$$

Function general deceptive of order k , f_{decK} :

$$f_{decK}(u, k) = \begin{cases} k-1 & \text{for } u = 0 \\ k-2 & \text{for } u = 1 \\ \dots & \dots \\ k-i-1 & \text{for } u = i \\ \dots & \dots \\ k \cdot n & \text{for } u = k \end{cases} \quad (6)$$

Function $f_{deceptivek}$:

$$f_{deceptivek}(x) = \sum_{i=1}^{\frac{n}{k}} f_{decK}(x_{ki-k+1}, \dots, x_{ki}) \quad (7)$$

Function *Checkerboard*:

The goal of the checkerboard problem is to create a checkerboard pattern of 0's and 1's in an $N \times N$ grid. Only the primary four directions are considered in the evaluation. For each position in an $(N-2) \times (N-2)$ grid centered in an $N \times N$ grid, 1 is added for each of the four neighbors that are set to the opposite value. The maximum evaluation for the function is $4(N-4)(N-4)$.

Function *symmetric*:

$$symmetric(u) = \begin{cases} u & \text{if } 2 \cdot u < n \\ n - u & \text{otherwise} \end{cases} \quad (8)$$

4.2 FDAs Used in the Experiments

The FDAs that were chosen for the experiments are briefly described below. Our choice was due to the possibility of incorporating the proposed changes to the available implementations of these algorithms.

1. The UMDA, which is a FDA with a simple probabilistic model that assumes variables are independent. UMDA generates new solutions by only preserving the proportions of the values of a variable, independently of the values for the remaining variables. This approach can work well even for problems where variables are not completely independent.
2. A tree based FDA (Tree-FDA), this algorithm is similar to the MIMICs version presented in [2], where a tree shaped network approximates the joint distribution. It has been used before in the optimization of binary problems, as well as of functions defined on integers.
3. A mixture of trees FDA (MT-FDA) [11], where a mixtures of trees is used to represent the distribution. Mixtures of trees can represent, condensed in just one model, different patterns of interactions among the variables of the problem.

4.3 Numerical Results

n	N	SP			No SP		
		s	eval.	f	s	eval.	f
12	12	46	45.2	11.39	50	43.7	11.39
12	24	98	73.8	11.98	97	69.8	11.97
12	30	96	81.6	11.95	100	86.3	12.00
36	36	18	214.9	34.31	26	232.5	34.59
36	80	85	409.9	35.83	85	397.9	35.82
36	100	91	466.6	35.91	98	483.9	35.98
100	500	47	3886.8	98.95	48	3826.7	98.76
100	800	70	5810.9	99.56	77	5697.8	99.60
100	1000	81	6920.0	99.77	85	6994.0	99.93

Table 4: Comparison of the two selection procedures for the Boltzmann selection using the UMDA in the optimization of the *OneMax* function

Every experiment consists of 100 runs of the algorithm for the given parameters. The number of times the algorithm found the optimum (s), the average number of functions evaluations (eval.), and the average fitness of the function (f) are calculated from these experiments. As two different criteria for comparison

between SP and No SP we take the number of times the optimum was reached in the 100 runs, and the average fitness. When for one of these criteria the algorithms are tied, the algorithm with the minimum number of average function evaluations is considered as the winner.

Table 4 shows the results of the optimization of function *OneMax* using the UMDA, when the Boltzmann selection is used with base $b = e$. Results are presented for different number of variables and population sizes. The maximum number of generations was 50. As it can be seen in the table, if we consider the times the optimum was reached No SP was the best 8 times, and SP was the best only once. Considering the average fitness, No SP was better than SP 6 times, and SP was the best the other 3 times. In general, the differences in the results are not significant.

FDA	$Prior$	Mut	SP		
			s	eval.	f
Tree-FDA	<i>no</i>	<i>no</i>	56	940.3	3.95
MT-FDA	<i>no</i>	<i>no</i>	45	1633.4	3.94
Tree-FDA	<i>no</i>	<i>yes</i>	72	1606.9	3.97
MT-FDA	<i>no</i>	<i>yes</i>	67	2053.6	3.97
Tree-FDA	<i>yes</i>	<i>no</i>	93	2120.6	3.99
MT-FDA	<i>yes</i>	<i>no</i>	74	2095.0	3.97
FDA	$Prior$	Mut	No SP		
Tree-FDA	<i>no</i>	<i>no</i>	46	820.5	3.94
MT-FDA	<i>no</i>	<i>no</i>	45	1494.3	3.94
Tree-FDA	<i>no</i>	<i>yes</i>	68	1789.0	3.97
MT-FDA	<i>no</i>	<i>yes</i>	55	2225.2	3.95
Tree-FDA	<i>yes</i>	<i>no</i>	88	2324.0	3.99
MT-FDA	<i>yes</i>	<i>no</i>	70	2146.6	3.96

Table 5: Comparison of the two selection procedures for the Proportional selection, using the Tree-FDA and the MT-FDA in the optimization of the $f_{3deceptive}$ function under different conditions

Table 5 shows results of the two selection procedures for Proportional selection, using the Tree-FDA and the MT-FDA. In all the cases the function used was $f_{3deceptive}$, $n = 12$. Column 2 refers to the use of probabilistic priors during the optimization procedure. The use of priors has been proposed in [7] to enhance the performance of FDAs, and to accelerate their convergence. Column 3 refers to the traditional mutation operator used by GAs. In this case we have used a mutation rate of 0.02.

The idea of this experiment is to study if there exist significant differences in the behavior of the algorithms when adding a perturbation to the population (in this example by means of priors and mutation). Consider-

<i>Function</i>	<i>n</i>	<i>N</i>	<i>FDA</i>	SP			No SP		
				s	eval.	<i>f</i>	s	eval.	<i>f</i>
<i>Checkerboard</i>	25	300	UMDA	93	8739.5	35.87	81	9052.2	35.64
			Tree-FDA	91	6174.9	35.91	93	6517.9	35.93
			MT-FDA	93	10620.3	35.93	84	11548.1	35.83
<i>f_{deceptive3}</i>	24	300	UMDA	8	8672.0	22.08	7	7476.0	21.89
			Tree-FDA	73	5993.3	23.67	63	6185.1	23.57
			MT-FDA	53	11007.6	23.41	53	11018.9	23.43
<i>symmetric</i>	24	200	UMDA	99	6334.8	23.99	100	6576.0	24.00
			Tree-FDA	90	4830.1	23.88	89	4680.9	23.88
			MT-FDA	87	8278.9	23.86	86	8590.4	23.84

Table 6: Comparison of the two selection procedures for the Proportional selection using different FDAs and functions

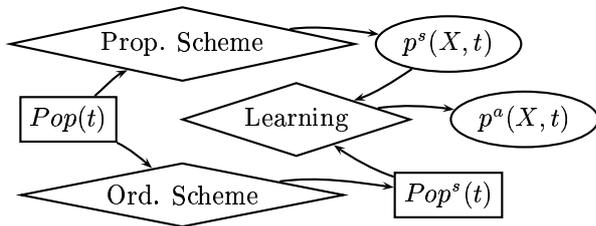


Figure 1: Alternative ways for the selection operator in FDAs. $Pop(t)$: populations at time t ; $p^s(X, t)$, $p^a(X, t)$: Join probabilities determined by selection and the graphical model approximations.

ing the number of times the optimum was reached, SP was the best in 5 of the 6 experiments. The same happens if the average fitness criterion is considered. We hypothesize that in this case the stochastic noise caused by the use of SUS contributed to infuse a convenient diversity to the generated population.

Table 6 presents the results for functions $f_{deceptive3}$, *Checkerboard*, and *symmetric*. In this experiment a maximum of 60 generations was allowed. Considering the number of times the optimum was reached SP was the best in 7 of the 9 experiments. The picture is less clear if we analyze the average fitness because SP was the best only in 5 of the 9 cases.

5 Conclusions

The main result achieved in this paper is to show that new possibilities of doing selection arise when crossover operators are replaced by probabilistic modeling of the solutions. Basically, we have proven that when using Proportional and Boltzmann selection, no selected pop-

ulation needs to be constructed in order to generate the individuals in the next generation, according to the selection probabilities. In figure 1 we present how Proportional and Ordinal based selection schemes can be inserted in FDAs, and related with their other components.

We have conducted a number of experiments to evaluate what is the influence of using sampling methods like SUS when they are not needed. Our preliminary results show that the use of SUS, although redundant, does not significantly deteriorate the results, and in some cases can improve them. This is probably due to the side effect that the random noise associated to the procedure for sampling the selection probabilities can have in introducing diversity to the search. In this paper it has been shown that the learning of the probabilistic model from the joint selection probability distribution can be done also for Ordinal based schemes. We expect that our results have shed some light to the differences that exist between GAs and FDAs.

Bibliography

- [1] J. E. Baker. Reducing bias and inefficiency in the selection algorithm. In *Proceedings of the Second International Conference on Genetic Algorithms*, pages 14–21. Lawrence Erlbaum Associates (Hillsdale), 1987.
- [2] S. Baluja and S. Davies. Using optimal dependency-trees for combinatorial optimization: Learning the structure of the search space. In *Proceedings of the 14th International Conference on Machine Learning*, pages 30–38. Morgan Kaufmann, 1997.

- [3] D. E. Goldberg. *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley, Reading, MA, 1989.
- [4] J. H. Holland. *Adaptation in natural and artificial systems*. University of Michigan Press, Ann Arbor, MI, 1975.
- [5] P. Larrañaga and J. A. Lozano. *Estimation Distribution Algorithms. A new tool for Evolutionary Optimization*. Kluwer Academic Publishers, Boston/Dordrecht/London, 2001.
- [6] T. Mahnig and H. Mühlenbein. Comparing the adaptive Boltzmann selection schedule SDS to truncation selection. In *Evolutionary Computation and Probabilistic Graphical Models. Proceedings of the Third Symposium on Adaptive Systems (ISAS-2001)*, pages 121–128, Habana, Cuba, March 2001.
- [7] T. Mahnig and H. Mühlenbein. Optimal mutation rate using Bayesian priors for Estimation of Distribution Algorithms. In K. Steinhöfel, editor, *Proceedings of the First Symposium on Stochastic Algorithms: Foundations and Applications, SAGA-2001*, volume 2264 of *Lecture Notes in Computer Science*, pages 33–48. Springer, 2001.
- [8] H. Mühlenbein and T. Mahnig. Convergence theory and applications of the Factorized Distribution Algorithm. *Journal of Computing and Information Technology*, 7(1):19–32, 1998.
- [9] H. Mühlenbein, T. Mahnig, and A. Ochoa. Schemata, distributions and graphical models in evolutionary optimization. *Journal of Heuristics*, 5(2):213–247, 1999.
- [10] H. Mühlenbein and G. Paaß. From recombination of genes to the estimation of distributions I. Binary parameters. In A. Eiben, T. Bäck, M. Shoenauer, and H. Schwefel, editors, *Parallel Problem Solving from Nature - PPSN IV*, pages 178–187, Berlin, 1996. Springer Verlag.
- [11] R. Santana, A. Ochoa, and M. R. Soto. The Mixture of Trees Factorized Distribution Algorithm. In L. Spector, E. Goodman, A. Wu, W. Langdon, H. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M. Garzon, and E. Burke, editors, *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-2001*, pages 543–550, San Francisco, CA, 2001. Morgan Kaufmann Publishers.
- [12] K. Sastry and D. E. Goldberg. Modeling tournament selection with replacement using apparent added noise. In *Intelligent Engineering Systems Through Artificial Neural Networks. Proceedings of the Conference ANNIE 2001*, volume 2, pages 129–134, 2001.

A Markov Network based Factorized Distribution Algorithm for optimization

Roberto Santana

Institute of Cybernetics, Mathematics, and Physics (ICIMAF)

Calle 15, e/ C y D, Vedado

CP-10400, La Habana, Cuba

rsantana@cidet.icmf.inf.cu

Abstract- In this paper we propose a population based optimization method that uses the estimation of probability distributions. To represent an approximate factorization of the probability, the algorithm employs a junction graph constructed from an independence graph. We show that the algorithm is able to extend the representation capabilities of previous algorithms that use factorizations. A number of functions are used to evaluate the performance of our proposal. The results of the experiments show that the algorithm is able to optimize the functions, and it overperforms other evolutionary algorithms that use factorizations.

Keywords. Genetic algorithms, EDA, FDA, evolutionary optimization, estimation of distributions.

1 Introduction

In the application of Genetic Algorithms (GAs) [8, 6] to a wide class of optimization problems is essential the identification and mixing of building blocks. It has been early noticed that the Simple GA (SGA) is in general unable to accomplish these two tasks for difficult problems (e.g. deceptive problems). Perturbation techniques, linkage learners and model building algorithms are among the alternatives proposed to improve GAs. They try to identify the relevant interactions among the variables of the problem, and to use them in an efficient way to search for solutions.

Model building techniques refer to GAs that construct a probabilistic model of the solutions instead of the crossover operator. In this paper we use the term Estimation Distribution Algorithms (EDAs) [16] to call this type of algorithms. These algorithms construct in each generation a probabilistic model of the selected solutions. The probabilistic model must be able to capture a number of relevant relationships in the form of statistical dependencies among the variables. Dependencies are then used to generate solutions during a sampling step.

It is expected that the generated solutions share a number of characteristics with the selected ones. In this

way the search is led to promising areas of the search space. EDAs are also known as Iterated Density Estimators Evolutionary Algorithms [1], and Probabilistic Model Building Genetic Algorithms [20]. The interested reader is referred to [9] for a good survey that covers the theory, and a wide spectrum of EDAs applications.

One efficient way of estimating a probability distribution is by means of factorizations. A probability distribution is factorized when it can be computed by a small number of factors. A subclass of EDAs will group the algorithms that use factorizations of the probability distribution. In this paper we call to this subclass Factorized Distribution Algorithms (FDAs)¹ [15].

FDAs have outperformed other evolutionary algorithms in the optimization of complex additive functions, and deceptive problems with overlapping variables [15]. They have been recently applied as well for the solution of the bipartitioning [14], and satisfiability problems [18]. However, a shortcoming of FDAs is that the probabilistic model they are based on is constrained to represent a limited number of interactions. In this paper we investigate the issue of extending the representation capabilities of FDAs. To this end we introduce the Markov Network FDA (MN-FDA), a new type of FDA based on an undirected graphical model, and able to represent the so called "invalid" factorizations [15].

The paper is organized as follows. In section 2 we discuss the problem of obtaining a factorization of the probability. Section 3 presents the main steps for learning an approximate factorization from data. Section 4 explains the way the sampling step has been implemented. We introduce the MN-FDA in section 5. Section 6 presents the functions used in our experiments. We discuss the numerical results of the simulation. Section 7 analyzes the MN-FDA in the context of recent related research on evolutionary computation, we also present in this section the conclusions of our paper.

¹In the literature the term FDA is frequently used to name a particular type of Factorized Distribution Algorithms. Our definition covers it, and other algorithms that use factorizations.

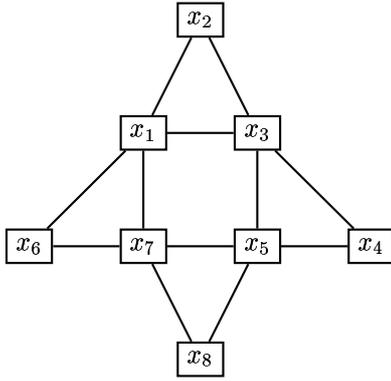


Figure 1: Independence graph

2 Factorization of a probability

The central problem of FDAs is how to efficiently estimate a factorization of the joint probability of the selected individuals. To compute a factorization the theory of graphical models is usually employed. One example of graphical models are Bayesian networks, where the dependencies relationships between the variables of the problem are represented using directed graphs. A number of Bayesian FDAs have been proposed in the literature [5, 19, 13]. We will focus on another type of graphical representation based on undirected graphs. The following definitions will help in the explanation of our proposal.

Let $X = (X_1, X_2, \dots, X_n)$ represent a vector of integer random variables, where n is the number of variables of the problem. $x = (x_1, x_2, \dots, x_n)$ is an assignment to the variables, and $p(x)$ is a joint probability distribution to be modeled. Each variable of the problem has associated one vertex in an undirected graph $G = (V, E)$. The graph G is a conditional independence graph respect to $p(x)$ if there is no edge between two vertices whenever the pair of variables is independent given all the remaining variables.

Definition 1. Given a graph G , a clique in G is a fully connected subset of V . We reserve the letter C to refer to a clique. The collection of all cliques in G is denoted as \mathcal{C} . C is maximal when it is not contained in any other clique. C is the maximum clique of the graph if it is the clique in \mathcal{C} with the highest number of vertices.

Definition 2. A junction graph (JG) of the independence graph G is a graph where each node corresponds to a maximal clique of G , and there exists an edge between two nodes if their corresponding cliques overlap.

Definition 3. A junction tree (JT) is a single connected junction graph. It satisfies that if the variable X_k is a member of the junction tree nodes i and j ,

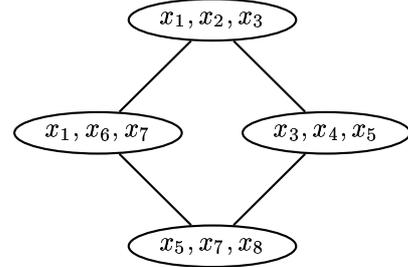


Figure 2: Associated junction graph

then X_k is a member of every node on the path between i and j . This property is called the running intersection property.

Figures 1 and 2 respectively show an example of an independence graph, and its associated junction graph.

If the independence graph G is chordal, an exact factorization of the probability based on the cliques of the graph exists. The factorization can be represented using a JT . If G is not chordal, a chordal supergraph of G can be found by adding edges to G in a process called triangulation. The problem is that we can not guarantee that the maximum clique of the supergraph will have a size that would make feasible the calculation of the marginal probabilities. The problem of finding a triangulation with maximum clique of minimum size is NP-complete.

To obtain an exact factorization of the probability is usually an infeasible task. In such cases we can use approximate factorizations. This is the approach we follow. Our goal is to find an approximate factorization that contains as many dependencies as possible, but without adding new edges to the graph. An exact factorization would comprise all the dependencies represented in the independence graph. We will assume that approximate factorizations of the probability are more precise as they include more of the dependencies represented in the independence graph.

The approximate factorization will be represented using a labeled JG . The algorithm for learning the probabilistic model has four main steps.

Algorithm 1: Model learning

- 1 Learn an independence graph G from the data (the selected set of solutions).
- 2 Find the set L of all the maximal cliques of G .
- 3 Construct a labeled JG from L .
- 4 Find the marginal probabilities for the cliques in the JG .

3 Learning of approximate factorization

In this section we consider in detail the different steps for learning a factorization from data.

3.1 Learning of an independence graph

The construction of an independence graph from the data can be accomplished by means of independence tests. To determine if an edge belongs to the graph, it is enough to make an independence test on each pair of variables given the rest. Nevertheless, from an algorithmic point of view it is important to reduce the order of the independence tests. Thus, we have adopted the methodology followed previously by Spirtes [22]. The idea is to start from a complete undirected graph, and then try to remove edges by testing for conditional independence between the linked nodes, but using conditioning sets as small as possible.

To evaluate the independence tests we use the Chi-square independence test. If two variables X_i and X_j are dependent with a 75 percent of significance the corresponding X^2 value of the Chi-square test is used to assign a weight $w(i, j)$ to the edge $i \sim j$. This weight stresses the pairwise interaction between the variables. When the independence graph is known in advance, we assume $w(i, j) = w', \forall i \sim j \in E$. The weight of any subgraph G' of G is calculated as $w(G') = \sum_{i \sim j \in G'} w(i, j)$. In this way the weights of the maximal cliques can be calculated.

3.2 Maximal cliques of graph

To find all the cliques of the graphs the Ken and Kerbash algorithm [2] is used. This algorithm uses a branch and bound technique to cut off branches that can lead to cliques. Its memory requirements are at most $\frac{1}{2} \cdot M \cdot (n + 3)$ storage locations to contain arrays of integers, where M is the size of the largest connected component in the input graph [2]. When the undirected graph is very dense, it can be made sparser before the calculation of the cliques of the graph by removing the edges with lower weights.

3.3 Construction of the labeled JG

Algorithm 2 receives the list of cliques L with their weight, and outputs a list L' of the cliques in the JG . The first clique in L' is the root, and the labels of cliques in the JG correspond to their position in the list. Each clique in the JG is a subset of a clique in L .

We focus now on step 5 of algorithm 2. The condition of maximizing the number of variables in $C \cap (L'(1) \cup L'(2) \cdots \cup L'(N\text{Cliques}))$ states that the clique

Algorithm 2: Algorithm for learning a JG

- 1 Order the cliques in L decrementally according to their weight.
 - 2 Add element $L(1)$ to list L'
 - 3 Remove element $L(1)$ from L
 - 4 While L is not empty
 - 5 Find the first element C in L such that the number of variables in $C \cap (L'(1) \cup L'(2) \cdots \cup L'(N\text{Cliques})) \neq C$, and $C \cap L'(1) \cap L'(2) \cdots \cap L'(N\text{Cliques})$ is maximized
 - 6 If $C = \emptyset$
 - 7 Remove all the elements in L
 - 8 else
 - 9 Insert C in L'
-

C in L that has the highest number of overlapping variables with all the variables already in L' , will be added to L' . The number of overlapping variables has to be less than the size of the clique, constraint meaning that at least one of the variables in C has not appeared before. If there exist many cliques with maximum number of overlapped variables, the one that appears first in L is added to L' . On the other hand, if the maximum number of overlapped variables is zero, then in the JG there exists more than one connected component. In this case we have a set of junction graphs, however we have preferred to abuse the notation and call it JG , whether it has one or more connected components. Finally, the addition of cliques stops when all the variables are already in the JG .

3.4 Calculation of the marginal probabilities

Marginal probabilities are found by calculating the number of counts associated to each configuration, and normalizing. In the implementation, the learned model's parameters can be changed by adding a perturbation in the form of probabilistic priors [10]. The effect of this type of priors is similar to the effect of mutation in GAs. In many problems they improve the behavior of the algorithm. This fact can be explained by the important role played by mutation in avoiding premature convergence, and allowing the exploration of new regions during the search.

4 Sampling of the approximate factorization

To create the new generation of solutions FDAs sample points from the probabilistic model. The MN-FDA sampling algorithm follows the order determined by the

labels of the JG . The variables corresponding to the first clique in the JG are instantiated sampling from the marginal probabilities. For the rest of cliques, each subset of variables that has not been instantiated is sampled conditionally on the variables already instantiated that belong to the clique. The process is very similar to Probabilistic Logic Sampling (PLS) [7] when it is used in junction trees. There exists however an important difference. The definition of JT discards the existence of cycles. A labeled JG can contain cycles. This fact allows the representation of more interactions, but it does not essentially change the performance of the sampling algorithm. The reason is that in every step of the JG sampling algorithm, the conditioning and conditioned subsets of variables will belong to the clique whose variables are being sampled.

5 MN-FDA

FDAs based on undirected graphical models represent the factorizations using a JT . The FDA^* [15] uses a fixed model of interactions, only the parameters of the cliques are learned in each generation. The simplest FDA is the Univariate Marginal Distribution Algorithm (UMDA) [11]. The UMDA assumes that all the variables are independent, and in every step it makes a parametric learning of the univariate probabilities.

The FDA-learning [17] is based on a theoretical algorithm [4] that learns a chordal independence graph straight from the data, allowing to change the model structure in each generation. If the underlying probability model is decomposable, the algorithm recovers it, if not it recovers a chordal supergraph. The JT is constructed from the chordal graph found.

Our algorithm will be called Markov Network FDA (MN-FDA), its pseudo-code is presented in algorithm 3. The main difference between it and previous FDAs based on undirected models is that it uses as its probabilistic model a JG , allowing factorizations that have not to be correct.

Analogously to GAs, different replacing strategies can be incorporated to the MN-FDA. Additionally, in the case of proportional selection the learning step can be done straight on the probabilities determined by the selection, without the need of constructing a selected set [21].

6 Experiments

In our experiments we compare the behavior of the MN-FDA with other FDAs. First, a number of functions commonly used to evaluate evolutionary algorithms are presented. Also a practical problem used

Algorithm 3: MN-FDA

```

1 Set  $t \leftarrow 0$ . Generate  $N \gg 0$  points randomly.
2 do {
3   Select a set  $S$  of  $k \leq N$  points according to
   a selection method.
4   Learn a  $JG$  from the data
5   Calculate the marginal probabilities for all
   the cliques in the  $JG$ .
6   Generate a the new population sampling
   from the  $JG$ .
7    $t \leftarrow t + 1$ 
8 } until Termination criteria are met

```

in the experiments is described. All the problems used in the experiments are defined on binary variables. The numerical results and the analysis of the experiments are presented afterwards.

6.1 Functions used in the experiments

Deceptive functions were introduced by Goldberg to show the deceptive nature of the GAs behavior, and to address the problems given by the convergence to local optima of the function. The following 4 elementary deceptive functions of k variables are used to define some of the additive functions used in our experiments. They are defined in terms of the univariate value $u(x) = \sum_{i=1}^n x_i$.

u	0	1	2	3	4	5
f_{dec}^3	0.9	0.8	0	1		
f_{dec}^4	3	2	1	0	4	
$IsoT_1$	m	0	0	0	0	0
$IsoT_2$	m	0	0	0	0	$m - 1$

$$Onemax(x) = \sum_{i=1}^n x_i \quad (1)$$

$$f_{3deceptive}(x) = \sum_{i=1}^{i=\frac{n}{3}} f_{dec}^3(x_{3i-2}, x_{3i-1}, x_{3i}) \quad (2)$$

$$Deceptive_4(x) = \sum_{i=1}^{i=\frac{n}{4}} f_{dec}^4(x_{4i-3}, x_{4i-2}, x_{4i-1}, x_{4i}) \quad (3)$$

$$F_{IsoP}(n, m, k, x) = \left(\sum_{i=1}^n x_i \right) + k \cdot (m + 1) \cdot ((1 - x_1) \cdot \dots \cdot (1 - x_m) x_{m+1} \cdot \dots \cdot x_n) \quad (4)$$

<i>OneMax</i>				<i>BigJump</i> (30, 3, 1)				<i>Deceptive4</i>			
<i>n</i>	<i>Alg.</i>	<i>N</i>	<i>succ.</i>	<i>n</i>	<i>Alg.</i>	<i>N</i>	<i>succ.</i>	<i>n</i>	<i>Alg.</i>	<i>N</i>	<i>succ.</i>
30	UMDA	30	75	30	UMDA	200	100	32	UMDA	800	0
30	LFDA _{0.25}	100	2	30	LFDA _{0.25}	200	58	32	FDA	100	81
30	LFDA _{0.5}	100	38	30	LFDA _{0.5}	200	96	32	LFDA _{0.25}	800	92
30	LFDA _{0.75}	100	80	30	LFDA _{0.75}	200	100	32	LFDA _{0.5}	800	72
30	LFDA _{0.25}	200	71	30	LFDA _{0.25}	400	100	32	LFDA _{0.75}	800	12
30	MN-FDA	30	72	30	MN-FDA	100	92	32	MN-FDA	600	90
30	MN-FDA	100	98	30	MN-FDA	200	100	32	MN-FDA	800	100

Table 1: Comparison between the MN-FDA with the UMDA and the LFDA for different functions

When analyzing interactions between variables it is important to consider interactions that do not depend on the linear codification of solutions. To this end we considered function $F_{IsoTorus}$ (5) where x_{up} , x_{left} , etc., are defined as the appropriate neighbors, wrapping around.

$$F_{IsoTorus}(x) = IsoT_1(x_{1-m+n}, x_{1-m+n}, x_1, x_2, x_{1+m}) + \sum_{i=2}^n IsoT_2(x_{up}, x_{left}, x_i, x_{right}, x_{double}) \quad (5)$$

Function $BigJump$ (6) was introduced in [12]. A valley has to be crossed in order to reach the global optimum of this function. The bigger the parameter m is for this function, the wider the valley. k can be increased to give bigger weight to the maximum.

$$BigJump(u, n, m, k) = \begin{cases} u & \text{for } 0 \leq u \leq n - m \\ 0 & \text{for } n - m \leq u \leq n \\ k \cdot n & \text{for } u = n \end{cases} \quad (6)$$

The generalized Ising model (7) is described by the energy functional (Hamiltonian) where L is the set of sites called a lattice. Each spin variable σ_i at site $i \in L$ either takes the value 1 or value -1 . A specific choice of values for the spin variables is called a configuration. The constants J_{ij} are the interaction coefficients. In our experiments we take $h_i = 0$, $\forall i \in L$. The ground state is the configuration with minimum energy.

$$H = - \sum_{i < j \in L} J_{ij} \sigma_i \sigma_j - \sum_{i \in L} h_i \sigma_i \quad (7)$$

6.2 Numerical results

In the following experiments N is the population size, $succ$ is the number of times the optimum was reached in 100 experiments, gen the average number of generations to convergence, \hat{f} the average fitness of solutions,

and $eval$ is the average number of evaluations needed to find the optimum.

In table 1 results of the MN-FDA for different functions are compared with results published in [12] for the UMDA and the LFDA (A FDA that uses a Bayesian probabilistic model). For the functions considered, the MN-FDA achieved equal or better results than the LFDA. We have observed that the learning algorithm used by the MN-FDA easily detects variables that are independent. The BN learning algorithms used by Bayesian FDAs may have problems recognizing independency, particularly if the parameter that specifies the density of the network (parameter α in the case of the LFDA) is small.

In table 2 we have included the results for the UMDA, the Tree-FDA, and the LFDA for other functions. For function $f_{3deceptive}$ the results of the MN-FDA are the best, and there is a significant difference with the results achieved by the LFDA. For function $F_{IsoTorus}$ LFDA finds the optimum more times than the MN-FDA, however its average fitness is lower. For function F_{IsoP} the Tree-FDA takes advantage of the chain-shaped structure of function F_{IsoP} to achieve the best results. It can be appreciated that the UMDA is not able to solve the problems with interactions.

We have generated 4 random instances of the Ising model for different number of variables ($n \in \{25, 36, 49, 64\}$). For each of the instances we investigate two different issues. First, the influence of using the prior information about the interactions of the variables. MN-FDA^s is a Markov Network FDA that does not learn the independence graph from the data. In this case the lattice where the Ising model is defined serves as the independence graph. The maximum size of the cliques is equal 2. The parameters of the cliques that belong to the JG are learned from the data. The second issue we study is the scaling of the algorithm.

An analysis of the results shows the convenience of using prior information about the optimization problem for increasing the efficiency of the MN-FDA. The

	$f_{3deceptive}$			$F_{IsoTorus}$			F_{IsoP}		
	<i>succ.</i>	\hat{f}	<i>gen</i>	<i>succ.</i>	\hat{f}	<i>gen</i>	<i>succ.</i>	\hat{f}	<i>gen</i>
MN-FDA	77	11.97	8.0	78	210.78	5.8	69	1190.69	5.0
UMDA	0	0	0	17	175.01	8.7	0	0	0
Tree-FDA	31	11.90	9.3	70	210.70	6.3	99	1190.99	5.8
LFDA	45	11.91	6.4	85	210.55	6.0	76	1190.75	4.6

Table 2: Comparison between the MN-FDA with other FDAs for different functions.

small population size that is enough for the convergence of the MN-FDA^s is not sufficient for the MN-FDA. As expected, when the number of variables increases a higher population size is needed to solve the problem.

<i>Inst.</i>	<i>N</i>	MN-FDA ^s		MN-FDA	
		<i>succ.</i>	<i>eval.</i>	<i>succ.</i>	<i>eval.</i>
I^{25}	200	100	849	43	1163
I^{36}	400	86	2316	41	2453
I^{49}	700	82	3841	36	4201
I^{64}	700	67	6031	28	6641

Table 3: Results of the MN-FDA for different Ising instances.

7 Conclusions

In this paper we have presented a FDA that approximates the probability distribution determined by selection using a labeled JG . The JG is found by calculating the maximal cliques of a Markov Network that can be given as an input or learned from the data. Our work is related with previous work by Muehlenbein et al. [15], where approximate factorizations were recognized as an alternative for modeling probabilistic distributions. Our research has led to a different way of finding these approximations. It is also related with the work presented by Brown et al. [3] in the application of MRFs to GAs. They have used probabilistic models of GA fitness functions to generate new solutions. Our work shows a number of relevant differences with this approach:

1. The use of statistical test to learn the structure of interactions. In [3] the structure of the interactions is known a priori.
2. The construction of the JG from the MN
3. The use of PLS on the JG . In [3] the Metropolis algorithm is used to generate new solutions.

The results of the experiments show that the MN-FDA is able to optimize theoretical functions as well as functions derived from practical problems, overperforming other evolutionary algorithms. The MN-FDA generalizes other FDAs by learning factorizations that have not to be correct. More theoretical investigation is needed to determine bounds for the convergence of the MN-FDA. Other practical optimization problems must be tried to assess the performance of the algorithm.

Bibliography

- [1] P. A. N. Bosman and D. Thierens. Linkage information processing in distribution estimation algorithms. In W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, and R. E. Smith, editors, *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99*, volume I, pages 60–67, Orlando, FL, 1999. Morgan Kaufmann Publishers, San Francisco, CA.
- [2] C. Bron and J. Kerbosch. Algorithm 457—finding all cliques of an undirected graph. *Communications of the ACM*, 16(6):575–577, 1973.
- [3] D. F. Brown, A. Garmendia-Doal, and J. A. W. McCall. Markov random field modelling of royal road genetic algorithms. In P. Collet, editor, *Proceedings of EA 2001*, volume 2310 of *Lecture Notes in Computer Science*, pages 65–76. Springer Verlag, 2002.
- [4] L. M. deCampos and J.F.Huete. Algorithms for learning decomposable models and chordal graphs. Technical Report DECSAI 970213, University of Navarra, 1997.
- [5] R. Etxeberria and P. Larrañaga. Global optimization using Bayesian networks. In *Proceedings of the Second Symposium on Artificial Intelligence (CIMAF-99)*, pages 151–173, Habana, Cuba, March 1999.

- [6] D. E. Goldberg. *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley, Reading, MA, 1989.
- [7] M. Henrion. Propagating uncertainty in Bayesian networks by probabilistic logic sampling. *Uncertainty in Artificial Intelligence*, 2:317–324, 1988.
- [8] J. H. Holland. *Adaptation in natural and artificial systems*. University of Michigan Press, Ann Arbor, MI, 1975.
- [9] P. Larrañaga and J. A. Lozano. *Estimation Distribution Algorithms. A new tool for Evolutionary Optimization*. Kluwer Academic Publishers, Boston/Dordrecht/London, 2001.
- [10] T. Mahnig and H. Mühlenbein. Optimal mutation rate using Bayesian priors for Estimation of Distribution Algorithms. In K. Steinhöfel, editor, *Proceedings of the First Symposium on Stochastic Algorithms: Foundations and Applications, SAGA-2001*, volume 2264 of *Lecture Notes in Computer Science*, pages 33–48. Springer, 2001.
- [11] H. Mühlenbein. The equation for response to selection and its use for prediction. *Evolutionary Computation*, 5(3):303–346, 1997.
- [12] H. Mühlenbein and T. Mahnig. *Theoretical Aspects of Evolutionary Computing*, chapter Evolutionary Algorithms: From Recombination to Search Distributions, pages 137–176. Springer Verlag, Berlin, 2000.
- [13] H. Mühlenbein and T. Mahnig. Evolutionary synthesis of Bayesian networks for optimization. *Advances in Evolutionary Synthesis of Neural Systems, MIT Press*, pages 429–455, 2001.
- [14] H. Mühlenbein and T. Mahnig. Evolutionary optimization and the estimation of search distributions with applications to graph bipartitioning. *International Journal on Approximate Reasoning*, 2002. to appear.
- [15] H. Mühlenbein, T. Mahnig, and A. Ochoa. Schemata, distributions and graphical models in evolutionary optimization. *Journal of Heuristics*, 5(2):213–247, 1999.
- [16] H. Mühlenbein and G. Paaß. From recombination of genes to the estimation of distributions I. Binary parameters. In A. Eiben, T. Bäck, M. Shoenauer, and H. Schwefel, editors, *Parallel Problem Solving from Nature - PPSN IV*, pages 178–187, Berlin, 1996. Springer Verlag.
- [17] A. Ochoa, M. R. Soto, R. Santana, J. C. Madera, and N. Jorge. The Factorized Distribution Algorithm and the junction tree: A learning perspective. In A. Ochoa, M. R. Soto, and R. Santana, editors, *Proceedings of the Second Symposium on Artificial Intelligence (CIMAF-99)*, pages 368–377, Habana, Cuba, March 1999.
- [18] M. Pelikan and D. E. Goldberg. Hierarchical BOA solves Ising spin glasses and Max-Sat. IlliGAL Report No. 2003001, University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory, Urbana, IL, January 2003.
- [19] M. Pelikan, D. E. Goldberg, and E. Cantú-Paz. BOA: The Bayesian Optimization Algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99*, volume I, pages 525–532, Orlando, FL, 1999. Morgan Kaufmann Publishers, San Francisco, CA.
- [20] M. Pelikan, D. E. Goldberg, and F. Lobo. A survey of optimization by building and using probabilistic models. *Computational Optimization and Applications*, 21(1):5–20, 2002.
- [21] R. Santana. Factorized distribution algorithms: Selection without selected population. 2003. Submitted for publication.
- [22] P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction and search*. Lecture Notes in Statistics. Springer-Verlag, New York, 1993.

CONFLICT RESOLUTION BY RANDOM ESTIMATED COSTS

ROMAN V BELAVKIN

School of Computing Science, Middlesex University, London NW4 4BT, UK

Abstract: Conflict resolution is an important part of many intelligent systems such as production systems, planning tools and cognitive architectures. For example, the ACT-R cognitive architecture [Anderson and Lebiere, 1998] uses a powerful conflict resolution theory that allowed for modelling many characteristics of human decision making. The results of more recent works, however, pointed to the need of revisiting the conflict resolution theory of ACT-R to incorporate more dynamics. In the proposed theory the conflict is resolved using the estimates of the expected costs of production rules. The method has been implemented as a stand alone search program as well as an add-on to the ACT-R architecture replacing the standard mechanism. The method expresses more dynamic and adaptive behaviour. The performance of the algorithm shows that it can be successfully used as a search and optimisation technique.

keywords: conflict resolution, decision making, search, optimisation, rule-based systems, cognitive modelling.

1 INTRODUCTION

Many problems do not have a unique solution. Moreover, some problems may have infinitely large number of similar solution paths. A conflict occurs when several alternative decisions are available corresponding to different solution paths. Intelligent systems, such as rule-based systems, planning tools, cognitive architectures, rely on different strategies to resolve a conflict. The simplest method is a random or ordered choice of rules. Other strategies use recency or specificity of rules. More sophisticated methods can use statistical information about previous successes and failures of applying the rules to infer the probability of a success. In addition, some methods take into account costs of the rules, which represent the efforts (e.g time) required from the problem solver to perform the actions.

Statistical (Bayesian) methods proved to be very successful not only for a conflict resolution, but also for modelling some aspects of human behaviour. For example, the ACT-R cognitive architecture [Anderson and Lebiere, 1998] uses subsymbolic statistical information to choose a single production rule from a set of several rules matching the current goal (conflict set). ACT-R models have been successful in predicting many properties of human decision making, problem solving and learning. Despite the success, however, some recent works have pointed out several problems and limitations of the conflict resolution theory in ACT-R. These problems will be summarised in the first section of this paper.

The new method introduced in this paper relies on the same statistical information as in ACT-R, but uses it in a different way. The new method is more adaptable to a changing environment. Its dynamics is a consequence of the entropy reduction during problem solving. In addition, the method revises and unites

several parameters in ACT-R. Many ideas were inspired by the progress in the theories of neural plasticity [Sejnowski, 1977a, 1977b; Bienenstock, Cooper, and Munro, 1982], as well as the information theoretic approach to cognitive models of decision making, learning and emotion [Belavkin and Ritter, 2003].

In this paper the theory and the algorithm will be presented in a general form, so that they could be applied to different domains. In the end of the paper a program demonstrating the method will be described, and its performance will be discussed. It will be suggested that the method can be used as a powerful optimisation and search technique.

2 CONFLICT RESOLUTION IN ACT-R

The ACT-R [Anderson and Lebiere, 1998] cognitive architecture uses a *utility* values U_i attached to every production rule i , and in the case of a conflict the rule with the highest U_i value is selected ($i = \arg \max U_i$). The utility of rule i is defined as

$$U_i = P_i G - C_i + \xi(\sigma^2). \quad (1)$$

Here P_i is the *expected probability* that the goal will be achieved after rule i has fired, and C_i is the average *cost* of the rule (average time required to achieve the goal). These rules-specific values are learned in ACT-R statistically using the records of past successes and failures as well as the efforts spent on each rule. Another two members of equation (1) are G — the *goal value* parameter (usually measured in time units), and $\xi(\sigma^2)$ — the *expected gain noise*, a random number derived from a normal distribution with zero mean and variance σ^2 .

This conflict resolution scheme (1) allowed ACT-R to model many characteristics of human problem solving, such as probability matching and the effect of the pay-

off (reinforcement) on choice behaviour. Indeed, one can see from (1) that utility is a function of probability of success and the goal value.

Noise ξ in conflict resolution proved to be a good candidate for modelling different levels of expertise. It was shown in [Jones, Ritter, and Wood, 2000] that by increasing the noise variance σ^2 a model of an adult problem solver may begin to behave more like a young problem solver. Thus, learning may result in reduction of noise with time. It was suggested during the ACT-R workshop in August 2001 that such dynamics could potentially improve the fit of some models to data.

Moreover, it has been proposed that noise variance should follow the uncertainty of a success [Belavkin and Ritter, 2003], and its changes may play an important role for optimising the choice behaviour. In addition, it was shown that goal value G , if controlled dynamically, may optimise the expenditure of efforts [Belavkin, 2002]. Indeed, noise and goal value control breadth and depth of the search for a solution respectively.

Modern cognitive architectures have mechanisms not only for learning the statistics of existing rules, but also they may learn new rules (e.g. chunking in SOAR [Newell, 1990] or production compilation in ACT-R). It was proposed that noise ξ should be rule-specific and affect the new rules more than the ‘older’ rules in the system.

Unfortunately, this is not possible in the current conflict resolution mechanism because the goal value G and noise variance σ^2 are global parameters affecting all the rules simultaneously. Moreover, they are constants, and the theory does not explain their dynamics.

As we can see from the discussion above that being a successful theory of conflict resolution for some aspects of human problem solving, the current conflict resolution mechanism in ACT-R may not yet be complete. The new method introduced here is an attempt to overcome the described problems.

3 COST AND SUCCESS PROBABILITY

One may question the need of having the success probability P , as well as the goal value G and cost C in the utility equation (1). Let us consider the cost C of achieving the goal as a random variable, and let $P(C)$ be the probability that the goal will be achieved at the cost C (probability that the cost is exactly C). The expected value of the cost is

$$E\{C\} = \sum_C C P(C) \quad \left(E\{t\} = \int_0^\infty t P(t) dt \right),$$

where the summation is made across all possible values of C (or an integral on $t \in [0, \infty]$ if C represents continuous time). The distribution function $P(C)$ gives the value of success probability for any cost. That is

the expected probability P for given C or G in (1).

Knowledge of distribution functions $P(C)$ for different decisions would allow the problem solver to calculate their expected costs $E\{C\}$, and to choose the best rule (or strategy) to solve the problem. Of course the difficulty here is that when solving a problem for the first time nothing is known about $P(C)$. The only way to sample these distributions is by trying to solve the problem using different strategies. Moreover, even if we were determined to find out what the costs are by trial, we would soon realise that some costs are very hard to ‘measure’ directly.

For example, random rotation of the edges of a Rubik’s cube may eventually assemble the puzzle, but the chance, as we say, is very low. More correctly would be to say that the probability of assembling the puzzle quickly is very low. This means that most likely the cost of such random rotation strategy is very high, and one would have to spend a lot of time waiting for the result. The question is when to give up and try another strategy?

The ability to give up on hopeless solutions without exploring them in full is a very important property of human problem solving. One of the important property of the algorithm that will be introduced is that it specifies exactly how deeply an alternative should be explored.

4 PROBLEM SOLVING AS AN OBSERVATION OF A POISSON PROCESS

Let us imagine that a computer solves some problem using a particular algorithm, and each time after the goal state has been achieved, the computer is restarted and is given to solve the same problem again (1).



Figure 1: A computer running an algorithm in a loop. The goal state is observed at a rate $\lambda = \frac{1}{E\{C\}}$, where $E\{C\}$ is the expected cost.

Now, if the expected cost of the solution that the computer is using is $E\{C\}$, then we shall observe the goal state every $E\{C\}$ seconds, or at a rate $\lambda = \frac{1}{E\{C\}}$. We may consider the occurrence of the goal state as a Poisson process. The probability of observing n events by the time t is

$$P(n | \lambda) = \frac{(\lambda t)^n}{n!} e^{-\lambda t}, \quad n = 0, 1, 2, \dots \quad (2)$$

Here λ is called the mean count rate ($\lambda = 1/E\{C\}$), and $n = 0, 1, 2, \dots$ is the number of observations of the event by the time moment t .

Note, that for $\lambda = 0$ (or $E\{C\} = \infty$) the probability (2) becomes zero for any t , which corresponds to a case when the event is impossible. Thus, for an event to be possible the rate must be $\lambda > 0$ (or $E\{C\} < \infty$). Perhaps, when solving a problem, one must assume that the goal state is possible (be optimistic). That is $\exists G < \infty : E\{C\} \leq G$.

Now, let us consider some special cases of the Poisson probability (2).

Failure probability is the probability that the event will not occur ($n = 0$), and according to (2) it is

$$q(t) = P(n = 0) = e^{-\lambda t} . \quad (3)$$

The shape of the above function is shown by a declining dashed curve on Figure 2.

Success probability is the probability that the event will occur at least once ($n > 0$):

$$p(t) = P(n > 0) = 1 - q(t) = 1 - e^{-\lambda t} . \quad (4)$$

The success probability is shown on Figure 2 by an inclining dashed curve. One can see that it increases with time if $\lambda > 0$.

When solving a problem, especially for the first time, what we are interested in is the first occurrence of the goal state. Moreover, often we do not need to solve exactly the same problem again. Therefore, the probability of the very first success is of special interest.

First success probability that the event will occur exactly once ($n = 1$) is

$$p_1(t) = P(n = 1) = \lambda t e^{-\lambda t} . \quad (5)$$

The shape of the above function is shown by a solid curve on Figure 2. One may notice that it increases with time up to a certain maximum and then decreases again.

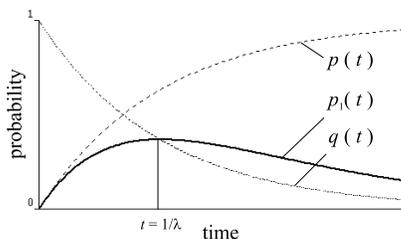


Figure 2: Probability of failure $q(t)$ decreases with time (declining dashed curve), probability of success $p(t)$ increases with time (inclining dashed curve). Probability of the first success $p_1(t)$ (solid curve) has a unique maximum in $t = 1/\lambda = E\{C\}$.

Let us find the time moment corresponding to the maximum of the first success probability:

$$\dot{p}_1(t) = \frac{d}{dt} \lambda t e^{-\lambda t} = 0 \quad \Rightarrow \quad t = \frac{1}{\lambda} .$$

We can see that this time moment corresponds to the expected cost $E\{C\}$ (the most likelihood cost). Note, that at this point the probability of the first success equals the probability of failure:

$$p_1(t) = q(t) = e^{-1} , \quad t = 1/\lambda .$$

It can be shown that for a system with two outcomes (first success and its complement) this is the moment of the maximum entropy, and hence it is the best moment to make a new estimate of λ using new information. If the new estimate turns out to be much greater than expected, then it may also be the optimal moment to change strategy or give up.

5 ESTIMATION OF THE EXPECTED COST

Up to this point we have been talking about problem solving as an observation of a Poisson process with known rate λ . Indeed, equation (2) describes the conditional probability $P(n | \lambda)$ of observing n events for a given value of λ (t is parameter). In reality, when solving a problem, the rate λ is unknown, and the expected cost is what we are trying to estimate. What is known is the number of successes n and the amount of time (or cost) that have been spent.

Let us estimate the rate λ (and, hence, $E\{C\}$) from the observed number of successes n after spending t amount of time. This can be done using posterior probability $P(\lambda | n)$, which can be obtained by the Bayes' formula

$$P(\lambda | n) = \frac{P(n, \lambda)}{P(n)} = \frac{P(n | \lambda)P(\lambda)}{P(n)} .$$

One can show that when *a priori* all the values of λ are equally probable, and the likelihood probability is described by the Poisson distribution (2), then the posterior probability can be found from the likelihood:

$$P(\lambda | n) = t P(n | \lambda) .$$

Now, using the above formula for the posterior probability, we can find the posterior mean estimate of λ :

$$E\{\lambda\} = \int_0^{\infty} \lambda P(\lambda | n) d\lambda = \frac{n+1}{t} ,$$

and hence $E\{C\} \approx \frac{t}{n+1}$. Note that this estimation is biased towards successes (optimistic). Such an estimate is more useful than $\frac{t}{n}$ because it can be used even when no successes have yet been observed ($n = 0$). For this reason the algorithm was named OPTIMIST.

Interestingly, the OPTIMIST algorithm finds support in several studies on kinetics of choice and animal learning. Myerson and Miezin [1980] found that the response frequency in rats can be explained by the Poisson distribution [see also Mark and Gallistel, 1994].

6 RECURSIVE ESTIMATION

Let us return to the example of a computer solving a problem (Figure 1), but with one difference: the expected cost $E\{C\}$ is unknown. Also, this time we control when the computer is restarted. Our goal is to restart the computer in such a way, that the goal state appeared at the highest rate possible.

Let us denote by Δt the time interval after which we restart the computer. If we restart the computer too late $\Delta t > E\{C\}$, then obviously the rate at which the goal state occurs will not be the highest. On the other hand, if we restart the computer too early $\Delta t < E\{C\}$, then often the computer will not have enough time to finish solving the problem.

Let us conduct a series of trials registering the first occurrence of the goal state during time intervals Δt : if the goal state is registered, then we shall restart the computer immediately after it; otherwise, restart the computer after Δt . One may notice that there are only two possible outcomes of such trials (binomial trials):

Failure: the goal state has not been achieved, the number of successes n does not change, the overall effort (time) spent increases by $C = \Delta t$.

Success: the goal state is achieved, the number of successes n increases by one, and the effort increases by $C \leq \Delta t$.

By counting n number of successes and summing up the time spent $t = C_0 + \dots + C_k$ in k trials, we can estimate the expected cost $E\{C\}$ using posterior mean formula:

$$E\{C\} \approx \bar{C} = \frac{t}{n+1}. \quad (6)$$

Now, starting with some small $\Delta t = C_{\min}$ let us set each next Δt equal to the last estimation of $E\{C\}$:

$$\Delta t_{k+1} = \bar{C}_k = \frac{\sum_{i=0}^k C_i}{n+1}.$$

An example of step by step calculation of \bar{C}_i and Δt_{k+1} for ten trials ($k = 0, \dots, 10$) is shown in Table 1. The dynamics of the estimated cost (6) during 20 trials is shown on Figure 3. With no successes registered ($n = 0$) the estimated value grows exponentially until it becomes greater than the expected cost $E\{C\}$, which means that the system spends enough efforts (time) to register first successes ($n = 1, 2, \dots > 0$). As the number of successes n increases, the estimated value \bar{C} decreases converging to the expected cost $E\{C\}$.

One can see that if the expected cost of the algorithm our computer is using is finite $E\{C\} < \infty$, then with trials the estimated cost \bar{C} , and hence the restarting time Δt , will converge to $E\{C\}$:

$$\lim_{k \rightarrow \infty} \Delta t_{k+1} = \lim_{k \rightarrow \infty} \bar{C}_k = E\{C\}.$$

Also, as a result, the goal state will occur at the highest rate possible.

Table 1: Example of estimation of $E\{C\}$ in 10 trials with failures in the first two and successes in the following eight trials.

k	n	\bar{C}_k	Δt_{k+1}
0	0	C_{\min}	\bar{C}_0
1	0	Δt_1	\bar{C}_1
2	0	$\Delta t_1 + \Delta t_2$	\bar{C}_2
3	1	$\frac{\Delta t_1 + \Delta t_2 + \Delta t_3}{2}$	\bar{C}_3
...
10	8	$\frac{\Delta t_1 + \dots + \Delta t_{10}}{9}$	\bar{C}_{10}

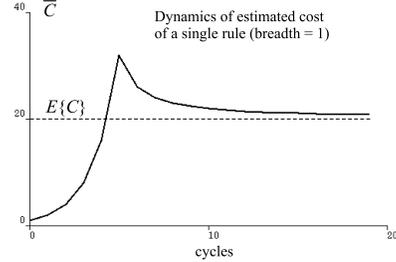


Figure 3: Estimated cost \bar{C} converges to the expected cost $E\{C\}$ with cycles $k \rightarrow \infty$ and the number of successes $n > 0$.

7 CONFLICT RESOLUTION

In previous sections we discussed how to estimate the cost of one particular strategy (algorithm, decision or production rule) by estimating the rate of a hypothetical Poisson process. As an illustration we used the example of a computer set into an endless loop solving a problem (Figure 1). In a similar manner let us represent a conflict by a choice of several such computers tackling the same problem, but with only one computer used at a time.

Let us denote the options (choice of computers, strategies or rules) by x , and suppose that each computer uses different algorithm with a different expected cost $E\{C(x)\}$. Our goal is to find the fastest (cheapest) x .

Let us record for each option x the following information: $k(x)$ — the number of times x was used, $n(x)$ — the number of successes for x , $t(x) = C_0(x) + \dots + C_{k(x)}(x)$ — the efforts (all time) spent using x . After trying each option we can estimate its expected cost $E\{C(x)\} \approx \bar{C}(x)$ using equation (6). In order to resolve the conflict we introduce a *random estimated cost*:

$$\tilde{C}(x) = \frac{k(x)\bar{C}(x) + \xi(\bar{C}(x))}{k(x) + 1}. \quad (7)$$

Here $\xi(\bar{C}(x))$ is called a *random prediction*, and it is a random variable defined in such a way that its expected value equals the estimated cost $\bar{C}(x)$ ($E\{\xi\} = \bar{C}(x)$). For example, we can use the following function: $\xi(\bar{C}(x)) = \text{rand} \in (0, 2\bar{C}(x))$.

The conflict is resolved by selecting an alternative x with the smallest random estimated cost:

$$x = \arg \min [\tilde{C}(x)].$$

8 PROPERTIES OF THE RANDOM ESTIMATE

One can see from (7) that the random estimated cost is a mixture of two components: the estimated cost \bar{C} and the random predication ξ made based on the last estimated cost \bar{C} . The contribution of the latter component for individual rule x depends on the number of trials k (experience), and it decreases. This means that if new production rules are learned during problem solving, their expected cost will be more affected by the random predication ξ .

The expected value of the random estimated cost \tilde{C} equals the expected cost $E\{C\}$, which follows from its definition (7). Moreover, with trials k the value of the random estimated cost \tilde{C} not only converges to the the expected cost $E\{C\}$, but also its value becomes more stabilised (less plastic) as the number of samples k increases. This property recalls of an important plasticity effect known for neural networks and explained by the *covariance* learning rule [Sejnowski, 1977a, 1977b; Bienenstock et al., 1982].

Because with successes the estimated cost \bar{C} decreases, so does the expected value of the random ξ , as by definition $E\{\xi\} = \bar{C}$. This way, with successes, the random estimated cost \tilde{C} decreases on average and becomes less random. We may compare this process with cooling the system down in simulated annealing [Kirkpatrick, Gelatt, and Vecchi, 1983].

On the contrary, if the number of failures increases, then the estimated cost \bar{C} grows exponentially. As a result, the contribution of the random predication ξ in (7) becomes more and more noticeable. This way, with failures, the random estimated cost \tilde{C} increases on average and becomes more random. We may compare this process with heating the system up.

The information acquired during the binomial tests of rules acts as reward or penalty signals in reinforcement learning theories [Barto, 1985; Barto and Anandan, 1985]. Indeed, initially all the alternatives x have equal chances to be selected from the conflict set, because no posterior information is available, and the choice is random. On successes the estimated cost \bar{C} , and consequently the expected value of \tilde{C} , decreases. As a result, the chance of the successful rule to be selected on the next trial increases. This is similar to excitation effect in neural networks. On the contrary, if a failure occurs, then the estimated cost \bar{C} , and consequently the expected value of \tilde{C} , increases. This leads to inhibition of the failed alternative x , because its chance to be selected next time decreases.

9 METHOD PERFORMANCE

There are currently two applications in which the OPTIMIST algorithm has been tested: an add-on to the ACT-R architecture replacing the conflict resolution mechanism [see Belavkin, 2002], and a demonstration

search program (all implemented in Common Lisp). The interface of the latter is shown on Figure 4. The program presents a search space with several gadgets controlling its parameters. The alternatives x representing different choice of strategy are located along the horizontal axis (breadth of the search space). The cost (depth of the search) is represented by the vertical axis.

When a particular rule x is selected, it is depicted by a vertical beam going up from the corresponding position (second alternative is shown selected on Figure 4). The height of the beam represents the current maximal cost (Δt). The outside world is represented by a distribution of the real costs. They are represented by thick horizontal bars. The distribution shape is controlled by the user, and it is not known to the algorithm. Figure 4 shows parabolic distribution of the real costs with the smallest (optimal) cost positioned in the middle. If the cost payed by the algorithm is enough to achieve the goal (the beam is higher than the real cost bar), then a success is registered (goal achieved); otherwise, a failure occurs. The horizontal line represents the latest estimation of the expected cost. The estimated costs for each alternative are stored in the memory of the program and are represented by thin horizontal bars.

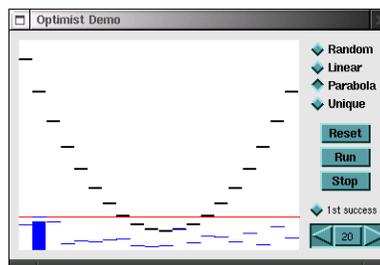


Figure 4: Interface of the OPTIMIST demonstration program

The program demonstrated the following behaviour of the algorithm with the four distinguishable stages. In the beginning the search is completely random and not very deep (the heights of the beams are small). When no successes are registered the estimated costs begin to grow exponentially (the beams begin to rise higher). The system ‘heats up’. When the depth explored by the algorithm is greater than the real cost, the first successes occur, and the estimated cost decreases. For some period of time the number of successes is comparable to the number of failures, and the system appears as ‘boiling’. When the algorithm finds more optimal solutions the system starts to cool down which is represented by a decrease of the estimated cost, and the choice is concentrated more on the successful entries with smaller (more optimal) costs. Finally, the system stabilises choosing only the optimal alternative and exploring just enough to reach the success. This stage can be compared with crystallisation.

If the distribution of the real costs changes after crys-

tallisation, the system heats up again until the algorithm finds another solution. This demonstrates the ability of the system to adapt to a changing environment. The speed of adaptation, however, decreases with the ‘age’ of the system.

Figure 5 shows from left to right the dynamics of choice proportion for parabolic distribution of the real costs with the optimal in the middle (breadth set to 50 alternatives). One can see that the breadth of the search decreases. The dynamics of the estimated cost is shown on Figure 6, and it illustrates the depth of the search as a function of trials (cycles).

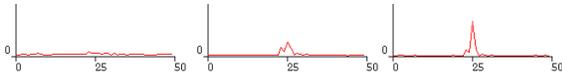


Figure 5: Dynamics of the choice proportion (from left to right).

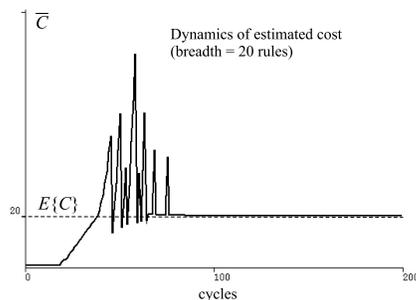


Figure 6: Dynamics of the estimated cost (optimal cost set to 20) for a conflict set of 20 alternatives (breadth 20).

The depth of search converges to the optimal. Moreover, the first solution found is not necessarily, but most likely to be the optimal. Indeed, the greater is the real cost of a solution, the less is its chance to be explored in full. On the contrary, the optimal solution path has the highest probability to be explored in full.

10 CONCLUSIONS

A new conflict resolution algorithm has been introduced, which united some of the parameters of the ACT-R cognitive architecture. The introduced learning and conflict resolution scheme addresses several problems of the ACT-R conflict resolution. It also implements other theories on kinetics of choice as a computational algorithm. In addition, the theory is general enough to be employed as a search and optimisation technique. The performance and cheap computational cost of the algorithm is encouraging for its application in various areas of computer science.

ACKNOWLEDGEMENTS

This work is sponsored by ESRC Credit and the ORS Award Scheme. Frank Ritter, David Elliman and David Wood provided useful comments and support for this work.

REFERENCES

- Anderson, J. R., and Lebiere, C. 1998. *The atomic components of thought*. Mahwah, NJ: LEA.
- Barto, A. G. 1985. “Learning by statistical cooperation of self-interested neuron-like computing elements.” *Human Neurology*, 4, 229–256.
- Barto, A. G., and Anandan, P. 1985. “Pattern-recognizing stochastic learning automata.” In *IEEE Transactions on Systems, Man and Cybernetics* (Vol. 15, pp. 360–375).
- Belavkin, R. V. 2002. *On emotion, learning and uncertainty: A cognitive modelling approach*. PhD Thesis, The University of Nottingham, United Kingdom.
- Belavkin, R. V., and Ritter, F. E. 2003. “The use of entropy for analysis and control of cognitive models.” In F. Detje, D. Dörner, and H. Schaub (Eds.), *Proceedings of the Fifth International Conference on Cognitive Modelling* (pp. 21–26). Bamberg, Germany: Universitäts-Verlag Bamberg.
- Bienenstock, E. L., Cooper, L. N., and Munro, P. W. 1982. “Theory for the development of neuron selectivity: Orientation specificity and binocular interaction in visual cortex.” *Journal of Neuroscience*, 2, 32–48.
- Jones, G., Ritter, F. E., and Wood, D. J. 2000. “Using a cognitive architecture to examine what develops.” *Psychological Science*, 11(2), 93–100.
- Kirkpatrick, S., Gelatt, C. D., and Vecchi, J. M. P. 1983. “Optimization by simulated annealing.” *Science*, 220(4598), 671–680.
- Mark, T. A., and Gallistel, C. R. 1994. “Kinetics of matching.” *Journal of Experimental Psychology*, 20(1), 79–95.
- Myerson, J., and Miezin, F. M. 1980. “The kinetics of choice: An operant systems analysis.” *Psychological Review*, 87(2), 160–174.
- Newell, A. 1990. *Unified theories of cognition*. Cambridge, Massachusetts: Harvard University Press.
- Sejnowski, T. J. 1977a. “Statistical constraints on synaptic plasticity.” *Journal of Mathematical Biology*, 69, 385–389.
- Sejnowski, T. J. 1977b. “Storing covariance with nonlinearly interacting neurons.” *Journal of Mathematical Biology*, 4, 303–321.

BIOGRAPHY

Roman Belavkin was born in Moscow in 1971. He graduated from Physics Department of the Moscow State University, but his interests switched from computer modelling of solar radiation to AI and cognitive modelling. He completed his PhD Thesis in 2002 in the University of Nottingham. He is currently based in Middlesex University in North London. His research is on cognitive modelling of the effects of emotion on decision making and learning. In his work he uses information theoretic approach for analysis of learning in cognitive models and architectures.



MODELLING TRAFFIC NAVIGATION NETWORK WITH A MULTI-AGENT PLATFORM

THIERRY HUET¹, TAHA OSMAN², CYRIL RAY¹

¹*Ecole Navale – IRENav*

BP 600 – Lanveoc-Poulmic, F-29240 Brest Naval, France

huet@ecole-navale.fr, ray@ecole-navale.fr

²*Nottingham Trent University – School of Computing and Mathematics*

Burton Street – Newton Building, NG1 4BU – Nottingham, England

taha.osman@ntu.ac.uk

Abstract: The increase of maritime traffic in the past few years have posed concerns about the safety of using this transport facility in the future. Recent accident statistics show that the age of boats, legislation and traffic flow control are nowadays insufficient to ensure the security of boats, crew, coasts and ecological system. Maritime authorities have set up Vessel Traffic Systems (VTS), which follows ship navigation specifically in high density areas. In this paper, we present a new computing approach to simulate and improve the security of maritime traffic. Based on agent technology and distributed systems this solution models the ship behaviours and adds to Vessel Traffic Systems the capability of comparing original and simulated ship tracks.

Keywords: Maritime Navigation Modelling, Traffic Analysis, Multi-Agent Systems, Distributed Object Computing.

1. INTRODUCTION

Sea traffic grows each years considerably. During 2001, intra-community goods transport increased by 27% [European Commission 2001]. An outcome of this growth is the congestion of different parts of the navigation's network such as the Cape Gris Nez, Ouessant in northern Atlantic and Malacca or Tsushima strait near the Asian coast. The result is a reduction of competitiveness in comparison to other means of transports and the increase of accident risks.

Disasters resulting from maritime accidents cause considerable damage and monopolize significant human and material resources [Paul 1997] in order to be solved. Relevant illustrations of the problem are the sinking of the ERIKA [BEA Mer 2000] in western coast of France which provoked a large oil leakage, and the sinking of the IEVOLI SUN [BEA Mer 2001b] which provoked chemical diseases near Chausey Islands. These accidents are mainly due to the excessive age of the ships, their bad maintenance, the crew lack of knowledge, and to non-compliance with the sea rules of transport. After the shipwreck of the ERIKA, recommendations were issued to enforce the modernization of naval equipment, improve crew training, and, more interestingly for our work, the establishment of centres for surveying the maritime traffic [BEA Mer 2001a]. In order to reduce the number of accidents, governments have two possibilities:

- The issue of regulations binding the ship's managers to increase security on board by training their staff and carrying-out regular maintenance of their ships. However, the economic market implies that if they want to survive, they have to propose the best price for freight transport. Therefore, generally, they prefer to reduce these costs.
- The survey of maritime traffic. With specific tools, national authorities can survey their territorial waters and apply national or international legislation, resulting in a cut in the amount of registered diseases.

Nowadays, many countries such as China, United States of America, Great Britain, and France have chosen a more pragmatic solution. They decided to install a large number of surveying tools and reinforce maritime legislation.

In this paper, we present a new method for modelling the maritime transport in a high traffic area. In section 2, we describe the problem of safety navigation and naval transport simulation. Section 3 presents technical problems related to the setting up of a safety navigation simulator and how they can be addressed by using new computing approaches. In section 4 we present the chosen solution we will implement to validate traffic navigation modelling. The last section presents outlooks and concludes this paper.

2. RELATED WORK AND SPECIFICATION OF THE REQUIREMENTS

The majority of transportation systems are monitored by computer systems. Applications related to road network monitor vehicles, gives information about construction, accident, detours [GoodWin and Pfrang 2001] and traffic analysis [Hadouaj et al. 2000]. For Air traffic control, applications are focused on traffic survey to prevent collision [Krozel and Peters 2000]. For transportation network, information is given on scheduling [Zhu et al. 2000], coordination between railways and road network on intermodal platform [Bürckert et al. 1999].



Fig. 1. Center equipped with VTS at Flint, Sweden (Öresund)

Vessel Traffic Systems (VTS, see figure 1) are used to manage maritime traffic [Amiel 2002]. It is a computer based system which uses a similar methodology to that used in air traffic control. One or several Radars are connected to the VTS and give ships bearing. Areas that are covered by VTS are bounded to ports, estuaries, straits and nearest coast. However, because of Radar and VTS costs, coastal area is not fully covered. Each ship is initially localized by Radar then, after vocal identification, the "track" is updated.

In order to identify high density areas where an improvement of the maritime traffic security should be set up, we in a first study, establish a following of ship tracks around western Europe [Huet 2002b]. In figure 2, dots represent the true roads followed by ships and their aggregations illustrate high density ships (represented by the darkest color). One of the most dangerous area is the Channel. Indeed the huge amount of ships and the lack of complete surveys of this area (see figure 3) lead as seen recently to accident (e.g. Tricolore shipwreck). Traffic flow in this area can reach up to 500 ships a day. In Gris Nez/Dover area (eastern part of the Channel), where the width is the shortest, this high density traffic have to be supervised with care in order to prevent potential shipwrecks. In the following, we will focus on the Channel area that can be considered as the most relevant area for the improvement of traffic security using new computing solutions.

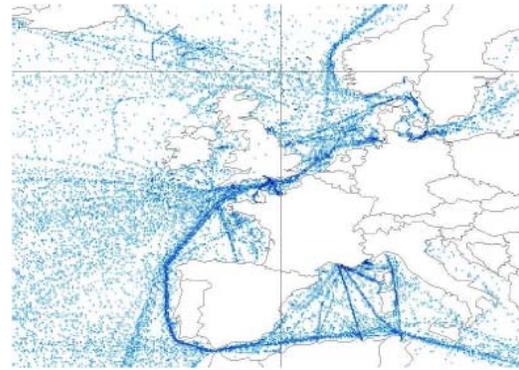


Fig. 2. Maritime traffic in Europe

In order to cover the Channel area, French and English government had set up VTS all along the coasts (see figure 3). In France, Maritime Rescue Co-ordination Center (MRCC) are fitted with VTS. On the French side, one can find a VTS associated to a CROSS (the French MRCC acronym). These are at Gris Nez, Cherbourg and Brest. In England, VTS are at Dover, Portland and Falmouth. Some VTS are connected to each other to exchange information. Hence, it is in theory possible to follow a ship from the beginning to the end of the Channel, but as illustrated in the figure 3 some areas are uncovered.

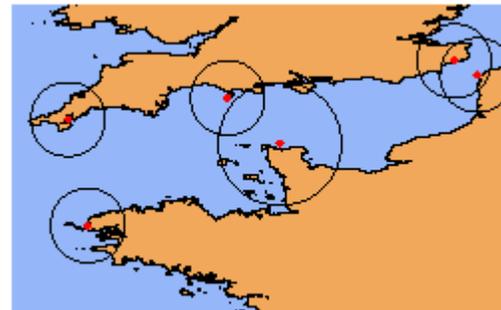


Fig. 3. Estimated area covered by VTS on the Channel

When a ship goes out of a VTS covered area, date, time, speed and direction parameters are given to the following VTS as a digital file. This file is also sent by fax to insure that the transmission has been correctly done. Then, the following VTS will propose an estimate arrival date and location. However, there is no information in-between covered areas, so if a hazardous situation arises, traditional methods of search and rescue are set up.

Adding a simulation interface in a VTS allows us to propose some additional services. In state of the art systems, VTS technology is fully automated. However, whereas following a ship on a screen is controlled, simulating its movement is yet a research topic. Therefore, For VTS systems it could be useful to: (1) estimate the vessel road when outside the radar cells; (2) compare the "true road" to the "estimated road" to anticipate potential problems; (3) replay a true

situation for training; (4) simulate a "day traffic" to estimate speed limitation, maximum density.

Optimum vessel track depends on different parameters such as weather conditions, human interaction or vessel geometry/attitude, and vessels interaction between themselves. In this context, we propose to model each "track" as a computer agent whose location is the major parameter.

3. ENABLING SAFETY NAVIGATION SIMULATION

Setting up such a safety navigation simulator requires taking into account lots of parameters. Indeed, to model the vessel movement in a distributed context, we have to take into account algorithms complexity. [Huet 2002a] demonstrates an agent technology that give the same algorithm complexity but increase interaction between user and model. We also have to take into account computer performances and operating systems [Ray and Huet 2000] and specifically the scalability problem. To avoid losing information, fault tolerance procedures should be used.

The agents computing paradigm fulfils most of the afore mentioned requirements. Agents are autonomous software entities that roam the Internet to perform tasks on behalf of the user. They access network resources more efficiently because they can move to the resources location rather than exchanging multiple network messages over congested bandwidth. Agent systems are also ideal for modelling interrelated objects behaviour. They use speech-act based communication primitives to cooperate [O'Brian and Nicol 1998], while maintaining their autonomy. The mobility and autonomy of mobile agents allow them to trail the track of ships, possibly migrating to other computing nodes that are closer to new ship tracking locations, while the advanced agent communication primitives should allow for the seamless integration of safety navigation rules into the cooperative behaviour of the agents representing the ships.

Despite their promise, there are a lot of issues such as trust, scalability, reliability that need to be resolved before agent systems can be adopted by distributed processing and Internet-based applications. In this work, we focus on problems that have direct implications on the safety navigation of marine traffic. The choice of an agent-based support for modelling of the various entities interacting in a maritime environment has some drawbacks. From an implementation point of view, the autonomy of each agent is ensured by the creation of an individual process or a thread.

Unfortunately computers have limited resources resulting in a limited number of manageable processes

(threads) and consequently agents. An advantageous configuration of a safety navigation simulator should imply the modelling of one ship using one agent because agent properties and behaviours can naturally replicate ship behaviour. This modelling approach is judicious but raises the problem of scalability of such a simulated system.

As mentioned earlier, computing systems have limited resources, thus a limited number of ship that can be modelled. In a realistic simulation of ship navigation in the Channel one can expect about 500 boats to manage by day. This figure is not so far from common limits of agent number that can be executed correctly by a computer (this limit depends on the computer itself but also on the operating system) and can lead to an overhead on agent computing engine. This overload is reinforced when the navigation space is larger or when the traffic increases. In order to provide an efficient simulator, this scalability problem needs to be addressed in order to ensure that any solution doesn't fall behind real-world response time.

Recent advances in computing offer new solution for the simulation of large-scale systems such traffic navigation system. The use of distributed system for efficient computing support for multi-agent platform allows to overcome the scalability problems. This distributed approach has been already used for different applications. In the domain of the understanding of urban phenomenon, a distributed agent approach is used for the simulation of traffic [Erol et al. 1999]. In [Ray and Claramunt 2002] a distributed platform models and simulates disaggregated data flows in an airport terminal.

The second major challenge is safeguarding the agent computing engine. Marine navigation systems are safety critical and faults affecting the driving application can have disastrous consequences. We mentioned earlier that agents are autonomous software objects that can migrate between Internet hosts and communicate with each other via sophisticated message exchange primitives to achieve a common goal, for example collision evasion. Hence, to ensure the reliability of the navigation system, we need to make the computing engine tolerant to faults that might affect the agent execution or the communication channel between the agents [Osman and Bargiela 2000a].

4. SOLVING SAFETY NAVIGATION

4.1 Ship modelling

The ship, identified by its location is the element, which is moving according to many dynamically changing environmental factors. Here we propose to model each ship by an agent. Since agents have built-

in support for multi-threaded behaviour, they can integrate sea/weather information (which is environmental information), vessel geometry and human interactions (figure 4).

We can find, in all navigation books, information to calculate the ship road while taking into account wind or stream drift. We will use this information to model the ship behaviour. This information depends on the type of navigation. Coastal navigation follows code of nautical procedures and practices, lights, buoys and fog signals, separation traffic scheme and sailing directions while ocean navigation follows shortest way rules (to go from one point to another one, the fastest is the shortest) and sailing directions.



Fig. 4. Modelling interactions between the ship and its environment

In [BEA Mer 2001b], events which caused the loss of the IEVOLI SUN are described. Major remarks in the report were the sea and weather condition, state of the ship, choice done by the captain to find the best road and ship condition:

- 1) Human interaction: the captain has to take into account several parameters to define its road. For instance, he has to take into account the distance between overlapping ships, chart information (such as traffic separation device).
- 2) Vessel Geometry/attitude: each vessel has its own performance parameters such as manoeuvrability, gyration, distance to stop or wind catching, stream catching, draught.
- 3) Environmental information integrates relationships between the ship behaviour and natural elements such as wind or tide effect.

We can identify currently several rules, which will be used in our modelling. These rules will be changed and expanded in the second step of our work. In a simplified case study, we will start by the modelling of ocean navigation. In that way, we can write:

$$\exists \lambda = \{\phi, G\}; \phi \in \{-180..180\}, G \in \{-90..90\} \quad (1)$$

λ is the ship location identified by its longitude (G) and its latitude (Φ)

$$\Lambda = \{\lambda_1 \dots \lambda_n\}; n \in \mathbb{N}^+ \quad (2)$$

Λ is the surveyed area.

$$\exists R = \{WP_1 \dots WP_n\} / WP_i \subset \Lambda; i \leq n \quad (3)$$

A road is defined by an ordered list of waypoints.

$$\exists \alpha \in \{-90 \dots 90\}, S \subset \Lambda; WP_i \in R \quad (4)$$

$$\Rightarrow \tan \alpha = \frac{G_i - G}{\phi_i - \phi} \quad (5)$$

$$\Rightarrow \delta = \sqrt{(G_i - G)^2 + (\phi_i - \phi)^2}; \delta \in \mathbb{R}^+ \quad (6)$$

A ship road is defined by an azimuth α and distance δ

$$\exists O / O \subset \Lambda \text{ and } \forall S \in \Lambda; S \notin O \quad (7)$$

An obstruction area is an area in the surveyed area where ships cannot advance.

$$\exists S_p \Rightarrow S_p \subset \Lambda = \{S, f(v)\} \quad (8)$$

Each ship has its own protection area. It is a circle whose the diameter varies with the speed v . A protection area is an obstruction. The protection area must not intersect with another obstruction. To ensure the correctness of the working model, we propose to integrate the model components step by step.

4.2 Space modelling

Because of the geographical position of different entities (e.g. VTS) over a maritime space, we need to partition the navigation space into smaller areas in order to replicate real world. This is also needed, as mentioned to overcome the scalability problem. This maritime environment can be modelled using a hierarchic approach [Bargiela 2000], which organizes information at different level of granularity. The hierarchic decomposition is particularly adapted for the modelling of real world system that have to be implemented on distributed system. This description of the different elements constituting the environment shows three essentials components. First, this scheme allows for the identification of geographical entities of the model (nodes of the tree, figure 5). Secondly, this scheme shows dynamic behaviour of these entities. Finally it shows the entity that can be mapped on the computer for the distribution (a partition).

Partitions are managed by space managers that coordinate start up and inter agent communication. They also facilitate agent migration as modelled ships move into their neighbouring partitions. Let us notice that a *buffer zone* is needed if space manager has limited view of the space it manages (existence of uncovered area).

There are several benefits of such a distributed approach: (1) it provides manageable computing object entities (agents) in order to solve scalability.

(2) it provides local reference for agent communication and thus more efficiency in communication because objects in the same area are on the same computer. (3) it reflects real structure of how sea space is managed.

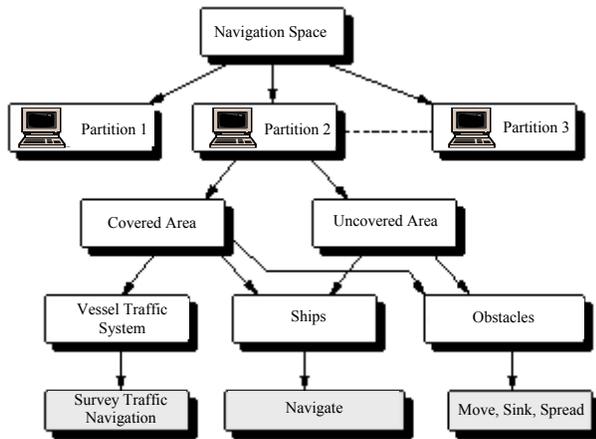


Fig. 5. Space modelling

4.3 Reliability concerns

We mentioned that naval navigation systems are safety critical, which poses reliability concerns on the proposed agent-based computing engine. The mobile agent technology is a software revolution, its goal is to utilize networked resources in a more opportunistic fashion than the traditional client/server systems, but the hardware platform on which the agents run doesn't differ from that exploited by the traditional distributed computing paradigms. Hence, the reliability of agent-based systems is affected by faults to the external agent environment, i.e. the networked hosts and the communication channel. Transient failures, caused by a temporary memory fault for example, might affect the executing agent only, while permanent faults caused by host failure will crash the running agent platform and all the executing agents. The integrity of inter-agent messaging and can be violated by faults in the communication link as well as the failure of the sending or receiving agent. Host and communication link failures can also disrupt agent migration. Our navigation model presumes that the ship agents must exchange information for track correction, collision evasion, etc., therefore we also need to address the complex problem of coordinating the recovery of the mobile agents in a collaborative environment. Here we must take into consideration how the failure of a single agent or a communication transaction can affect the consistency of the global state of collaborative agents applications. Classic distributed fault-tolerance issues such as domino effect and duplicated messages [Osman and Bargiela 2000b] are also relevant for such agent applications.

There are two main approaches to fault-tolerance of distributed systems: replication and checkpointing. Replication techniques rely on executing replicas of the application processes (agents) on redundant hardware, then the application should be able to continue executing reliably as long as at least one replica is alive. In-principle, replication techniques should have smaller overhead than checkpointing methods, that can block the application execution while retrieving a previously stored state of the agent (checkpoint) from stable storage. However, we argue that agent computing is not suitable for high-performance applications with strictly constrained response time for which such overhead can be regarded intolerable. The agent code has to be interpreted to support the portability necessary for agent mobility, which slows-down the performance of the agent code in comparison to executables that are pre-compiled into native machine code. There is also the overhead of marshalling/unmarshalling the execution code (byte-code serialization) and agent load/start-up time as agents travel their itinerary to fulfil a user task [Schill et al. 1998]. The advantage of using the agent-computing paradigm for naval navigation paradigm is in the multi-agent systems ability to logically map into the dynamic topology of roving objects (ships) and intelligent processing of their interaction in the maritime environment, rather than delivering exceptionally high performance.

Checkpointing offers a low-cost alternative to agent systems reliability, where live replicas running on redundant hardware are not required [Silva et al. 2000]. Checkpointing is easier to implement, as the management of consensus between many replicas is not required. It also fits naturally into the agent-computing model since serializing the agent code in preparation for migration effectively constitutes taking a checkpoint.

There are many flavours of checkpointing and rollback recovery: messaging logging, consistent checkpointing, and hybrid methods. The choice and detailed design of the rollback recovery protocol will fundamentally depend on the study of execution, communication, and reliability requirements of the navigation system navigation system, which will prevail in the next stage following this feasibility study. Many implementation decisions for the fault-tolerance layer will also largely depend on our choice of mobile-agent platform. The requirement to support mobility, agent tracking, message forwarding, etc. has made the agent platforms very complex middleware that significantly varies from one implementation to another. For example some agent platforms rely on weak migration [Rothermel and Schwehm 1999], i.e. proxies are used at the remote platform instead of

physically de-serializing and transporting the agent code. Weak migration significantly complicates the fault-tolerance protocol since the agents no longer have autonomous execution state.

5. CONCLUSIONS

In this position paper we presented simulation for safety navigation and identified technical limitations we encountered during the first step of an implementation of our model.

In the first part of this document, we explain why it is useful to simulate traffic navigation and presented the problem of sea traffic congestion and why we need to address it. In the second part, we criticize the traditional methods currently used to survey the ships and suggest how it can be improved by applying the multi-agent technology to represent the simulated vessels. We identify probable technical problems facing the agent-based simulator such as algorithm complexity, reliability, resource management and scalability. Then, we described a modelling approach to maritime navigation that integrates ships behaviour, and suggested a multi-agent distributed systems approach for implementing the proposed model. We explained how the autonomous and mobile nature of the multi-agent computing paradigm perfectly fits into space modelling of maritime navigation.

The next stage of this project will involve implementing the navigation simulation model on a multi-agent computing bed and testing a collision alarm system based on the proposed model. We will also perform an evaluation study of the scalability and availability results of the simulation by contrasting them to real-life data.

ACKNOWLEDGMENT

We would like to thank the French Ministry of Foreign Affairs (Alliance Program) and the British Council for their financial support.

REFERENCES

AMIEL, L. 2002. Opening up VTS systems. In *New Computer Tools for Safety Navigation – First French/Chinese Workshop*. French Naval Academy and Shanghai Maritime University, Shanghai, Shanghai Maritime University, China.

BARGIELA, A. 2000. Strategic directions in simulation and modelling. Tech. rep., The Nottingham Trent University, United Kingdom. january. UK Computing Research Strategy Meeting.

BEA MER. 2000. Rapport d'enquête sur le naufrage de l'ERIKA. Tech. rep., BEA Mer/Ministère des Transports. october. 14 pages.

BEA MER. 2001a. Rapport annuel. Tech. rep., BEA Mer/Ministère des Transports. june. 4 pages.

BEA MER. 2001b. Rapport d'enquête sur le naufrage de l'IEVOLI SUN. Tech. rep., BEA Mer/Ministère des Transports. december.

BÜRCKERT, H.-J., FUNK, P., AND VIERKE, G. 1999. An intercompany dispatch support system for intermodal transport chains. Tech. Rep. TM-99-02, Deutsches Forschungszentrum für Künstliche Intelligenz. 12 pages.

EROL, K., LEVY, R., AND WENTWORTH, J. 1999. Application of agent technology to traffic simulation. Tech. rep., U.S. Department of Transportation, McLean, Virginia, USA.

EUROPEAN COMMISSION. 2001. White Paper, European Transport Policy for 2010. Tech. rep., Directorate- General for Energy and Transport. september. 22 pages.

GOODWIN, K. AND PFRANG, M. 2001. Web delivery of real-time GIS-based traveler safety information. In *14th Annual Geo-Spatial Information Systems for Transportation Symposium*. Bureau of Transportation Statistics, Crystal Gateway Marriot, Arlington, Virginia, USA.

HADOUAJ, S. E., DROUGOUL, A., AND ESPINASSE, S. 2000. How to combine reactivity and anticipation : the case of conflict resolution in a simulated road traffic. In *MABS'2000 Workshop*, A. Bundy, Ed. LNAI. Springer Verlag, 82–xx.

HUET, T. 2002a. Amélioration des performances : Étude de la complexité de la sélection des sondes. Tech. rep., IRENav. june.

HUET, T. 2002b. Using internet to survey the maritime traffic. Tech. rep., Ecole navale – IRENav. december.

KROZEL, J. AND PETERS, M. 2000. Decentralized control techniques for distributed air/ground traffic separation. Tech. rep., Nasa Ames Research Center. june.

O'BRIAN, P. AND NICOL, R. 1998. Fipa – towards a standard for software agents. *BT Technology Journal* 16, 3.

OSMAN, T. AND BARGIELA, A. 2000a. Fadi: A fault-tolerant environment for open distributed computing. *IEEE Proc. on Software* 147, 3, 91–99.

OSMAN, T. AND BARGIELA, A. 2000b. Fault-tolerance for mobile agent systems and applications. In *Proc. Of 2000 Workshop on Agent-based Simulation*. 211–221.

PAUL, L. 1997. A vessel traffic system analysis for the korea/tsushima strait. In *Energy-Related Marine Issues In the Regional Seas of Northern Asia*. Berkeley, California, USA.

RAY, C. AND CLARAMUNT, C. 2002. A new distributed computing approach for the modelling and simulation disaggregated data flows: application to an airport system. In *Geoid International Colloquium on Integrated Land-use and Transportation Models*. Québec, Canada, 14 pages.

RAY, C. AND HUET, T. 2000. Active objects in a distributed environment. In *Middleware 2000*. New York Palissade, U.S.A. poster session.

ROTHERMEL, K. AND SCHWEHM, M. 1999. Mobile agents. In *Encyclopedia for Computer Science and Technology*, M. D. Inc., Ed. Supplement 25, vol. 40. New York, 155–176.

SCHILL, A., HELD, A., BÜHOMAK, W., SPRINGER, T., AND ZIEGERT, T. 1998. An agent based application for personalised vehicular traffic management. In *Proc. of the 2nd International Workshop on Mobile Agents*, K. Rothermel and F. Hohl, Eds. Lecture Notes in Computer Science. Springer-Verlag, Berlin, Germany, 99–111.

SILVA, L., BATISTA, V., AND SILVA, J. 2000. Fault-tolerant execution of mobile agents. In *Proc. of the International Conference on Dependable Systems and Networks*. New York, USA, 135–143.

ZHU, K., LUDEMA, M., AND HEIJDEN, R. D. 2000. Air cargo transport by multi-agent based planning. In *Proc. of the 33rd Hawaii International Conference on System Science*.



Dr Thierry Huet is Lecturer in Computer Science at the French Naval Academy. He is currently working at the Research institute of the French Naval Academy (IRENav) on safety navigation applications. He has been responsible for the success of several GIS and software development projects for European (CORINE, MEDSTAT projects), National (CNES, the French space agency) and regional (SIGMIP project) governmental organisations. Among its interests are cooperative and distributed networks for solving problems, reuse of software components, time and GIS.



Dr Taha Osman is a lecturer in the School of Computing and Mathematics at the Nottingham Trent University. He received a B.Sc. honours degree in Computing from Donetsk Polytechnical Institute, Ukraine in 1992. He joined the Nottingham Trent University in 1993 where he received an MSc in Real-time Systems in 1994 and was awarded a PhD in 1998 for his work on developing fault-tolerant distributed processing environments. His current research interest is focused on the utilisation of mobile agent systems for developing robust, open-access wireless computing environments.



Cyril Ray received a M.Sc. in computer science from University of Rennes in 1999. He is currently a PhD student in computer science at the French Naval Academy Research Institute (IRENav). From 1998 to 1999, he worked as a student at IRISA (INRIA Rennes) where, his research activities focused on software engineering and distributed systems. Within the COMPOSE team he took part in the development of profiling tools for code specialisation and automatic generation of optimised Java Bytecode. Then, within the ADP group he worked on taxonomies of distributed shared memory, and on a new protocol which aims to describe a fail-safe distributed shared memory. Within the GIS group at French Naval Academy, his Ph.D. research is oriented towards distributed simulation of complex systems and the development and application of qualitative discrete-event simulation applied to the modelling and simulation of transportation systems.

FAST LEARNING NEURAL NETS WITH ADAPTIVE LEARNING STYLES

DOMINIC PALMER-BROWN, SIN WEE LEE, JON TEPPER and CHRIS ROADKNIGHT

*Computational Intelligence Research Group,
School of Computing, Leeds Metropolitan University,
Beckett Park, Leeds LS6 3QS (d.palmer-brown@lmu.ac.uk).*

Abstract - There are many learning methods in artificial neural networks. Depending on the application, one learning or weight update rule may be more suitable than another, but the choice is not always clear-cut, despite some fundamental constraints, such as whether the learning is supervised or unsupervised. This paper addresses the learning style selection problem by proposing an adaptive learning style. Initially, some observations concerning the nature of adaptation and learning are discussed in the context of the underlying motivations for the research, and this paves the way for the description of an example system. The approach harnesses the complementary strengths of two forms of learning which are dynamically combined in a rapid form of adaptation that balances minimalist pattern intersection learning with Learning Vector Quantization. Both methods are unsupervised, but the balance between the two is determined by a performance feedback parameter. The result is a data-driven system that shifts between alternative solutions to pattern classification problems rapidly when performance is poor, whilst adjusting to new data slowly, and residing in the vicinity of a solution when performance is good.

Keywords: neural networks, fast learning, performance feedback, adaptive learning styles.

1. MOTIVATIONS AND OBJECTIVES

There are some basic observations and principles that motivate research into neural networks and other systems that are capable of leaning ‘on the fly’. These concern the ability to rapidly adapt to discover provisional solutions that meet criteria imposed by a changing environment.

1.1 Provisional Learning

The adaptive systems of interest in this type of research are not required to solve an optimisation problem in the traditional sense; they search heuristically for good solutions (solutions that are fit for purpose according to the chosen criteria of the target application) in a hyperspace that may contain many plausible solutions. However, heuristic information may be expressed by an objective function of some kind, which the system tries to ‘optimise’. The classic example is error minimisation, in which in general the data is imperfect, e.g. limited, sparse, missing, error-prone, and subject to change (non-stationary). Therefore, the error minimum is really just a local minimum: local to a subset of data and an episode of time.

Whilst this does not preclude the discovery of solutions that work for all data-time, it does mean that such generalisation involves extrapolations and assumptions that cannot be justified on the sole basis of the available information. In such circumstances, it is reasonable, when a new

candidate solution is found, for it to be held – as a provisional hypothesis – until or unless it is rejected, or until it can be replaced by a stronger hypothesis.

1.2 Fast Learning

Slow, iterative and intensive sampling based methods (eg. Gradient descent methods, and Bayesian methods involving Monte Carlo and related methods) are *inherently* non-real-time, in the sense that they require multiple presentations of sets of patterns or samples, and therefore they cannot respond to the changing environment *as it is* changing. This contrasts sharply with the human case. Humans learn ‘as they go along’, to a significant extent, without the need for multiple presentations of each exemplar or pattern of information.

1.3 Performance-guided Learning

An important concern in artificial intelligence is how to combine top-down and bottom-up information. This applies to learning systems. For example, reinforcement learning is very effective at rewarding successful strategies, or moves, during learning; supervised learning is a powerful means of modifying an ANN when it makes mistakes; and genetic algorithms are effective at selecting for improvement across generations of solutions. These are important and effective approaches, not to be dismissed simply because they are not fast, or because they are computationally intensive.

Fascinating results and innovations are still occurring with these approaches, as this conference testifies [Vieira et al 2003, Andrews 2003, Lee et al 2003]. Equally unsupervised learning, which does not harness top-down information, is an extremely useful tool, for example as an alternative or complement to clustering; but in its purest form it does not (by definition) make use of any information on the current performance of learning, in order to guide adaptation in appropriate directions.

Ideally, learning should be rapid, and yet capable of taking external indicators of performance into account; and it should be capable of reconciling the data (bottom-up) with feedback concerning how the ANN is organising the data (top-down).

2. ADAPTIVE RESONANCE

The points raised above have led to the development of PART (Performance-guided Adaptive Resonance Theory), which has two of antecedents, ART (the original Adaptive Resonance Theory), and SMART (Supervised Match-seeking ART).

2.1 Adaptive Resonance Theory (ART)

ART [Carpenter and Grossberg, 1988] performs unsupervised learning. A winning node is accepted for adaptation if:

$$\frac{|w \cap I|}{|I|} \geq \rho, \text{ where } w \text{ is the weight vector, } I \text{ is}$$

the input vector and ρ is the so-called vigilance parameter, which therefore determines the level of match between the input and the weights required for a win. Weight adaptation is governed by: $w_{ij}^{(new)} = n(I \cap w_{ij}^{(old)}) + (1-n)(w_{ij}^{(old)})$. As a result, only those elements present in both I and w remain after each adaptation, and learning is fast. In fact, it is guaranteed to converge in 3 passes of any set of patterns when $n=1$.

2.2 Supervised Match-seeking Adaptive Resonance Tree (SMART)

In order to convert ART into a supervised learning system that would therefore learn prescribed problems, SMART was developed [Palmer-Brown, 1992]. In this case the winning nodes are labelled with a classification. When a node with a label wins, if the classification is correct, learning proceeds as usual. If the class is wrong, a new node is initialised with the values of I , so that it would win in competition with the current winning node. An upper limit may be imposed on the number of nodes, in which case further learning results in

some nodes becoming pointers to subnets, which learn in the same way as the first net. Hence the system is a fast, self-growing network tree.

2.3 Information Loss

The main limitation that was found with ART and SMART was the ‘one strike and you’re out’ nature of the adaptation. Nodes sometimes need to retain information that is relevant to only a subset of the patterns for which they win. The $w \cap I$ intersection is responsible for this information loss, but it is also the reason for the rapidity and stability of the learning process. Thus, the challenge is to retain these positive characteristics whilst preventing the learning from throwing away information when it is needed. This objective, along with the points made in section 1, has led to the development of Performance-guided Adaptive Resonance (PART).

3. PERFORMANCE-GUIDED ADAPTIVE RESONANCE (PART)

A non-specific performance measure is used with PART because, in many applications, there are no specific performance measures (or external feedback) available in response to each *individual* network decision. PART consists of a distributed network and a non-distributed network, in order to perform feature(s) extraction followed by feature classification, in two stages. Fig. 1 illustrates the architecture in the context of a particular application [Sin Wee et al, 2002].

3.1 dP-ART Learning

On the presentation of a binary input pattern I , the network categorises the input pattern by comparing it against the stored knowledge in the existing distributed output categories of $F2_1$ layer. This is achieved by calculating the bottom-up activation, using (1):

$$T_i = \frac{|w_i \cap I|}{\beta + |w_i|} \quad (1)$$

As this architecture is based on a *distributed* P-ART, there is more than one winning node, in this case $D = 3$. The $F2_1$ nodes with the highest bottom-up activation are selected (D of them), and they have their weights adapted. If a distributed output category is found with the required matching level, using (2), as in ART:

$$\frac{|wI \cap I|}{|I|} \geq \rho \quad (2),$$

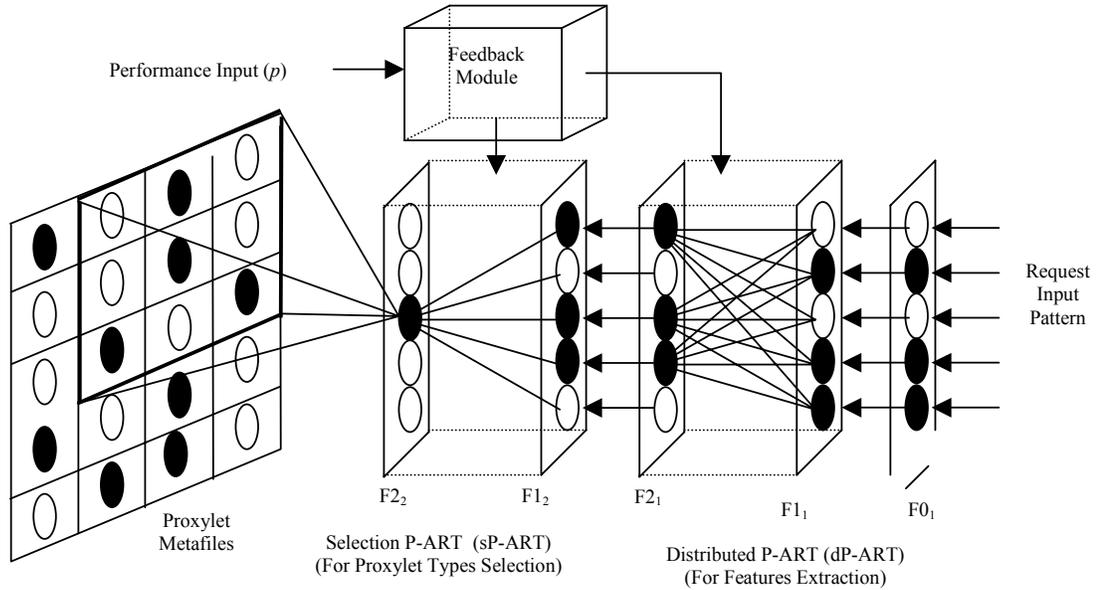


Figure 1. The PART System

Then learning occurs, according to equation (3):

$$w_{ij}^{(new)} = (1 - p) (I \cap w_{ij}^{(old)}) + p (w_{ij}^{(old)} + \beta (I - w_{ij}^{(old)})),$$

where $w_{ij}^{(old)}$ = the top-down (a similar equation applies for the bottom-up weights) vectors at the start of the input presentation; p = performance parameter; I = binary input vector; and β = the 'drift' constant. The effect of (3) is $w_{ij}^{(new)} = \alpha$ (fast_ART learning) + β (LVQ) (equation 4), where LVQ stands for Learning vector quantization.

The α - β balance is determined by performance feedback. Therefore P-ART does unsupervised learning, but its learning style is determined by its performance, which may be updated at any time. So PART combines minimalist ART learning with Learning Vector Quantization (LVQ) [Kohonen, 1990], and by substituting p in (3) with 0 for poor performance, (3) can be simplified to:

$$w_{ij}^{(new)} = (I \cap w_{ij}^{(old)}) \quad (5)$$

Thus, fast learning is invoked, causing the weights to reach their new asymptote on each input presentation:

$$w_j \rightarrow I \cap w_j^{(old)} \quad (6)$$

In contrast, for excellent performance where $p = 1$, (3) can be simplified to:

$$w_{ij}^{(new)} = (w_{ij}^{(old)} + \beta (I - w_{ij}^{(old)})) \quad (7)$$

Thus, a simple form of clustering or LVQ occurs at a speed determined by β .

Equation (3), whenever performance is not perfect, enables the top-down weights to drift towards the input patterns. With alternate episodes of $p = 0$ and $p = 1$, the characteristics of the learning of the network will be the joint effects of the (5) and (6). This joint effect enables the network to learn using fast and convergent, 'snap' learning when the performance is poor, yet be able to drift towards the input patterns when the performance is good. Drift will only result in slow (depending on β) reclassification of inputs over time, keeping the network up-to-date, without a radical set of reclassifications for exiting patterns. By contrast, snapping results in rapid reselection of a proportion of patterns to quickly respond to a significantly changed situation, in terms of the input vectors (requests) and/or of the environment, which may require the same requests to be treated differently. Thus, a new classification may occur for one of two reasons: as a result of the drift itself, or as a result of the drift enabling a further snap to occur, once the drift has moved weights away from convergence.

3.2 sP-ART

The distributed output representation of categories produced by the dP-ART acts as input to the sP-ART. The architecture of the sP-ART is the same as that described above except that only the F2₂ node with the highest activation is selected for learning. The effect of learning within sP-ART and

dP-ART is that specific output nodes will represent different groups of input patterns until the performance feedback indicates that sP-ART is indexing the correct outputs (called proxylets in the target application).

4 AN EXAMPLE APPLICATION

4.1 The Performance Feedback

The external performance feedback into the P-ART reflects the performance requirement in different circumstances. Various performance feedback profiles in the range $\{0,1\}$ are fed into the network to evaluate the dynamics, stability and performance responsiveness of the learning. Initially, some very basic tests with performances of 1 or 0 were evaluated in a simplified system [Sin Wee et al, 2002]. Below, the simulations involve computing the performance based on a parameter associated with the winning output neuron. In the target application, provided by BT [Marshall and Roadknight, 2000, 2001], factors which contribute to good/poor performance include latencies for proxylet (eg software) requests with differing time to live, dropping rate for request with differing time to live, and different charging levels according to quality of service, and so on.

4.2 Application Layer Active Network (ALAN)

The ALAN architecture was first proposed by [Fry and Ghosh, 1999] to enable users to supply JAVA based active-service codes known as *proxylets* that run on an edge system (Execution Environment for Proxylets – EEPs) provided by the network operator. The purpose of the architecture is to enhance the communication between servers and clients using the EEPs, that are located at optimal points of the end-to-end path between the server and the clients, without dealing with the current system architecture and equipment. This approach relies on the redirecting of selected request packets into the EEP, where the appropriate proxylets can be executed to modify the packet's contents without impacting on the router's performance.

In this context, P-ART is used as a means of finding and optimising a set of conditions that produce optimum proxylet selections in the Execution Environment for Proxylets (EEP), which contains all the frequently requested proxylets (services).

4.3 Simulations

The test patterns consist of 100 input vectors. Each test pattern characterizes the features/properties of a realistic network request, such as bandwidth, time, file size, loss and completion guarantee. These test patterns were first presented in random order for 25 epochs where the performance, p , is calculated

according to the average bandwidth of selections. This continuous random-order presentation of test patterns simulates the real world scenario where the order of patterns presented is such that a given network request might be repeatedly encountered, while others are not used at all.

4.4 Results of simulations

In Figure 2, we show the performance calculated across the simulation epochs. An epoch consists of 50 patterns, randomly selected. Performance feedback is updated at the end of each epoch. The network starts with low performance and the performance feedback is calculated and fed into the dP-ART and sP-ART after every simulation epoch, to be applied during the following epoch. Epochs are of fixed length for convenience, but can be any length. Fig. 3 shows the selection frequency of the proxylet type. In this case, we have the following bandwidth bands: Low bandwidth proxylet: $0 \rightarrow 600$ Kb/s; Median bandwidth proxylet type: $601 \rightarrow 1200$ Kb/s; High bandwidth proxylet type: >1201 Kb/s.

At the first epoch (refer to Fig. 2), the performance is set to 0 to invoke fast learning. A further snap occurs in epoch 7 since low performance has been detected. Note that during epochs 7 and 8, there is a significantly higher selection of high bandwidth proxylet types, caused by the further snap and continuous new inputs that feed into the network. As a result, performance has been significantly increased at the start of ninth epoch.

At epochs 16, 20 and 27, from Fig. 2, there is a significant decrease in performance. As illustrated in Fig. 3, this is due to a significant increase in the selection of low bandwidth proxylet types and a decrease in high bandwidth proxylets. This is due to the drift that has occurred since the last snap, with a number of patterns still appearing for the first time. The performance induced snap takes the weight vectors to new positions. Subsequently, a similar episode of decreased performance occurs, for similar reasons, and a further snap in a different direction of weight space follows, enabling reselections (reclassifications), resulting in improved performance.

By the 28th epoch, where $p = 0.81$, the performance has stabilised around the average performance of 0.85. At this stage, most of the possible input patterns have been encountered several times. Until new input patterns are introduced or there is a change in the performance circumstances, the network will maintain at this high level of performance.

As shown in Fig. 4, the average proxylet execution time is introduced into the performance criterion

calculation to encourage the selection of high execution time proxylet types. In this case, we have the following execution time bands: Short execution time proxylet: 1 → 300 ms; Median execution time proxylet type: 301 → 600 ms; Long execution time proxylet type: > 600 ms.

This criterion is fed into the P-ART at every 100th epoch. The result indicated when the new performance criterion is introduced in the 100th epoch, rapid reselection of a proportion of the patterns occurs on a consistent basis.

Other parameters such as cost, file size will be added to the performance calculation to produce a more realistic simulation of network circumstances in the future.

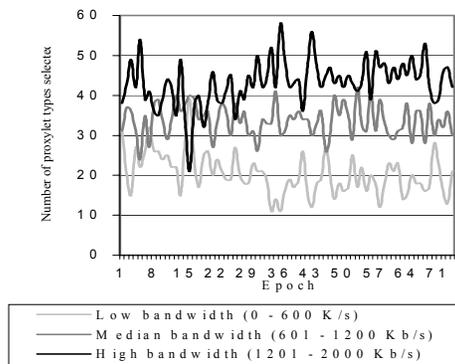


Fig 2 Performance levels of the network
Figure 2. Performance.

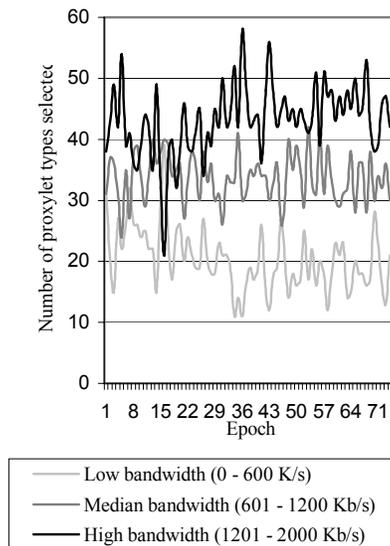


Fig. 3 Selection frequency of the 3-bandwidth bands of proxylet types at each epoch.

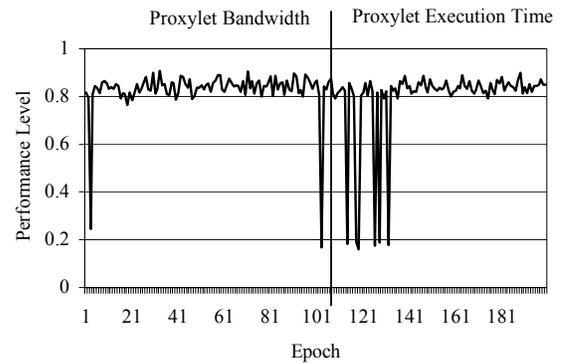


Fig. 4. Performance level before/after a problem change.

5. CONCLUSION

The PART system is able to adapt rapidly to changing circumstances. It manages to reconcile top-down and bottom-up information by finding a new provisional solution to the pattern classification problem whenever performance deteriorates. There is clearly potential to apply this approach to a wide range of problems, and to develop it in order to fully explore the objectives stated in section 1.

REFERENCES

- S. G Andrews. "Novel neural network methods for describing attributes contained within lesions images". In Proc. of ESM2003.
- G.A. Carpenter, S. Grossberg, "The ART of Adaptive Pattern Recognition by a Self-Organising Neural Networks", *IEEE Computer*, vol. 21(3), pp. 77 – 88, 1988.
- T. Kohonen, "Improved Versions of Learning Vector Quantization", *International Joint Conference on Neural Networks*, San Diego, vol I, 545 – 550, 1990.
- T. Kohonen, "The Self-Organizing Maps", *Proceeding of the IEEE*, vol. 78(9), pp. 1464 –1480, 1990.
- J Lee, S Mian, R Rees, and G Ball. "Preliminary Artificial Neural Network Analysis of SELDI Mass Spectrometry Data for the Classification of Melanoma Tissue". In Proc. of ESM2003.
- S.W. Lee, D. Palmer-Brown, J. Tepper and C.M. Roadknight, "Performance-guided Neural Networks for Rapidly Self-Organising Active Network

Management”, in: *Soft Computing Systems: Design, Management and Application*, A. Abraham, J. Ruiz-del-Solar, and M. Koppen, Eds., Netherland: IOS Press, 2002, pp. 21 - 31.

S.W. Lee, D. Palmer-Brown, J. Tepper and C.M. Roadknight, “Performance-guided Neural Network for Self-Organising Network Management”, *Proceeding of London Communications Symposium*, University College London, pp. 269 – 272, Sep 2002.

M. Fry and A. Ghosh, “Application Layer Active Network”, *Computer Networks*, vol. 31(7), pp. 655 – 667, 1999.

I.W. Marshall and C.M. Roadknight, “Provision of Quality of Service for Active Services”, *Computer Networks*, vol. 36(1), pp. 75 – 85, 2001.

I.W. Marshall and C.M. Roadknight, “Differentiated Quality of Service in Application Layer Active Networks”, in: *Active Networks, LNCS 1942*, Yasuda, Eds., Springer-Verlag, 2000, pp. 358-371.

D. Palmer-Brown, “High Speed Learning in a Supervised, Self Growing Net”, in: *Proceeding of ICANN 92*, I. Aleksander and I. Taylor, Eds., Brighton, vol. 2, pp. 1159-1162, 1992.

C. Vieira, P Mather, P Alpin. “Improving Artificial Neural Network Performance by Using Temporal-Spectral Features for Agricultural Crop Classification”. In Proc. of ESM2003.

IMPROVING ARTIFICIAL NEURAL NETWORK PERFORMANCE BY USING TEMPORAL-SPECTRAL FEATURES FOR AGRICULTURAL CROP CLASSIFICATION

CARLOS ANTONIO OLIVEIRA VIEIRA¹
PAUL MATHER²
PAUL APLIN²

¹*UFV - Universidade Federal de Viçosa
Departamento de Engenharia Civil,
Campus da UFV-DEC Viçosa MG 36.571-000, Brazil
carlos.vieira@ufv.br*

²*The University of Nottingham
The School of Geography,
University Park, Nottingham, NG7 2RD, UK
Paul.Mather@nottingham.ac.uk and Paul.Aplin@nottingham.ac.uk*

Abstract: A method for improving artificial neural network performance by using multi-temporal, multi-spectral and multi-source remotely-sensed data as features for classifying agricultural crops is described. The procedure characterizes all the pixels in a scene by considering their intensity values as a function of time of imaging and spectral waveband. An analytical surface is interpolated through these data points, which may be irregularly spaced. Two fitted function interpolation methods were used to generate and parameterize the analytical surfaces. Then, the surface coefficients were input to two different supervised classifiers (Maximum Likelihood and Artificial Neural Network algorithms). Results show that classification accuracy is significantly improved in comparison with the use of any single-date image. Classification accuracies in excess of 87% were achieved. The advantages of the methodology described in this paper is that it takes account of the reflectance spectra at different points in the growing season, and that the time periods between images, as well as the wavebands, need not be the same at each date. Thus, the procedure can handle data from sensors such as SPOT HRV and Landsat TM. In addition, the use of coefficients to represent the analytical surfaces significantly reduces the amount of data processing, whilst maintaining information reliability.

Keywords: image classification, multitemporal classificatios, multi-source classification, neural networks.

1. INTRODUCTION

Remote sensing has been used to provide input data by aerial measurements for many agricultural applications, including monitoring crop production and yield forecasting from very early days [Steven et al., 1997]. Many companies and governments require a forecast to plan their processing requirements and marketing. Although remote sensing is already an established forecasting tool for agricultural applications, traditional methods (e.g., aerial photographs) are not able to cover enough samples, or wide enough areas. Therefore, it is important for crop yield forecasting to expand its boundaries to incorporate images from orbiting platforms, since these kinds of images can provide a much better statistical sample for large areas. Thus, crop yield prediction by satellite observation could become a commercial reality.

Before one can apply a forecasting model to particular crops, it is necessary to separate them from all other cover types. This identification process is referred to as

classification. Although there are many classification strategies available, problems remain in getting the best accuracy performance from a given classification method. The classification strategy and its parameters may be inadequate; or, the features used in the classification process may not be well-suited to the technique of crop identification, thus causing a lower accuracy performance; or perhaps, the available spatial resolution and temporal frequency of the data is not matched to the expected accuracy.

For practical applications, it is essential that classification systems be robust and exhibit good generalisation. Although the range of image processing techniques has been greatly expanded, from classical statistical approaches to neural network methods, there is no single classification algorithm capable of deriving generic products from remotely sensed data. The performance of these algorithms is strongly

dependent upon data selection and on the efforts devoted to the design phase. Therefore, researchers must seek alternative methods for achieving improved generalisation performance.

Efficient crop management practices require accurate and rapid information about crop distributions. Commonly, multispectral remotely sensed images are used to distinguish crop types on the basis of their spectral properties [Mather, 1999]. However, such analysis involving single-date images has the drawback that, since maximum discrimination between different crop types occurs at different stages in the growth cycle, not all differences are incorporated in the procedure. Moreover, different crop types represented in the area under study may be at different stages of growth. In addition, the temporal 'profile' of the spectral reflectance curve of each crop is not taken into account. Such profiles may be of considerable value in discriminating between crop types, which may be difficult to distinguish at certain points in the growth cycle. Furthermore, results derived from data obtained by different sensors may not be comparable due to differences in spectral and spatial characteristics. Finally, since agricultural crops are dynamic, it is often useful to observe their development over time (e.g., crop yield estimation). A solution is to use multitemporal images for crop monitoring [Badhwar et al., 1982]. For most current multitemporal classification techniques, a correspondence of time to growth state is established for each possible crop category that minimises the smallest difference between the given multispectral-multitemporal vector and the category mean vector indexed by growth state [Haralick et al., 1980]. These techniques, however, are fairly inaccurate since only relatively few static spectral and temporal 'snapshots' contribute to crop identification. That is, images with specific spectral wavebands acquired on specific dates are used, rather than images with entire spectral and temporal continua. Using the latter may increase crop classification accuracy since they contain more information than the former [Labin and Strahler, 1994].

This paper demonstrates a method for improving artificial neural network performance by using the spectral-temporal signatures of remotely sensed images as features for classifying agricultural crops. Per-pixel classifications are performed using multispectral, multitemporal and multisource data, whereby analytical surfaces representing the spectral and temporal continua of each feature (pixel) are interpolated and their coefficients are used as discriminating variables.

2. STUDY AREA AND DATA SET

The study area was located near the town of Littleport in Cambridgeshire, eastern England. This area was approximately at mean sea level with gently undulating

topography. The agriculture of the region was characterized by rotational crop plantation techniques.

Eight remotely sensed images acquired throughout the 1994 summer growing season were used for analysis. These included four Landsat TM images (11 June, 27 June, 20 July, 14 August) and four SPOT HRV images (13 May, 28 June, 30 July, 14 August). Only six spectral wavebands of Landsat TM imagery were used since the thermal infrared band (band 6) was omitted from analysis. In addition, local farmers' Field Data Printouts for 1994 were collected and used to generate a ground reference data set.

All images were geometrically registered to the British National Grid. For each image, registration was performed using 17 ground control points and nearest neighbor re-sampling, since this technique maintained the original pixel values [Jensen, 1986]. In each case, the root-mean-square error associated with registration was less than 0.5 pixels.

Atmospheric correction was performed to account for atmospheric differences between multitemporal images. Initially, image digital numbers were corrected to radiance using information supplied with the image data files [Teillet and Fedosejevs, 1995]. Radiance was then converted to apparent reflectance (recorded at the sensor) and finally to surface reflectance. The final step used an inversion of the 5S (Simulation of the Satellite Signal in the Solar Spectrum) model [Tanré, 1990].

3. THE SPECTRA-TEMPORAL RESPONSE SURFACES (STRS) MODEL

[Badhwar et al., 1982], [Badhwar, 1984], [Haralick et al., 1980], [Lambin and Strahler, 1994] and [Ortiz et al., 1997] consider the problem of characterizing the temporal dimension but none utilizes the method proposed by [Vieira et al., 1998, 2000], involving the use of the spectra-temporal response surfaces (STRS), which provide for the generalisation in time of spectral reflectance properties of agricultural areas. The type and sequence of procedures used in the generation and potential use of the STRS representations are outlined in **Figure 1**.

The STRS approach is based on a view of multi-band and multitemporal imagery from different sources represented in a three-dimensional space, the axes of which are time (x), spectral waveband (y) and reflectance (z). Measurement from a number of different sensors in the optical wavebands can be plotted

in this space. A bivariate polynomial of the form: $z = F(x,y)$, where $F()$ indicates a polynomial function of some order, is generated for each of the crop types in the area of study. Two methods were used in order to generate the fitted surfaces: polynomial trend surface analysis (PTS) and collocation (COL), since fitted function interpolation can impose a prescribed general behavior on the surface to override aberrant, anomalous, or noisy data. [Watson, 1999] and [Lam, 1983] give comprehensive reviews on these

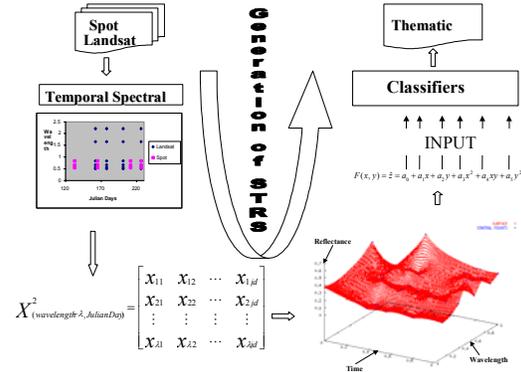


Figure 1. An outline of the methodology followed in this study to generate the STRS representations

interpolations methods and [Mather, 1976] reviews polynomial trend surfaces.

These analytical functions are then parameterized and their coefficients, rather than the pixel values in each spectral band, are used as input features in the image classification process.

4. METHODOLOGY

4.1. Sampling Techniques and Classification Phase

From the co-registered and radiometrically corrected image set, two independent sample sets (total 1440 pixels) were selected using stratified random sampling technique and representing the six most common cover types in the study area: Potatoes, Sugar beet, Wheat, Fallow, Onions, and Peas. Each sample has 120 patterns per class (total 720 pixels). One sample (selected at random) was used to training the classifier and the other one was reserved for validating the methodology.

The image acquisition dates were expressed in the form of Julian days (x -axis) and the spectral dimensions (y -axis) were characterized by their medial waveband values computed in the form of wavelengths. Thus, the spectral bands were labeled using the medial wavelength values of 0.458, 0.56, 0.66, 0.83, 1.645, 2.215 – given to the six available TM channels (except the thermal infrared TM band 6) - and 0.545, 0.645, 0.84 – given to the three HRV channels respectively.

The radiometric properties are expressed in the form of reflectance values along the z -axis. Furthermore, for

each pixel, 36 three-dimensional control points were generated (4 TM images with 6 bands plus 4 SPOT HRV images with 3 bands). It is important to mention that the values along the x , y and z axes are scaled into the interval between 0 and 1, sometimes referred to as normalization, before the interpolation phase.

Initially the control points were used to fit a surface using a Polynomial Trend Surface as described earlier. Although a surface order of 7 (36 coefficients) explained over 99% of the sum of squares, using a surface order of 3 (10 coefficients) experimentally proved to be enough to characterize the analytical surfaces. Then, the same control points were used to fit a surface using the Collocation Interpolator. As the interpolated coefficients show different magnitudes on their values, they were again scaled collectively to the interval between 0 and 1 before the training and test phases. One pixel example of the PTS and Collocation analytical surfaces is shown in **Figure 2** (a to b) for several crops.

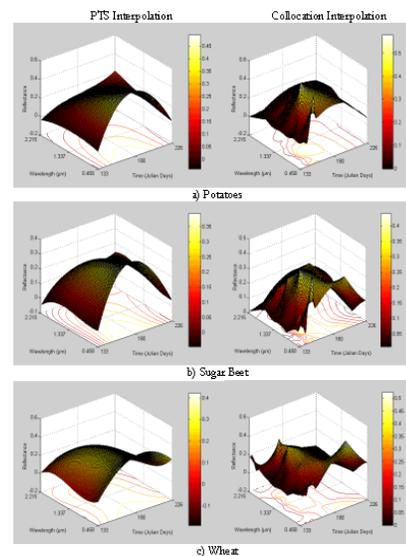


Figure 2. Analytical surfaces and contours for several crops.

According to [Vieira et al., 2000] the Maximum Likelihood (ML) classifier is the algorithm that best combines classification accuracy and computational economy when these coefficient are used as input to the classification process. Therefore, a supervised classification was performed using the Maximum Likelihood (ML) algorithm developed by [Mather, 1999] and adapted to classify 3D surface coefficients.

For the purposes of comparison, a single-date image (Landsat TM, acquired on 27th June 1994) was used to perform a standard classification in order to compare the results of

this multitemporal and multisource method against a classification based on a single-date image. For each pixel, the six reflectance values are considered together and therefore generate a six dimensional vector, to be used as input to the supervised classifiers: Maximum Likelihood (ML), and two variants of an Artificial Neural Network (ANN and ANNT).

Both artificial neural network architectures chosen are multilayer perceptrons using the backpropagation algorithm [Benediktsson et al., 1990; Bischof et al., 1992; Civco, 1993]. The only difference between the models is in the input layer. The first ANN model was implemented having one pixel per spectral band in the input layer. Therefore, this neural network had 6 nodes in the first layer. The input nodes in the ANNT model represented a 3 by 3 window of pixel data from each band of the image (total 54 nodes in the input layer) as the input [Paola, 1995]. This input modification takes local texture information into account.

All neural networks configurations tested had an output layer with 6 nodes, corresponding to the 6 general crop classes. The number of hidden layers and the number of hidden nodes were found using a building up procedure. This method, described by Hirose et al. (1991), begins with a small network composed of an input and an output layer, which are defined respectively by the number of discriminating variables and the number of classes involved in a given problem, with just one neurone in the hidden layer. The criterion for adding neurones to the hidden layer is based on the behaviour of the error during the training phase.

As the main interest in this algorithm is the minimisation of the global error, it is expected that the error will evolve to small values during training. Therefore, if after a number of cycles (e.g., 100 cycles) when the error does not decrease by more than one percent of its previous value, a new hidden unit is added and the connection weights are randomly re-initialised over the previously-defined interval. This process is repeated until the network converges to an acceptable global error value. With the above algorithm, the number of hidden units can only increase. In some cases, the number of hidden units becomes rather large, hence a counter-strategy is used. Once the network performance is judged to be satisfactory, the most recently added hidden unit is removed until the network no longer converges. The last network to converge is then taken as the optimum choice. The learning rate and momentum were set initially at 0.2 and 0.9 respectively. The learning rate was reduced during the training to 0.1 after 1000 epochs.

For this second experiment, two sample sets were selected using stratified random sampling based on the reference image (ground truth), which was generated in the same scale and projection system as the remotely sensed data. Each sample has also 120 patterns per class (total 720). One sample set was used to train the

classifiers and another independent sample set was reserved to assess the accuracy of the classification.

4.2. Accuracy Assessment

In order to perform a systematic investigation of the relative (improvement of accuracy) cost involved in the incorporation of the temporal dimension into the crop classification process, standard accuracy measures derived from a confusion matrix were computed, using an independent test data set based on the Field Data Printouts. The measures based on the confusion matrix were overall accuracy, individual class accuracy, producer's accuracy and user's accuracy. The calculations associated with these measures are described in standard textbooks (e.g., [Mather, 1999]). The Kappa coefficient, conditional Kappa for each class, and test Z statistics, all of them widely used statistics derived from the contingency matrix, were also computed [Congalton and Green, 1999].

In addition, a pairwise test statistic for evaluating the significance of the classifiers (represented here by their respective confusion matrices), was calculated utilizing the Kappa coefficients. These results are summarized in form of a *significance matrix*, in which the major diagonal elements indicate if the respective classification result is meaningful. In this single confusion matrix case, the Z value can be computed using the formula $Z = Ka / \sqrt{\text{var}(Ka)}$, where Z is standardized and normally distributed and var is the large sample variance of the Kappa coefficient K. If $Z \geq Z_{\alpha/2}$, the classification is significant better than a random classification, where $\alpha/2$ is the confidence level of the two-tailed Z test and the degrees of freedom are assumed to be infinity. On the other hand, the off diagonal elements give an indication, again if $Z \geq Z_{\alpha/2}$, that the two independent classifiers are significantly different. The formula used to test significance of the difference of the two independent Kappa coefficients is: $Z = |Ka_1 - Ka_2| / \sqrt{\text{var}(Ka_1) + \text{var}(Ka_2)}$, where the Ka_1 and Ka_2 are the two Kappa coefficients in comparison [Congalton and Green, 1999].

5. RESULTS AND DISCUSSIONS

Classification accuracies for six agricultural crops using the six multispectral bands of a single-date TM Landsat image, Polynomial Trend Surface (PTS) and Collocation as input features into three supervised classification algorithms - maximum likelihood (ML),

artificial neural networks (ANN) and artificial neural network texture (ANNT) are presented in **Table 1**. Individual classification accuracy for each crop (Conditional Kappa * 100), overall accuracy, the value of the Kappa coefficients and their variances, and test Z statistic are reported in this table. These accuracies were calculated from an independent dataset (720 patterns). The pixels received the label of the output class having the highest probability.

Table 1. Classification accuracies for six agricultural crops using Single-Date LANDSAT Image, Polynomial Trend Surface (PTS) and Collocation (COL) and three classification algorithms - maximum likelihood (ML), artificial neural networks (ANN) and artificial neural network texture (ANNT). The table shows individual classification accuracy for each crop (Conditional Kappa * 100), overall accuracy, the value of the Kappa coefficients and their variances, and test Z statistic. If the absolute value of the test Z statistic is greater than 1.96, the result is significant better than a random classification at the 95% confidence level. These accuracies were calculated from an independent dataset test (720 patterns).

INTERPO.	LANDSAT(27/06/94)			STRS	
	ML	ANN	ANNT	PTS-ML	COL-ML
Potatoes	64.5	66.8	71.9	95.9	94.9
Sugar Beet	53.8	57.9	58.6	73.8	75.3
Wheat	70.9	75.6	95.5	89.3	92.8
Fallow	80.4	81.8	79.7	70.8	63.3
Onions	84.9	89.7	88.0	95.9	97.8
Peas	53.5	67.9	80.8	93.0	100.0
OVERALL(%)	72.9	77.6	81.7	87.4	87.2
Kappa	0.675	0.732	0.780	0.848	0.847
Variance	0.000394	0.000347	0.000299	0.000219	0.000222
Z	33.99	39.28	45.09	57.27	56.88

As the absolute value of the test Z statistic is greater than critical value of 1.96, all the classification results are significant better than a random classification at the 95% confidence level. Moreover, it is noteworthy that the level of accuracy was gradually improved by employing on the single-date Landsat image the different classifiers: ML (72.9%), ANN (77.6%) and ANNT (81.7%) respectively. However, the overall performance level attained with the features generated using the STRS (i.e., the PTS and Collocation coefficients) as input features to ML classifiers were considerably greater (by 5.7%) than those obtained by a single-date image. The ML classifier, when compared to ANN classifier, is the algorithm that best combines classification accuracy and computational economy when these coefficients are used as inputs to the classification process [Vieira et al., 2000]. Oddly, fallow (or set-a-side) is the only individual category for which the accuracy was decreased using PTS and Collocation features. Therefore, it could be concluded that using these features, the ML classifier is confused by some residual patterns of crops growing in the field from the previous crop rotation, which sometimes happens on fallow land.

The lower performance achieved with ML classifier using only the TM multispectral bands is believed to be due in part to a non-linear separability of the classes under study and to a magnitude of training data set

inconsistent with the design properties and assumptions of the supervised maximum likelihood algorithms. Moreover, for some of the crops (e.g., sugar beet and potatoes, or onions and peas) the multispectral profiles for that date are not very well separated. Even so, the neural models produce a satisfactory performance on the same data set. Furthermore, the separability of the classes are considerably improved when the local spatial variance of individual pixels is implicitly taken as input to the neural network model by employing a 3 x 3 window as implemented in the ANNT algorithm.

Table 2 provides the computed Z values for a pairwise statistical test in order to check the significance of the improvements on the classification accuracy. The classification accuracy obtained using the STRS approach (PTS and Collocation using ML algorithm) were found to be significantly improved in relation to the individual classifiers ML, ANN and ANNT, in which only a multispectral single-date image was used as discriminate variables (see yellow pair, $Z > 1.96$ at 95% of confidence level). This demonstrates a need to utilise the STRS approach if one is to achieve the highest accuracies possible in crop discrimination. Moreover, there is no significant difference between the performance of the ML using PTS or Collocation coefficient as input features (see blue pair, $Z = 0.05 < 1.96$). Therefore, it could be concluded that, for this data set, these two sets of feature variables may 'work together' because they produce approximately equal classifications.

Table 2. Results of Kappa Analysis for comparison among the classifiers. The table also presents the Kappa coefficients and variance for each classifier. The Z values (in major diagonal and off diagonal elements) were computed using formula as describe in subsection 4.2.

CLASSIF	ML	ANN	ANNT	TSA	COL
KAPPA	0.675	0.732	0.78	0.848	0.847
VAR	0.000394	0.000347	0.000299	0.000219	0.000222
ML	34.01				
ANN	2.09	39.30			
ANNT	3.99	1.89	45.11		
TSA	6.99	4.88	2.99	57.30	
COL	6.93	4.82	2.94	0.05	56.85

As expected, the use of neural network models significantly overcomes the performance of the ML classifier using a single date Landsat TM image. However, the results indicate that there are no significant differences in performance between the ANN and ANNT algorithms ($Z = 1.89 < 1.96$) at the same confidence level.

6. CONCLUSIONS

A method for improving artificial neural network performance by using multi-temporal, multi-spectral and multi-source remotely-sensed data as features for classifying agricultural crops has been shown to be effective in identifying general agricultural crop classes over an area in East Anglia (UK). Classification accuracies in excess of 87% were achieved, even though parts of some of the images are covered by clouds. The basic assumption of the method, that different crops have different spectral-temporal trajectories, has been used in earlier studies. However, the methods used to characterize the spectral reflectance changes over a growing season using a spectral-temporal surface represents a promising new approach, for several reasons. First, the method can deal with multi-sensor data, as the spectral bands measured at each date do not need to be the same. Second, data points obscured by clouds can be filtered out throughout the interpolation and parameterization procedures of the analytical surfaces. Third, the overall spectral variation of a given crop class over the growing season is captured by a set of coefficients, which are fewer in number than the training data pixels and hence produce computationally more efficient classifiers.

7. ACKNOWLEDGEMENTS

Preliminary research by Dr. Vieira was supported by the Brazilian Research Council (CAPES). The later stages of this study were conducted as part of the FET-ENVIS project, a European Commission-funded RTD (Proposal Number IST-1999-29005) performed in collaboration with Nansen Environmental and Remote Sensing Centre, Norway and Ruhr-Universität Bochum, Germany. We are grateful to Logica PLC and SPOT Image for permission to use their images. Computing facilities were provided by the School of Geography, The University of Nottingham, as well as the Civil Engineering Department of the Federal University of Viçosa - Brazil.

REFERENCES

Badhwar, G. D., Autin, W. W., Carnes, J. G., 1982, "A semi-automatic for multitemporal classification of a given crop within a Landsat scene". *Pattern Recognition*, v. 3, Pp. 217-230.

Badhwar, G. D., 1984, "Classification of corn-soybean using multitemporal Thematic Mapper data", *Remote Sensing of Environment*, v. 16, Pp. 175-182.

Benediktsson, J. A.; Swain, P. H. and Ersoy, O. K., 1990, "Neural network approaches versus statistical methods in classification of multisource remote sensing data". *IEEE Trans. on Geoscience and Remote Sensing*, GE-28, Pp. 540-552.

Bischof, H., Schneider, W., and Pinz, A. J., 1992,

neural networks". *IEEE Trans. on Geosc. and Remote Sensing*, v. 3, Pp. 482-490.

Civco, D. L., 1993, "Artificial neural networks for land-cover classification and mapping". *International Journal of Geographic Information Systems*, v. 7, Pp. 173-183.

Congalton, R. G., and Green, K., 1999, "Assessing the Accuracy of Remotely Sensed Data: Principles and Practices". Lewis Publishers, New York.

Haralick, R. M., Hlavka, C. A., Yokoyama, R., Carlyle, S. M., 1980, "Spectral-temporal classification using vegetation phenology". *IEEE Trans. on Geosc. and Remote Sensing*, GE-18/2, Pp. 167-174.

Hirose, Y., Yamashita, K. and Hijiya, S. 1991, "Back-propagation Algorithm which varies the number of hidden units". *Neural Networks*, v. 4, Pp. 61-66.

Jensen, J. R., 1986, "Introductory Digital Image Processing: A Remote Sensing Perspective". Prentice-Hall, Englewood Cliffs, NJ.

Lam, N., 1983, "Spatial interpolation methods: a review". *The American Cartographer*, v. 2, Pp. 129-149.

Lambin, E. F., and Strahler, A. H., 1994, "Indicators of land-cover change for change-vector analysis in multitemporal space at coarse spatial scales", *International Journal of Remote Sensing*, v. 10, Pp. 2099-2119.

Mather, P. M., 1976, "Computational Methods of Multivariate Analysis in Physical Geography". John Wiley and Son, Chichesters.

Mather, P. M., 1999, "Computer Processing of Remotely-Sensed Images: An Introduction". John-Wiley and Sons, Chichester, Second edition.

Ortiz, M. J., Formaggio, A. R., and Epiphano, J. C. N., 1997, "Classification of croplands through integration of remote sensing, GIS and historical database". *International Journal of Remote Sensing*, v. 18, Pp. 95-105.

Paola, J. D., and Schowengerdt, R. A., 1995, "A review and analysis of backpropagation neural networks for classification of remotely-sensed multispectral imagery". *International Journal of Remote Sensing*, v. 16, Pp. 3033-3058.

Steven, M. D., Werker, R., and Milnes, M., 1997, "Assimilation of satellite data in crop monitoring and yield prediction". In G. Guyot and T. Phulpin, (eds.), *Proceedings of the Seventh International Symposium on Physical Measurements and Signatures in remote*

Sensing, 7-11 April 1997, Courchevel, France, Rotterdam: A. A. Balkema, 853-857.

Tanré, D., Deroo, C., Duhaut, P., Herman, M., Morcrette, J. J., Perbos, J. and Deschamps, P. Y., 1990, "Description of a computer code to simulate the satellite signal in the solar spectrum: the 5S code". *International Journal of Remote Sensing*, v. 11, Pp. 659-668.

Teillet, P. M. and Fedosejevs, G., 1995, "On the dark target approach to atmospheric correction of remotely sensed data". *Canadian Journal of Remote Sensing*, v. 21, Pp. 374-387.

Vieira, C. A. O., Mather, P. M., and McCullagh, M., 2000, "The Spectral-Temporal Response Surface and its use in the multi-sensor, multi-temporal classification of agricultural crops". In *ISPRS: IAPRS*, v. 33, Amsterdam, The Netherlands, Part B2 (Amsterdam, International Society for Photogrammetry and Remote Sensing) Pp. 582-589.

Vieira, C. A. O., Mather, P. M., Tso, B. C. K., and McCullagh, M. J., 1998, "Using multi-temporal, multi-spectral and multi-source remotely sensed data to classify agricultural crops". In *RSS98 - Developing International Connections, Proceedings of the 24th Annual Conference and Exhibition of the Remote Sensing Society*, University of Greenwich, 9-11 September, Pp. 354-360.

Watson, D. F., 1999, "*Contouring: A guide to the analysis and display of spatial data*". Pergamon Press, Oxford, Pp. 101-161.

PRELIMINARY ARTIFICIAL NEURAL NETWORK ANALYSIS OF SELDI MASS SPECTROMETRY DATA FOR THE CLASSIFICATION OF MELANOMA TISSUE.

LEE J LANCASHIRE¹
SHAHID MIAN¹
ROBERT C REES¹
GRAHAM R BALL^{1*}

¹The Nottingham Trent University, School of Science, Clifton Campus, Clifton Lane, Nottingham, NG11 8NS, UK.

Correspondence author: graham.balls@ntu.ac.uk

Abstract: Over recent years studies have shown an increasing application of bioinformatics tools, and in particular, artificial intelligence techniques such as artificial neural networks (ANNs) for biological problems. Proteomic techniques such as SELDI-MS (Surface Enhanced Laser Desorption/Ionization Mass Spectrometry) may be used to distinguish patterns derived from diseased tissue used to identify biomarkers representative of a certain pathological state. This paper describes research building upon studies by Ball *et al.* (2002) in identifying potential biomarkers using ANNs for the analysis of mass spectrometry data from melanoma tissue. This approach utilised ANNs to model for 72 melanoma tissue samples (36 stage 1 and 36 stage 4) to identify ions as potential biomarkers and any possible interactions between these. Preliminary results have shown that approximately 20,000 inputs can be screened to just 20 molecular ions which are capable of accurately predicting tumour grade. Using the additive approach described by Ball *et al.* (2002) individual molecular ions were used to predict tumour grade and model performance evaluated using a receiver operating characteristic (ROC) curve. The accuracy of the model with 1 ion was 58.3% with a sensitivity of 63.9% and specificity of 52.8%. With a 2 ion model, ANN performance increased to an accuracy of 70.8%, sensitivity of 66.7% and specificity of 75%. With a 3 and 4 ion model, the accuracy increased yet further to 77.8 and 83.3%, sensitivity to 72.2 and 77.8% and specificity of 83.3 and 88.9% respectively. Preliminary findings indicate the ANN approaches adopted allow optimisation and determination of the minimum number of ions (derived from SELDI-MS data) which can successfully predict tumour grade. This work continues so that these key ions may be determined in order to identify if they have an important role in tumour progression from low to high grade.

keywords: Artificial neural networks, methodologies, models and algorithms.

1. INTRODUCTION

Melanoma is a form of skin cancer which is difficult to treat if not detected early. If however, it is detected in its early stages, survival rate is promising. Therefore, new technologies need to be developed which either (i) detect the disease at an early stage, so that it can be treated before it progresses or (ii) identify biomarkers representative of a given pathological state, which may in turn be used in the development of novel treatments.

Mass spectrometry is an important tool which is used in linking proteins to their genes [Yates, 1998]. SELDI-MS is capable of rapidly analysing samples containing vast amounts of proteins with excellent reproducibility. It can be used in generating patterns that these masses of proteins produce, and therefore is useful in showing the differences between these patterns when the proteins are being expressed in different tissues, such as differences between tissues during various stages of disease. Techniques such as this may be

used to search for biomarkers associated with tumour progression and/or used in the early detection of the disease. Due to the vast amount of data generated by SELDI-MS, the development of robust computer algorithms is an absolute necessity [Ball *et al.*, 2002]. For this reason, this study involved using artificial neural networks to analyse SELDI-MS data from melanoma tissue.

ANNs are presently being utilised more than any other learning tool in biotechnology, in the modelling of complex data. This is particularly true in the field of cancer [Almeida, 2002]. Examples of their uses have been shown in prostatic cancer [Porter *et al.*, 2002; Batuello *et al.*, 2001], cervical cancer [Mango, 1998], lung cancer [Zhou *et al.*, 2002], ovarian cancer [Petricoin *et al.*, 2002] and breast cancer [Abbass, 2002; Simpson *et al.*, 1995], where ANNs have been shown to perform significantly better than physicians in the diagnosis of malignant and benign calcifications on mammograms [Markopoulos *et al.*, 2001]. ANNs have also been used in numerous other fields such

as the prediction of rehospitalization in patients suffering from strokes [Ottenbacher *et al.*, 2001], determining progression of glaucoma [Lin *et al.*, 2002], classification of bacterial growth [Hajmeer and Basheer, 2003], identifying factors which modify the responses of plant species to ozone [Balls *et al.*, 1996] and detecting the presence of fish species in rivers [Mastrorollo *et al.*, 1997].

In this study, a multi-layer perceptron ANN with a back propagation algorithm was used to model for 72 melanoma serum samples, 36 of which were low grade (stage 1) and the remaining 36 were high grade (stage 4). The purpose of the study was to identify any ions which were important in the correct classification of tumour grade and therefore may serve as potential biomarkers representative of a specific disease state. Techniques involved using relative importance values based on the weights of trained models to rank the importance of an inputs influence on the system [Balls *et al.*, 1996] and then removing inputs of low and no importance. This results in a more generalised model being developed, which enables the additive approach described by Ball *et al.* (2002) to be used in order to identify these important ions.

2. METHODS

2.1 SELDI-MS

Tissue preparation and SELDI mass spectrometry was carried out as described previously by Ball *et al.* (2002). In summary, two sequential 10-15 μm frozen tissue sections were cut for each tumour. The first section was stained with haematoxylin and eosin in order to determine tumour grade, purity and viability. The tissue was then placed directly onto 30 μl homogenising buffer (9.5 M urea, 3% CHAPS, 1 % DTT) for 15 min at room temperature with agitation to facilitate cell lysis and protein extraction. Homogenates were frozen at a temperature of $-80\text{ }^{\circ}\text{C}$ prior to SELDI analysis. Protein 'chips' were loaded with 2 μl of 50 % acetonitrile and 5 μl of cellular homogenate and exposed to the chip surface for 10 min at room temperature in a humid environment. Homogenates were then removed and the chip surface washed three times using 10 μl of water. The surface was then dried. Chip analysis was conducted at maximum laser intensity and 'phenomic fingerprints' derived from each tumour sample (Ball *et al.*, 2002).

2.2 Optimisation Of ANN Architecture

The study used a multi-layer perceptron ANN with a back propagation algorithm and a sigmoidal transfer function [Rumelhart and McClelland,

1986]. The particular architecture used contained three layers, with the hidden layer containing 2 hidden nodes. Determining the number of hidden nodes is essentially a trial and error procedure and 2 were found to give the best performance for this particular data (results not shown this paper). Architectures with learning rate and momentum values between 0.1 and 0.9 were trained in order to deduce the ANN model which performed best for this data. Using the mean squared error (MSE) value as a means of measuring prediction accuracy, it was found that a learning rate of 0.1 with a momentum value of 0.5 produced the lowest value. Training was conducted upon 60% of samples, with 20% being used for test sets, and the remaining 20% used for production (validation) sets until the model reached convergence. During training, the ANN model is optimised against the test set, and then validated against the production (validation) set. Convergence was determined by a failure of the model to improve the minimum MSE on the test data for 20,000 training events.

2.3. Determination Of Important Molecular Ions

The pattern recognition process involves several distinct phases which are (i) data representation, involving the initial determination of whether the tumour grade of the tissue was stage 1 or stage 4, (ii) feature extraction, where the analysis of the weights occurs, (iii) classification, where the ANN model assigns the data into either low or high grade classes and (iv) validation, where the ANN model is tested against unseen global data.

To determine which ions had the most influence on the system in correctly predicting tumour grade, the data was first screened for noise removal. To achieve this, data within the mass range of 2-5 KDa was trained over 50 random training/test/production subsets (so that a good level of confidence could be gained) and relative importance values for each individual ion was recorded in order to rank these ions according to their influence upon predicting tumour grade. These relative importance values are calculated from the analysis of weights of the trained network, values are calculated by taking the sum of the absolute weight values leading from each input to the output.

The data was then "shifted" up 500 Da so that the input data now ranged from 2.5-5.5 KDa and then trained as above. This process was repeated and the inputs were shifted over the whole data range (up to 30 KDa) which provided a proteomic profile showing relative importance of ions over the whole data range from 2-30 KDa taking account of potential interactions between ions.

From this relative importance analysis, ions with the greatest importance were selected from the data in order to reduce the number of inputs in the model. This was achieved by selecting the top 1,000 ions with the greatest relative importance values and repeating the training process as described previously. Relative importance analysis was again used to determine the top 500 ions from these 1,000. This was again repeated to deduce the top 300, 200, 100, 50, 30 then finally the top 20 ions from the initial data set of approximately 20,000 in terms of relative importance.

The next stage involved identifying the minimum number of ions from these 20 which were capable of correctly predicting tumour grade. This was achieved using an additive approach which involves training a number of different models. Using these 20 ions, each ion was used as a single input in predicting tumour grade, and for each model, 100 random training/test/production subsets were used (a process termed bootstrapping), in order to provide a measure of confidence in the predictions made. The MSE was calculated, and the ion model with the lowest value was selected for further training. All of the remaining ions were then added sequentially to this first input creating 19 two-ion models and these were trained as before with 100 random training/test/production subsets. The model with the best performance was selected to produce a three-ion model, and then the process was repeated and the ion with the best performance was again selected to produce a four-ion model.

3. RESULTS

The data obtained from SELDI-MS were analysed for relative importance values to create a relative importance profile for all data points with a m/z value of between 2 and 30,000 Daltons. Figure 1a-c shows the mean relative importance values from the 50 sub-models which were applied to each individual model.

The next stage involved selecting the ions which were most important in predicting tumour grade in order to optimise the model. This was achieved by ranking the ions in descending order of importance and selecting the top 1,000 ions. The training procedure was repeated and the top 500 ions were selected. This was repeated again so that the top 300 ions were selected and so on until a model containing the top 20 ions of importance was found. The purpose of this was to reduce the number of ions from an initial value of approximately 20,000 molecules to just the 20 that could predict tumour grade most accurately.

In order to identify the minimum number of ions which were able to accurately distinguish between

low and high grade tumours, an additive approach was used (as described in the previous section). This involved creating several models and assessing their performance with respect to the MSE value generated. Preliminary results have shown that the lowest MSE value obtained from the one-ion model was from an ion with a molecular mass of 7247 achieving a MSE of 0.235. Using a two-ion model, the error decreased to 0.205 using ions 7247 and 27867. With a three-ion model, containing the ions from the two-ion model and ion 4562, the error decreased further to 0.188. Finally, with a four ion model containing the addition of ion

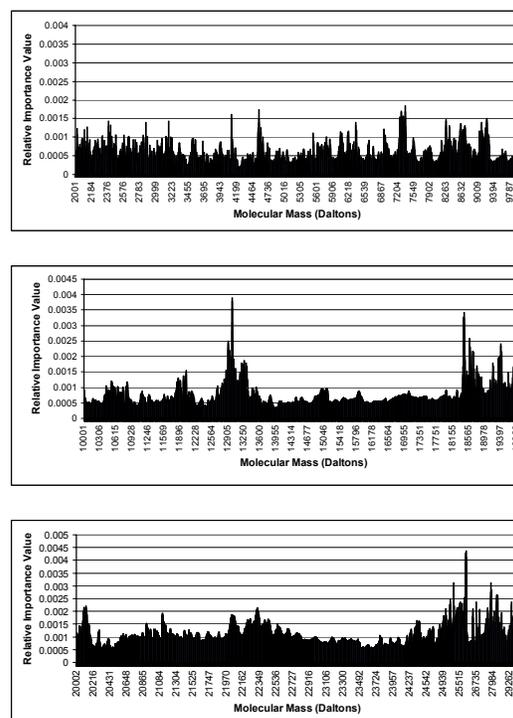


Figure 1a-c. Relative importance values for ion masses ranging from (a) 2,000-9,999 Da. (b) 10,000-19,999 Da and (c) 20,000-29,994 Da. These values illustrate a value obtained from 50 sub-models of each individual model in which different random weightings were applied to each model.

28470, the MSE value decreased further still to 0.16. Model performance from these preliminary results was then assessed using a Receiver Operating Characteristic (ROC) curve. A ROC curve determines the number of true positives (or correctly defined stage 4 melanoma tissue), true negatives (correctly defined stage 1 tissue), false positives (incorrectly defined stage 4 tissue) and false negatives (incorrectly defined stage 1 tissue). It achieves this by plotting the true positive rate against the false positive rate at different possible cutpoints (in this case, prediction errors). The curves for the one to four-ion models can be seen in Figure 2. The ROC curves for all models were compared and the results are presented in Table 1.

It is clear from Table 1 that a ROC curve provides information about several different variables. Briefly, accuracy is the overall ability of the model

Ions in model	Accuracy (%)	Sensitivity (%)	Specificity (%)	Positive Predictive Value (%)	Negative Predictive Value (%)	AUC
1 ion	58.3	63.9	52.8	57.5	59.4	0.574
2 ion	70.8	66.7	75	72.7	69.2	0.748
3 ion	77.8	72.2	83.3	81.3	75	0.809
4 ion	83.3	77.8	88.9	87.5	80	0.854

Table 1. Comparison of performances for each model used

to correctly assign the tissue samples. The sensitivity is the percentage of the stage 4 tissues correctly classified whilst the specificity is the percentage of stage 1 tissues correctly classified. The positive predictive value shows the percentage of the true positives distinguished from the false positives and the negative predictive value is the percentage of true negatives from false negatives.

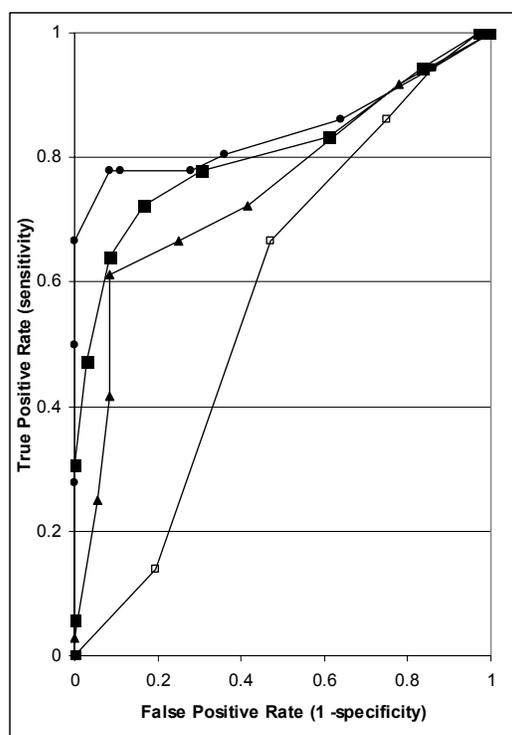


Figure 2. Diagram of ROC curves for all models. (◻) shows ROC curve for one-ion model. (▲) shows ROC curve for two-ion model. (■) shows ROC curve for three-ion model. (●) shows ROC curve for four-ion model.

The area under the curve, or AUC measures discrimination, that is, the ability of the model to correctly classify those with stage 4 and stage 1 disease. A perfect ROC curve (and therefore a

perfect test) has an AUC (area under the curve) value of 1, so the closer the curve follows the left hand border and then the top border of the ROC space, the more accurate the test. From the results it is clear that with increasing ions the accuracy of the model also increases. The one-ion model correctly classified 42 out of 72 tissue samples (58.3 %), with a two-ion model 51 out of 72 (70.8 %) samples were correct, using three ions, 56 of the 72 samples were assigned correctly (77.8%) and with the four-ion model, accuracy rose to 83.3 %, with 60 out of 72 samples being predicted correctly. The sensitivity and specificity of the models also showed a similar increase in performance as the accuracy. The sensitivity rose from 63.9% with the one-ion model, to 66.7 % with two-ions, a further increase was evident with the three-ion model to 72.2 % and with four-ions this rose again to 83.3 %. Meanwhile, specificity with one-ion was 52.8 % and increased by over 20 % when a second ion was added to the model (75 %). The three-ion model resulted in a specificity of 83.3 % which improved to 88.9 % for the four-ion model. The positive and negative predictive values also showed similar trends, rising from 57.5 % and 59.4 % with the one-ion model, to 87.5 % and 80 % with the four-ion model respectively. When assessing the AUC values, the one-ion model produced an AUC of 0.574 signifying a poor test. The two-ion model had an AUC of 0.748 which represents a fair test. The three and four-ion models had AUC values of 0.809 and 0.845 respectively illustrating good tests.

4. DISCUSSION

Results shown are those of preliminary work being carried out with the aim of identifying biomarkers capable of accurately predicting tumour grade from SELDI-MS data and therefore may be important in either developing novel therapies or in early detection of disease. The first stage of this study was to identify the top 20 ions (out of an initial 20,000) which were capable of predicting tumour grade. The next stage involved using an additive approach in order to determine and identify the minimum number of key ions that were capable of predicting tumour grade and thus may be important in tumour progression from low to high grade. ROC curves were generated for each model and these showed the increase in performance over the four models. The ion with the lowest error was identified as ion 7247 which predicted tumour grade with an accuracy of 58.3 %. The two-ion model contained ions 7247 and 27867 which predicted correctly 70.8 % of the time. The three-ion model consisted of ions 7247, 27867 and 4562 and classified the tissues correctly to a value of 77.8 %. Finally, the four-ion model contained ions 7247, 27867, 4562 and 28470 and performed with an accuracy of 83.3 %.

5. CONCLUSION

In conclusion, these preliminary findings show that when combining two powerful tools in SELDI-MS and ANNs, we can optimise the models so that the inputs with little or no influence upon the system may be removed in order to find the essential ions involved in the prediction of tumour grade. Although the method may be limited because the data set was relatively small due to the difficulties in obtaining tissue samples (these data sets will be expanded in future studies), the high classification accuracy of the models on truly unseen data shows that the models have generalised well enough to overcome this. Further research is ongoing and work continues on developing the models in order to conclude which, and how many ions the optimal model for the classification of tumour types from tissue contains. Once this is achieved, the next phase will involve methods for the analysis of interactions between ions within the system. Once these essential ions are identified they may be sequenced to determine their corresponding molecule/protein, which is essential in order to establish diagnostic markers.

6. REFERENCES

- Abbass, H.A. 2002. "An evolutionary neural networks approach for breast cancer diagnosis". In *Artificial Intelligence in Medicine.*, 25. Pp265-281.
- Almeida J.S. 2002. "Predictive non-linear modelling of complex data by artificial neural networks". In *Current Opinions in Biotechnology.* 13. Pp72-76.
- Ball G. Mian S. Holding F. Allibone R.O. Lowe J. Ali S. Li G. McCardle S. Ellis I.O. Creaser C. and Rees R.C. 2002. "An integrated approach utilizing artificial neural networks and SELDI mass spectrometry for the classification of human tumours and rapid identification of potential biomarkers". In *Bioinformatics.* 18 (3). Pp395-404.
- Balls G.R. Palmer-Brown D. and Sanders G.E. 1996. "Towards unravelling the complex interactions between microclimate ozone dose and ozone injury in clover". In *Water, Air and Soil Pollution.* 85. Pp1467-1472.
- Batuello J, Gamito E, Crawford D. Han M. Partin A.W. McLeod D.G. and O'Donnell C. 2001. "Artificial neural network model for the assessment of lymph node spread in patients with clinically localized prostate cancer". *Urology.* 57 (3). 481-85.
- Hajmeer M. and Basheer I. 2003. "Comparison of logistic regression and neural network-based classifiers for bacterial growth". In *Food Microbiology.* 20. Pp45-55.
- Lin A. Hoffman D. Gaasterland D.E. and Caprioli J. 2003. "Neural networks to identify Glaucomatous visual field progression". *American Journal Ophthalmology.* 135 (1). Pp49-54.
- Mango L.J. 1998. "Neural network-assisted cervical cancer screening". In *Journal of clinical ligand assay.* 21 (2). Pp203-207.
- Markopoulos C. Kouskos E. Koufopoulos K. Kyriajou V. and Gogas J. 2001. "Use of artificial neural networks (computer analysis) in the diagnosis of microcalcifications on mammography". In *European Journal of Radiology.* 39, Pp60-65.
- Mastrorillo S. Lek S. Dauba F. and Belaud A. 1997. "The use of artificial neural networks to predict the presence of small-bodied fish in a river". In *Freshwater Biology.* 38. Pp237-246.
- Ottbacher K.J. Smith P.M. Illig S.B. Linn R.T. Fielder R.C. and Granger C.V. 2001. "Comparison of logistic regression and neural networks to predict rehospitalization in patients with stroke". In *Journal of Clinical Epidemiology* 54. Pp1159-1165.
- Petricoin E.F. Ardekani A.M. Hitt B.A. Levine P.J. Fusaro V.A. Steinberg S.M. Mills G.B. Simone C. Fishman D.A. Kohn E.C. and Liotta L.A. 2002. "Use of proteomic patterns in serum to identify ovarian cancer". In *The Lancet.* 359. Pp.572-577.
- Porter C.R. O'Donnell C. Crawford D.E. Gamito E.J. Sentizimary B. De Rosalia A. and Tewari A. 2002. "Predicting the outcome of prostate biopsy in a racially diverse population: A prospective study". In *Urology.* 60 (5). Pp831-835.
- Rumelhart D.E. and McClelland J.L. 1986. "Parallel distribution processing: Explorations in the microstructure of cognition". Volume 1: Foundations. MIT Press. Cambridge, MA, USA.
- Yates J.R. 1998. "Mass spectrometry and the age of the proteome". In *Journal of Mass Spectrometry.* 33. Pp1-19.
- Simpson H.W. McArdle C. Pauson A.W. Hume P. Turkes A. and Griffiths K. 1995. "A non-invasive test for the pre-cancerous breast". In *European Journal of Cancer.* 31A (11). Pp1768-1772.
- Zhou Z. Jiang Y. Yang Y. and Chen S. 2002. "Lung cancer cell identification based on artificial neural network ensembles". In *Artificial Intelligence in Medicine.* 24. Pp25-31.

Computer Mediated Communication and Organizational Culture: An Agent-Based Simulation Model

Enrique Canessa
Universidad "Adolfo Ibañez"
Faculty of Science and Technology
Balmaceda 1620, Viña del Mar, Chile
E-mail: ecanessa@uai.cl

Rick L. Riolo
University of Michigan
Center for the Study of Complex Systems
4477 Randall Lab., Ann Arbor, MI 48109-1120, USA
E-mail: rriolo@umich.edu

KEYWORDS

Agent-based modeling, organizational communication, organizational culture.

ABSTRACT

This paper examines the mutual relationship between the organizational use of Computer Mediated Communication (CMC) and organizational culture (OC). CMC supplements communication among members of an organization to maintain the culture, especially when those persons cannot communicate by other means. On the other hand, a strong OC allows a more effective use of CMC by providing members with some of the necessary common ground to better understand the information exchanged. These relationships are investigated using an agent-based model (ABM). Our ABM incorporates many partial theories into a coherent and fully defined model, which helps formalize and integrate those theories. Although we have empirically validated the ABM, our model allows us to go beyond what can easily be done using empirical research, such as analyzing non-linearities and interaction effects. Additionally, the ABM allows us to investigate dynamics and generate hypotheses that could then be tested using empirical studies. In this paper, we present some of the results of the ABM that show that OC can influence the effectiveness of CMC and that CMC can help maintain and stabilize a culture.

INTRODUCTION

Computer Mediated Communication (CMC) allows two or more persons who are not physically together to exchange information through a computer system. Many studies have suggested that CMC has the potential to provide tools for enhancing the flow of information in an organization (Fulk and DeSanctis, 1995). However, research aimed at analyzing the effective use of CMC in organizations has arrived to contradictory conclusions. Positivist studies of CMC based on the Information Richness Theory (IRT) (Daft and Lengel, 1986) have found that CMC is inadequate to handle ambiguous situations. On the other hand, interpretivist studies of CMC have shown that CMC can accommodate the exchange of information even in confusing situations.

For IRT the communication richness (CR) of a medium explains why this medium is more or less effective. CR

refers to the ability of a communication system to transfer enough cues so that individuals can reach an understanding within a short time. For IRT, face-to-face communication is the richest media because it provides immediate feedback and allows the exchange of multiple cues through body language and tone of voice. Since CMC restricts the use of immediate feedback and/or the exchange of multiple cues, IRT views CMC as inherently a medium of low richness.

On the other hand, interpretivist studies have shown that organizational members can use CMC (e-mail) to effectively communicate under ambiguous conditions (Lee, 1994; Ngwenyama and Lee, 1997). These studies claim that the richness of any communication medium changes according to the organizational context in which it is used. The person who sends a message and the one who receives it are part of an organizational context, so they not only derive the meaning of the message from the information it provides, but also interpret it taking into account other information they have at their disposal, such as knowledge of the other person, of the situation at hand and of the organization.

As one can see, IRT-based studies focus mainly on the intrinsic characteristics of the communication medium and analyze them independently of individual and organizational context. For the interpretivist studies, the attributes of a communication medium are dependent on both the intrinsic and extrinsic characteristics of the medium. Those extrinsic characteristics originate from the individuals who use it and the organizational context.

HYPOTHESES

One way to succinctly incorporate organizational context into the analysis of CMC or any other communication system is in terms of OC (Zack and McKenney, 1995). One definition of OC states that it is "a pattern of basic assumptions, invented, discovered or developed by a given group, as it learns to cope with its problems of external adaptation and internal integration, that has worked well enough to be considered valid and therefore is to be taught to new members as the correct way to perceive, think, and feel in relation to those problems" (Schein, 1990). This definition of OC suggests that OC will contribute to enhancing the possibility of reaching a mutual understanding when members of the organization communicate. Common assumptions tend to homogenize how members handle their work-related problems, by contributing to a common

understanding, which will facilitate communication, especially when using low to medium richness media (Clark, 1996). This beneficial effect of OC will depend on how widespread and strongly members hold the assumptions embedded in the culture. A variable that represents this attribute of OC is its strength (Denison, 1990). These points can be summarized in hypothesis **H1**: *The stronger the OC, the higher the CR of the communication system*. Since that hypothesis and the ones below are applicable to any communication system used in an organization all the propositions are stated using the generic term communication system.

Note that if the initial strength of the OC is high, then members of the organization will have somewhat similar values, beliefs and assumptions. That common ground provided by a strong culture will facilitate the communication process. Thus, it will take the members a shorter time to reach a consensus than if the initial culture is weak, leading to hypothesis **H2**: *The stronger the initial OC, the faster the culture will stabilize*.

The previous hypotheses stated possible relationships between culture and CR. From a practical point of view, it is also interesting to see whether the mutually beneficial effects of a strong OC and high CR might be reflected in the effectiveness of the organization. Some case-based and empirical studies have suggested that a strong OC can enhance the performance of an organization (Denison, 1990 and references contained in that book). The main argument is that a strong culture establishes a common ground that facilitates the work among employees. Those beneficial effects might be reflected in the shortening of the time required to complete tasks. **H3**: *The stronger the OC, the shorter the task-completion time*.

Note that although these hypotheses plausibly follow from theory, because of non-linear interactions among variables and the general complexity of the phenomenon, it is not possible to determine a-priori that a particular formal model would generate results that would support those propositions.

THE MODEL

The model implements the conceptual ideas and mechanisms from the theory outlined in the introduction that bear on the hypotheses described in the previous section. There are many ways to implement those basic ideas. The present model is an attempt at a fairly simple implementation that captures the key mechanisms believed to be the most relevant. The following paragraphs describe the details of the model that are important for understanding the experiments carried out to test the hypotheses. For full details of the model, and for additional hypotheses and experiments see Canessa, 2002.

Organizational and Communicational Structure

The model assumes that an organization is a collection of groups of people that pursue some common goals. To reflect the relative difference in individual power in a firm, each agent has a number that represents its status. Members

of a group can freely communicate. In the case of inter-group communication, only some members of a group may directly communicate with members of other groups. The capability of members to communicate outside their groups might influence their status. Since members who have a broader communication network have more influence, the status of the agents that can communicate outside their own group will be higher than the status of those who cannot (Krackhardt and Hanson, 1993).

Task Assignment and Completion

The organization assigns to each agent a task to complete. Each task consists of a given number of contacts that the agent must make with other members in order to complete the task. Some steps are sequential--the agent must wait until it receives a reply before advancing to the next step; other steps are non-sequential. Sequential task steps occur with probability 0.5.

If an agent is authorized to communicate outside its group, with probability 0.6 the organization assigns to it steps that involve contacting agents in other groups; otherwise, that probability is 0.3. When a task that requires inter-group communication is assigned to an agent that is not authorized to communicate outside its group, that agent must relay inter-group messages through the agents that are authorized to communicate outside the group.

Each time an agent completes a task the organization assigns a new one to it. With probability 0.5 the organization changes the identity of the agents involved in completing the new task and/or the sequence in which the contacts must be made. The rules that agents observe when completing their assigned tasks are:

- An agent engages in completing one task at a time.
- The number of steps of a task that an agent can perform in a simulation step is equal to the number of messages an agent can answer.
- An agent processes messages that belong to its own task first. After that, if the number of already processed messages is smaller than the maximum the agent can process, then the agent processes messages from other agents. The order in which the agent processes those messages is dictated by the status of the senders, so that messages from high status agents are processed first.

Organizational Culture and Communication Effectiveness:

OC is represented as a list of dimensions (Axelrod, 1997). For this study the number of cultural dimensions was ten. The initial values for each dimension are sampled from a normal distribution with mean zero and a given variance. This variance defines the starting variability of the OC and thus, the corresponding initial OC strength. The larger the variance, the weaker the initial culture.

Communication effectiveness (CE) is defined as the probability that two agents can communicate without problems. CE is a function of the difference in culture

between two agents, based on the sum of the absolute value of the differences in values between corresponding dimensions for the two agents. A sigmoid curve is used to calculate CE:

$$CE_{jk} = \frac{1}{1 + e^{\left(\sum_{i=1}^N |T_{ij} - T_{ik}|\right)^{\alpha - \beta}} \quad (1)$$

where T_{ij} is the i th dimension of the culture of agent "j" and T_{ik} is the i th dimension for agent "k" and N equals the number of cultural dimensions. The constants α and β adjust the shape of the sigmoid curve. In this study, α was set to 0.25 and β to 5.0. The value of CE between two agents specifies the probability that the receiver of a message understands it. If the receiver understands the message, then it processes the message. If the receiver does not understand it, then the receiver replies with a clarification message. The sender of the first message responds to this clarification. Upon receiving the answer to the clarification from the sender, the receiver decides if it now understands the new message. This process continues until the receiver understands the message or the receiver or sender quits sending/answering clarifications. The receiver or sender quits sending/responding clarifications when the number of clarifications exceeds three. If the sender quits answering clarifications or the receiver notifies the sender that it quit sending clarification messages, then the sender selects a new receiver for the message. This change of receiver occurs only once. If after changing receiver, the message is still not understood, then the communication fails, and the organization discontinues the corresponding task and assigns a new task to the agent.

Communication Richness and Organizational Culture Change

Different communication channels exhibit varying capacities for transmitting different types of cues. Therefore, when agents communicate using a channel, this channel will allow them to transfer some ("visible dimensions") and will block the transfer of other dimensions. The visible dimensions will be the ones that can change during the simulation. The model allows establishing the number of visible dimensions for communications among agents that belong to the same group (intra-group communication) and for communications among agents that belong to different groups (inter-group communication). The reason for distinguishing between the richness of intra and inter-group channels is that the members who belong to the same group will have more opportunities to communicate through rich channels (for example face-to-face meetings) than members who belong to different groups (Olson and Olson, 2000).

The OC change between agents takes place every time two agents communicate. The message receiver will change its culture toward that of the sender in an amount proportional to the CE and the difference in status between them. When agents s (sender) and r (receiver) communicate, r 's culture will change according to:

\forall Visible Dimensions "i" between agent "s" and "r": (2)

$$T_{t,i,r} = T_{t-1,i,r} + CE_{sr} (T_{t-1,i,s} - T_{t-1,i,r}) \frac{Status_s}{Status_s + Status_r}$$

where $T_{t,i,r}$ is the value of dimension i at time t for r . The quotient of the statuses represents the asymmetrical nature of the influence that persons of different status can exert on each other (Salancik and Pfeffer, 1974). The bigger the difference in status between two persons, the higher the influence the person of higher status can exert on the person of lower status and, and vice versa.

The effect of CE reflects the influence a person might have on the culture of another if they can understand each other (Axelrod, 1997). Note that the change in OC is unidirectional; that is the sender influences the culture of the receiver and not vice versa. Since the receiver acts as sender when responding to the message and the original sender acts as receiver, the effect becomes bi-directional but not synchronous.

Sequencing of Events and Updating of the Model

The simulation is updated asynchronously (to avoid artifacts---cf. Huberman and Glance, 1993), as follows:

- a) Select at random without replacement agent A.
- b) Allow A to send messages for its current task.
- c) Process incoming messages for A and change its OC.
- d) See whether A's task is complete. If the task is complete:
 - i. Compute measures pertaining to the task.
 - ii. Assign a new task to A.
- e) Repeat steps a) through d) for all agents.
- f) Compute the measures and outputs of the model.
- g) Repeat a) through f) for as many steps as specified.

Measures and Outputs of the Model

The following measures are used in this paper:

- a) Average task-completion time for the organization (ATCTO). For all the completed tasks calculate the time it took to finish those tasks. Calculate an average time for the entire organization. This time reflects only the time agents spend communicating, so it applies only to assessing how long it takes members to carry out the communicational part of their jobs.
- b) Overall OC strength (OOCS) measures the strength of the OC by calculating the variance for each of the dimensions of the culture for the entire organization, combining them using:

$$OOCS = \frac{1}{1 + \sum_{i=1}^N \sigma_i^2} \quad (3)$$

where σ_i^2 is the variance of cultural dimension i and N (=10) is the number of cultural dimensions. Note that the stronger the culture, the smaller the variation and thus the closer OOCS will be to one.

- c) Organizational average culture (OAC) is the average value of the culture computed over all cultural dimensions and agents. The time series corresponding to

OAC will reflect the dynamics of the culture of the organization. When this time series remains unchanged, the system is in equilibrium.

- d) Average communication effectiveness for completing tasks for the entire organization (ACETO). Calculate the CE for each assigned task, even if it was not completed. The average CE for a task is calculated as the geometric mean of all the CE's between senders and receivers. For example, if agent 1 needs to communicate with agent 4 and to do so needs to go through agents 2 and 3 for completing the task, then:

$$CE_{\text{task}} = (CE_{12} CE_{23} CE_{34} CE_{43} CE_{32} CE_{21})^{1/6}$$

where CE_{12} = CE between agent 1 and 2, CE_{23} = CE between agent 2 and 3, and so on. Using the CE_{task} of all the assigned tasks, compute the average, which corresponds to ACETO. This measure reflects how well agents are communicating due to the intrinsic and extrinsic CR of the medium. If the intrinsic richness is high (i.e. the communication channel allows the transfer of many cultural dimensions), the culture is able to homogenize well and that increases the similarity among the agents' cultures. If the extrinsic richness is high (i.e. the agents' culture is already similar), the difference between the cultures of agents is low. In both cases, the CE_{task} will be high (close to one) and, correspondingly ACETO will also be high.

RESULTS

Before running experiments, we carried out extensive verification and validation of the program. Details are in Canessa, 2002. Table 1 shows the parameter values used in the experimental runs.

Table 1: Combination of Parameters Changed for Experimental Runs

Parameter	Value	Experimental condition label
CIR	6 visible cultural dimensions within group	Low intrinsic communication richness (LICR)
	4 visible cultural dimensions between groups	
	10 visible cultural dimensions within group	High intrinsic communication richness (HICR)
	8 visible cultural dimensions between groups	
IOCS	Variance of normal distribution = 5	Strong initial culture (SIC)
	Variance of normal distribution = 10	Weak initial culture (WIC)

CIR = Communication channel intrinsic richness
IOCS = Initial organizational culture strength

We used four different numbers of steps per task (10, 20, 30 and 40), which were matched up with the corresponding number of messages an agent could process per time step; e.g., for a 40-step task, each agent has the capacity to process 40 messages per step. These four pairs of values were combined with the two scenarios for CR and for initial strength of OC, for a total of sixteen combinations. Each of these combinations was simulated for 600 time steps and

replicated thirty times using different RNG seeds. Other parameters kept fixed were: the organization had 8 groups, each of 30 agents; three agents of each group were authorized to communicate directly with agents of other groups; these agents had a status of two, whereas the rest had a status of one.

Hypotheses Testing

The results for the 10, 20, 30 and 40-step tasks are very similar and thus we will report the outcomes for the 40-step tasks only. Tables 2 and 3 show the data gathered from the experimental runs, which we will use in testing the hypotheses and making other analyses. Specifically, Table 2 presents the OC strength (OOCs) and its standard deviation computed over the thirty replications using the last sixty data points of each run, where the system was in equilibrium.

Table 2: Organizational Culture Strength for the 40-step task

	SIC	WIC
LICR	0.0472 (0.0018)	0.0239 (0.0009)
HICR	0.7311 (0.0695)	0.5215 (0.0914)

(mean over the last 60 data points, std. deviation in parentheses, N = 30 replications)

Similarly, Table 3 presents the overall organizational CR and its standard deviation computed under the same conditions. Note that this overall organizational CR corresponds to the average communication effectiveness (ACETO), which encompasses both the intrinsic richness that does not change (due to the established number of visible cultural dimensions) and the extrinsic richness, which changes (because the culture of agents becomes more similar as time advances).

Table 3: Overall Organizational Communication Richness for the 40-step task

	SIC	WIC
LICR	0.8476 (0.018)	0.5792 (0.026)
HICR	0.9889 (0.0002)	0.9799 (0.0008)

(mean over the last 60 data points, std. deviation in parentheses, N = 30 replications)

If hypothesis H1 is true, we expect to see that the higher the value of OOCs in Table 2, the higher the corresponding value of ACETO in Table 3, which is the case. Table 4 shows the differences in ACETO, all of them significant. Thus, we conclude that H1 is supported.

Table 4: Differences in Means of Communication Richness corresponding to the four different values of organizational culture strength for the 40-step task

	Final Culture Strength	Comm. Richness	Difference in Comm. Rich.
Strongest final culture	0.7311	0.9889	
Moderately strong final culture	0.5215	0.9799	0.010
Weak final culture	0.0472	0.8476	0.1323
Weakest final culture	0.0239	0.5792	0.2684

(all differences significant at least at the 0.01 level)
For example: 0.010 = 0.9889 - 0.9799 and so on

To test hypothesis H2 we compute the steps required for the mean of the organizational average culture (OAC) to reach equilibrium (defined as starting when the time series of organizational average culture remained unchanged). Table 5 presents these figures.

Table 5: Mean Stabilization Time for Organizational Culture for the 40-step task

	SIC	WIC
LICR	20.5 (4.93)	34.3 (11.27)
HICR	18.0 (4.48)	19.3 (3.92)

(standard deviation in parentheses, N = 30)

If hypothesis H2 is true, we expect that the differences between the times corresponding to an initially weak and strong culture should be positive and significant.

Table 6: Differences in Stabilization Time of Organizational Culture between initially strong and weak cultures

	LICR	HICR
40-step task	13.80 (<< 0.001)	1.30 (0.237)

(p-values in parentheses)

The figures correspond to the difference in stabilization time between an initially weak and strong culture: stabilization time for initially weak culture - stabilization time for initially strong culture: $34.3 - 20.5 = 13.8$ (see figures in Table 5)

From the figures of Table 6, one can see that for low intrinsic CR, hypothesis H2 is consistently supported (the difference is positive and statistically significant), whereas for high intrinsic CR it is not (difference is relatively small and non-significant). Note that the decrease in stabilization time between an initially weak and strong culture is much more pronounced for low intrinsic CR than for a high one. This happens because a low intrinsic CR prevents some cultural dimensions from changing. Thus, if these unchanged dimensions are initially similar, as when an initially strong culture exists, then the extrinsic CR among agents will be always higher than when these unchanged dimensions are initially dissimilar, as when an initially weak culture exists. Since extrinsic CR dictates how much the culture between agents will homogenize per step, the higher the extrinsic richness, the faster the culture will homogenize. Hence, the impact of an initially strong or weak culture on stabilization time of the culture will be higher when intrinsic CR is low than when it is high.

Table 7 presents the means and standard deviations of the task-completion times for the entire organization (ATCTO).

Table 7: Task-completion Time for the 40-step task

	SIC	WIC
LICR	23.054 (2.307)	61.406 (12.221)
HICR	16.894 (0.059)	16.947 (0.068)

(mean over the last 60 data points, std. deviation in parentheses, N = 30 replications)

If hypothesis H3 is true, then we should see that the task-completion times for strong final cultures (situations where OACS is high in Table 2) would be shorter than the ones for

relatively weak final cultures. To assess that, we computed the difference in task-completion time at equilibrium for strong and weak final cultures, using the times of Table 7. Table 8 presents these differences. All the differences between these times are statistically significant. One can see that the task-completion times are shorter for strong cultures than for weak ones. Thus, hypothesis H3 is supported.

Note that the impact of an initially strong or weak culture is more pronounced for low intrinsic CR than for high (see Table 7). This interaction effect of CR on the relationship between culture and task-completion time occurs because a low intrinsic CR prevents some of the cultural dimensions from homogenizing. Under that condition, the initial similarity of the dimensions that an initially strong culture produces is more important than when a high intrinsic CR exists. In this latter case, almost all of the cultural dimensions will homogenize and thus will decrease the impact of an initially weak culture on task-completion time at equilibrium.

Table 8: Differences in Means of Task-completion Times corresponding to the four different values of organizational culture strength for the 40-step task

	Final Cult. Strength	Task completion time	Difference in Task-completion Time
Strongest final culture	0.7311	16.894	
Moderately strong final culture	0.5215	16.947	-0.053
Weak final culture	0.0472	23.054	-6.107
Weakest final culture	0.0239	61.406	-38.352

(all differences significant at least at the 0.01 level)

For example: $-0.059 = 13.897 - 13.956$ and so on

In addition to allowing testing the postulated hypotheses, the runs showed another interesting aspect of the system's behavior. In one of the runs, task-completion time exhibited a different dynamic from the rest of the runs. In general, task-completion time increases at the beginning of the simulation reaching a maximum and then it begins to asymptotically decrease toward a lower equilibrium value. This happens because the first completed tasks among all the tasks that the organization assigns are the ones that take agents a shorter time to complete. Since those short tasks are the ones the model includes in the first calculations of the mean task-completion time, that figure remains low. As time advances, agents complete the more complicated tasks, which increases the mean task-completion time. However, at the same time, the OC begins to homogenize, making it easier for agents to understand each other. This shortens the task-completion times, which in turn, decreases the mean value of that variable. Finally, the culture homogenizes as much as the conditions allow and the system reaches equilibrium. At this stage, the task-completion time reaches its equilibrium value.

However, in one run, the dynamics of task-completion time changed. At the time when that variable was reaching its equilibrium value, suddenly it jumped to a higher value, interrupting its asymptotic decrease. After that abrupt variation, the dynamic of task-completion time went back to

normal, i.e., it began to decrease reaching an equilibrium value. Examining the run, we found that the organization had assigned tasks to agents involving almost no change in the identity and sequence of contacts, from the beginning of the run until the moment the change in dynamics occurred. At that moment, the organization (by chance) drastically changed the identity of the agents involved in each step of the tasks and somewhat the sequence of contacts. Examining the culture of the agents, we saw that because the tasks were initially so stable, the agents had fine-tuned their culture to accomplish such tasks, creating very strong local cultures. These local cultures significantly differed from one another. Thus, when the organization changed the contacts for completing the tasks, the agents had to communicate outside these local cultures. Since these cultures were strong but different, agents could not immediately adjust to their new communication partners. This caused an increase in task-completion time. Eventually, as the local cultures homogenized, that measure improved.

DISCUSSION

As one can see from the results of the experiments, in general the postulated hypotheses were supported. This is not surprising since the model embeds part of the corresponding theory that supports such hypotheses. However, the interaction effects discovered were not postulated a priori based on the theoretical background. Although a close examination of the model helped explain why these interaction effects occurred, our intuition regarding the outcomes of the model was not completely right. Thus, the agent-based model served the purpose of enhancing the understanding of the phenomenon under study. The usefulness of this approach in this study agrees with similar ones reported in other papers (Axelrod, 1997).

The new relationships discovered have some useful implications. First, the results showed that the difference in the stabilization time of a culture between high and low intrinsic CR media for an initially strong culture is small. On the other hand, for a weak initial culture, the stabilization time is significantly shorter for media of high richness than for low ones. This might suggest that the use of low richness media, such as CMC, is appropriate for stabilizing a culture when it is already strong. However, when the culture is weak, one should favor the use of high richness media. Second, the interaction effect of CR on the relationship between task-completion time and the initial strength of a culture suggests that a modest increase in the strength of the culture might significantly increase organizational performance. This conclusion is important for virtually collocated work, which involves persons geographically separated working on common tasks through CMC.

The major part of the work on CMC has been conducted using experiments and survey or field research. This study

took a different approach to analyzing the bi-directional link between the use of CMC (or any communication system) and OC. In addition to the results presented, the ABM described here contributes to the CMC research in two other ways. First, the model may be used in future studies to help pinpoint some questions to be answered and consequently design experiments, surveys or field studies. Since the latter approaches generally cannot be easily repeated, it is very useful to have a means of anticipating the possible areas to focus on, leading to better design of experiments, surveys or field work. Second, the translation of some social science theories related to CMC that have been stated in words to a very precise operationalization, as required in ABM, helps formalize the theories. This assists in enhancing the mutual understanding among researchers and the transfer and accumulation of knowledge in the field.

REFERENCES

- Axelrod, R. (1997). The dissemination of culture: A model with local convergence and global polarization. *Journal of Conflict Resolution*, Vol. 41, pp. 203-226.
- Canessa, Enrique (2002). *Computer Mediated Communication and Organizational Culture: A Survey-Based Study and Agent Based Simulation Model*. Doctoral Dissertation. Rackham Graduate School, University of Michigan.
- Clark, H. H. (1996). *Using language*. New York: Cambridge University Press.
- Daft, R.L. and R.H. Lengel (1986). Organizational Information Requirements, Media Richness and Structural Design. *Management Science*, Vol. 32, No. 5, 1986, pp. 554-571.
- Denison, Daniel (1990). *Corporate Culture and Organizational Effectiveness*. John Wiley & Sons: New York.
- Fulk, Janet and DeSanctis, Geraldine (1995). Electronic Communication and changing organizational forms. *Organization Science*, Vol. 6, No. 4, pp. 337-349.
- Huberman, Bernardo and Natalie Glance (1993). Evolutionary games and computer simulations. *Proceedings of the National Academy of Science, USA*, Vol. 90, pp. 7716-7718.
- Krackhardt, D. and J. Hanson (1993). Informal networks: The company behind the chart. *Harvard Business Review*, July-August 1993, Vol. 71, pp. 104-111.
- Lee, Allen (1994). Electronic mail as a medium for rich communication: an empirical investigation using hermeneutic interpretation. *MIS Quarterly*, June 1994, Vol. 18, pp. 143-157.
- Ngwenyama, Ojelanki and Allen Lee (1997). Communication Richness in Electronic Mail: Critical Social Theory and the Contextuality of Meaning. *MIS Quarterly*, June 1997, Vol. 21, pp. 145-167.
- Olson, Gary and Judith Olson (2000). Distance matters. *Human Computer Interaction*, Vol. 15, No. 2-3, pp. 139-178.
- Schein, E. (1990). Organizational culture. *American Psychologist*, Vol. 45, No. 2, pp. 109-119.
- Stinchcombe, Arthur (1990). *Information and organizations*. University of California Press Berkeley CA.
- Zack, Michael and James McKenney (1995). Social context and interaction in ongoing computer-supported management groups. *Organization Science*, Vol. 6, No. 4, pp. 394-422.

NOVEL NEURAL NETWORK METHODS FOR DESCRIBING ATTRIBUTES CONTAINED WITHIN LESIONS IMAGES

S.G.ANDREWS

QinetiQ
KISystems, Consulting
The Alba Centre, Alba Campus
LIVINGSTON
EH54 7EG
sandrews@QinetiQ.com
stuart@embedded-software.org.uk

Abstract – Several novel methods based on intelligent recognition techniques are presented. This is an extension of our earlier work that utilises a number of these techniques, which included production rules, genetic algorithms and associative memories. The first additions involve the use of an implied grammar that is derived from a dermatology heuristic that describes lesion morphology characteristics; this is also known by its mnemonic ‘ABCDE’. The technique is combined with both our novel neural adaptive architecture and an image transform, the HALOGRAPH. This combination extracts the requisite components from the lesion image defined by the grammar, from which subsequently a diagnosis is generated. Another aspect of the lesion morphology is also considered: texture. The technique applied here makes use of Laws basis function to describe its structure. The texture landscape is defined as an energy profile. One of the limitations of our previous system involved the creation of volumes of information during the analysis phase, and to limit this effect we now apply PCA. The inclusion of this technique provides a reduction in the information content while retaining the essential detail, and it also acts as a method of feature recognition. A further side effect of this approach is that it can be used as an encoder to assist the simplification process of the internal definition of the evolving neural architecture.

Keywords: Production rules, Evolving Neural architectures, Image Recognition, and Genetic Algorithms, Grammars.

1. INTRODUCTION

The aim of the research of which this paper is a subset was to overcome some of the limitations that standard unsupervised neural architecture behaviour exhibits, and to develop an intelligent assistant that could be used both by general practitioners, and specialist dermatologists without having to resort to lengthy training.

Neural networks are used with increasing frequency in the context of medicine. Inspiration for artificial neural networks derives from their biological analogy the neurone. The earliest example of the artificial neural networks was the 'Perceptron', [Rosenblatt, 1958]. More complex neural architectures such as the Back-propagation network (BNN) [Rumelhart et al, 1986; Werbos, 1974], have been used in the detection of cancer cells [Moallemi, 1991], and with skin cancer recognition [Stoecker et al, 1997; Tomatis et al; 1997; Burroni et al, 1997]; A different architecture, namely the radial basis function network (RBF) [Burroni et al, 1997] was used in conjunction

with spectroscopics for detection of cervical precancer [Tummer et al, 1998]. Other topics such as breast cancer screening have used neural networks [Bridgett et al, 1995]. A method of learning medical image shapes, and also in the context of image compression was applied by [Panagiotidis et al, 1996] using a probabilistic neural network (PNN). The topic of cranial pressure and monitoring of neuro-surgical patients was investigated [Swiercz et al, 1995, 1998] using a recurrent neural architecture (RNN) [Tomatis et al, 1977]. More recently neural networks have been used in capturing and defining MRI images [Reyes-Aldasori et al, 2000].

2. METHOD

In this paper we present a number of extensions and improvements to our novel neural network [Andrews et al, 1998, 1999, 2000]. The network is used to recognise skin cancer images. The architecture uses intelligent techniques, namely production rules [Nilsson, 1980], and the genetic algorithm (GA)

[Holland, 1975; Harp, et al, 1991; Goldberg, 1989]. The first of these is a method for describing the internal structure of the neural architecture. One of the main behavioural features is its ability to remember feature patterns encountered. The concept that we make use of is that of association. We make reference to the concept of *associativity*. This concept has been extensively researched by [Anderson, 1970; Kohonen, 1974, 1988]. Training is handled by use of the memory concept, which allows any pattern previously encountered to be recalled. Any pattern anomalies are rapidly assimilated into the memory, leading to the modification of the neural definition. The behaviour of the novel neural network architecture makes use of an associative memory (AM) to control the assembly of the lesion image analysis. Each time a feature (pixel) is encountered a reference is made to the AM as to whether there had been a prior encounter with this feature, if not a neuron is assigned. Any further encounter with the same feature causes the neural strength to be incremented. A further fact that is elicited is the geographic distribution. This is assigned both neurons and associated strength. This process continues until all features are processed. Genetic algorithms (GA) [Holland, 1975; Harp, et al, 1991; Goldberg, 1989] are used in place of Hebbian learning [Hebb, 1949], and the feature histogram is used to influence the behaviour of the GA in its production of the weight definition.

Two further methods are used, which define characteristics contained within the lesion image. The first of these uses an implied grammar [Friedmann et al, 1985]. The second involves the lesion surface texture in terms of an energy map [Laws, 1979]. The first of these methods operates in conjunction with the implied grammar and an image transform, the HALOGRAPH [Andrews et al, 1998]. Other artificial intelligence techniques involved include information reduction and encoding (PCA) [Jolliffe, 1986; Sanger, 1989] and these operate with the novel neural network [Andrews et al, 1989, 2000].

The PCA is a powerful data analysis tool that is used in the context of multivariate analysis [Amari, 1977]. The technique used in Sanger's GHA is a generalisation in terms of the standard approach to PCA. One of the limitations of PCA which is that it makes various assumptions (e.g Gaussian Distribution) which as a result causes it not to characterise all the trends within the data. The transform simplifies the extraction and definition of the image details while simultaneously adding rigour to the diagnosis process thereby avoiding ambiguity.

The topic of surface texture is an important diagnostic tool [Stoecker et al, 1992, 1997]. Laws basis functions are used [Laws, 1979] to describe the details. These are used to differentiate the various micro-features of the surface texture that are present within the image. Having elicited the energy profile using this technique we then simplify the resulting product using a combination of PCA [Jolliffe, 1986] and neural network [Sanger, 1989]. The resultant definition is in effect a précis of the energy mapping. Using this approach allows for both recall and reconstruction of the précis. A further side effect of using PCA is the ability to be able to generalise. For instance, another image can be reconstructed using an already available PC definition [Cichocki et al, 1994, 1996]. A more comprehensive definition of the energy profile is constructed using the novel neural architecture [Andrews et al, 1998, 2000].

The reason for using Laws is that it enables the characterisation of micro-features that are contained within the lesion image. It also provides a method of extracting a scaled variance profile with regard to each type of micro-feature.

3. APPLICATION

The following application illustrates how the novel neural architecture [Andrews et al, 1998, 2000] acts in combination with associated neural architectures [Oja, 1982; Sanger, 1989] to describe facets of lesion morphology. The first of these processes was explained earlier. This defines the lesion surface texture in terms of an energy profile. Using the energy profile [Laws, 1979] allows several levels of definition to be described. The resulting details are extracted and encoded using PCAGHA [Sanger, 1989].

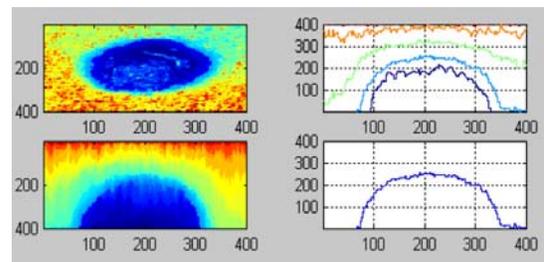


Figure 1 Lesion images & HALOGRAPH transformation of image.

This method allows for rapid recall of the original information. Modifications of the behaviour of the architecture [Andrews et al, 1998, 2000] are effected using the same process [Sanger, 1989] in which the

detail [Nilsson, 1980] is simplified (see fig 1). This is applied at each level of the definition decomposition.

The above definition illustrates how the implied grammar (mnemonic, 'ABCDE') and describes the lesion morphology in terms of its geometric profile, boundary condition, and pigmentation distribution. Using standard imaging techniques this decomposition becomes extremely complex.

Figure 2 shows how the grammar unfolds the aspects of the lesion morphology and then uses the results to assemble a diagnosis. The grammar forms a partnership with the novel image transform [Andrews et al, 1998] and three other neural architectures [Andrews et al, 1998; Nilsson, 1980; Sanger, 1989].

R1: if σ_1 & σ_2	then 'A'
R2: if $\beta\chi_1$ & $\beta\chi_2$	then 'B'
R3: if $\chi\delta_1$ & $\chi\delta_2, \dots, \chi\delta_n$	then 'C'

Figure 2 Example rules describing initial level of the implied grammar.

The grammar acts in a supervisory role to extract and define salient aspects of each lesion image in relation to each aspect of the heuristic. The neural networks [Oja, 1982; Sanger, 1989] are used to encode firstly the lesions boundary contour, and secondly the pigmentation distribution.

4. RESULTS

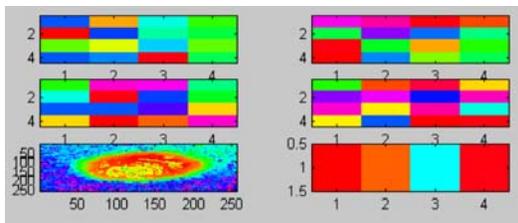


Figure 3 PCA definition of a lesion image

The above diagram, figure 3 shows how by applying PCA techniques based on Sangers GHA neural architecture [Sanger, 1989] the complexity of the information content can be significantly reduced. Without using the above approach the level of description is large. We mentioned earlier that the neural assembler decomposes the lesion image into a hierarchy that forms a binary-tree. Each level of the hierarchy has its own definitions. The initial level contains significant detail. Each subsequent level contains less and less detail (see figure 5 & figure 6).

The maximum level of decomposition is five after which there is little or no relevant information present. To illustrate the process involved we shall transform figure 1 into symbolic form.

In figure 4 we have four masks having 16 coefficients, which subsequently reduces to four PC's. By comparison with the table shown in figure 7, a similar table for images in figure 5 if expanded to their full form would occupy several pages. Should the full version be required it is still available.

R1 :if ω_1 & ω_2 & ω_3 & , ..., & ω_{16}	Then mask 1
R2: if ω_1 & ω_2 & ω_3 & , ..., & ω_{16}	Then mask 2
R3: if ω_1 & ω_2 & ω_3 & , ..., & ω_{16}	Then mask 3
R4: if ω_1 & ω_2 & ω_3 & , ..., & w_{16}	Then mask 3
R5: if $\pi\chi_1$ & $\pi\chi_2$ & $\pi\chi_3$ & $\pi\chi_4$	Then PC coeff 1

Figure 4 Rules that defines the PC's

The other table describes some of the rules that describe the expanded form without the inclusion of PCA.

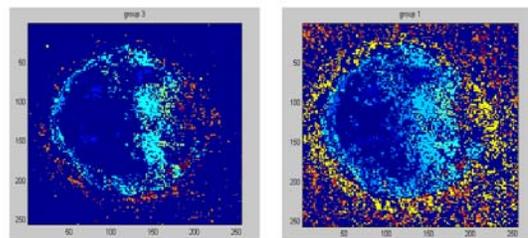


Figure 5 internal definition of a lesion image created by neural network

R1: If a0	Then b
R2: If b0 & b1 & b2 & b3 & b4	Then c
R4: If c1 & c2 & c3 & c4	then d1
R5: If a1 > k1 & a1 < k2	Then g1
R6: If a2 > k3 & a1 < k4	then g2
R7: if a1 > k5 & a1 < k6	then g3
R8: if a3 > k76 & a1 < k8	then g4

Figure 6 Extended rule definition

The graphs in figure 7, illustrate another aspect of each level of the hierarchy, the feature profile, and the

details that influence the genetic algorithm (GA) in the computation of the connective mesh.

The above detail shown in figure 6 pertains to the symbolic definition of the features contained within the lesion. The first of which is a single definition, the others increase in complexity. The last four rules pertain to the hierarchical decomposition which defines the level at which the feature resides in terms of the decomposition.

5 CONCLUSIONS

In our previous paper [Andrews et al, 2000] we showed how a novel neural architecture could be used to emulate the process of edge detection.

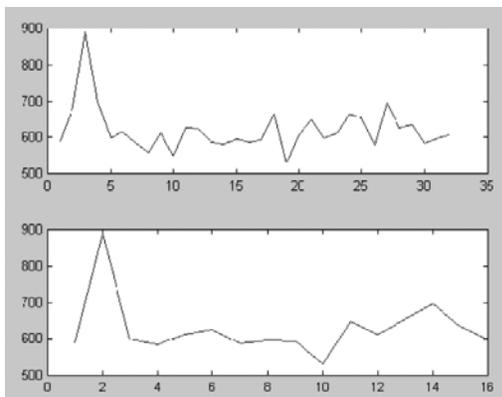


Figure 7 Graph profile of feature signature

In this paper we have extended the repertoire and tackled a few of the system's previous limitations. We have enhanced its capability by the inclusion of PCA that is used to simplify but also to encode the lesion image, and by the inclusion of a dermatology heuristic that describes the attributes of the lesion morphology [Friedman et al, 1985]. The heuristic, in the form of an implied grammar, is linked with the image transform, the HALOGRAPH [Andrews et al, 1998]. This combination adds rigour to the diagnostic process, and as a result there is less likelihood of ambiguity.

REFERENCES

ANDERSON J.A, (1970), 'Two Models of Memory Organisation', *Mathematical Bioscience*, 137-160.

ANDREWS S.G, HARRISON P V. (1998), 'The Fractal Dimension a Clue to Chaotic Growth an Approach to Skin Cancer Diagnosis'. *Intl. Conf EADV, Nice, 1998*.

ANDREWS S.G, ZEIN N.F, HONARY.B, HARRISON P, (1999), 'Analysis and Diagnosis of Dermatological Images with an Adaptive Associative Novel Neural Architecture', *EANN99, Warsaw Sept 13-19*.

ANDREWS S.G, ZIEN N F, (2000), 'Evolving Neural Architectures for Encoding and Recognition in Skin Cancer Lesion Images.' *ICAI2000, Las Vegas*.

AMARI S, (1977). 'Neural theory of association and concept formation'. *Biological Cybernetics*, 26: pp 175-185, 1977.

BRIDGETT N.A, BRANDT J, and HARRIS C.J. (1995): 'A Neurofuzzy Route to Breast Cancer Diagnosis'. *4th International Conference Artificial Neural Nets*, 448-453.

BROOMHEAD D.S, LOWE D, (1988). 'Multivariate Functional interpolation and adaptive networks'. *Complex Systems*, 2, 321-355.

BURRONI M, DELL'EVA G, PUDDU P, ATZORI F, BONO R, FERRANTI G, MACCHINI, PALLOTTA S, (1997): 'Early diagnosis of melanoma using artificial neural networks.' *Melanoma Research. Vol. 7, Supplement 1*.

CICHOCKI A, KASPRZAK W, SKARBK W, (1996) . 'Adaptive learning algorithm for principal component analysis with partial data'. In: *Cybernetics and Systems '96 volume 2. Austrian Society or Cybernetic Studies. Vienna, Austria, 1996*, pp 1014-1019.

CICHOCKI A, and UNBEHAUEN R, (1994). 'Neural networks for optimisation and signal processing'. *John Wiley & Sons. New York, 1994*.

GOLDBERG D.E. (1989): 'Genetic Algorithms in search, optimisation, and machine learning'. *Addison-Wesley*, 63.

GOUGH M.P. (1997): 'Associative List Memory'. *Neural Networks, Vol. 10. No 6*, 1117-1131.

FRIEDMAN R, RIGEL D, KOPF A, (1985). 'Early detection of malignant melanoma: The role the physician examination and self-examination of the skin'. *CA-35 (3)*. 130-150

HARP S.A, SALMAD T. (1991), 'Genetic Synthesis of Neural Network Architecture', (in *Handbook of*

Genetic Algorithm L. Davis Ed.) Van Norstrand Reinhold N.Y., 202-222.

HOLLAND J.H. (1975): 'Adaptation in natural and artificial systems'. Ann Arbor: University of Michigan Press.

KENERVA P, (1984). 'Self-Propagating Search: A unified theory of Memory'. PhD Thesis, Stanford University, CA. USA.

JOLLIFFE I.T, (1986). 'Principal Components Analysis', New York, Springer-Verlag.

KOHONEN T, (1974). 'An adaptive Associative Memory Principle', IEEE Trans. on Computers C-23, 444-445.

LAWS, K I, (1979). 'Texture Energy Measures'. DARPA Image Understanding Workshop (Los Angeles): 47-51.

MOALLEMI C. (1991): 'Classifying Cells for Cancer Diagnosis using Neural Networks'. IEEE Expert, 8-12.

NILSSON N J, (1980). 'Principals of Artificial Intelligence'. Palo Alto, Tioga.

OJA E, (1982), Principal components, minor components, and linear neural networks, Neural Networks, 5, 1992, 927-935

PANAGIOTIDIS N.G, KALOGERAS D, KOLLIAS S, D, STAFYLOPATIS A, (1996): 'Neural Network-Assisted Effective Lossy Compression of Medical Images'. Proceedings of IEEE, Vol. 84. Vol. 10, 1474-1487.

REYES-ALDASORA C C, ALGORRI GUZMAN M, (2000). 'A Combined algorithm for image segmentation using neural networks and 3D surface reconstruction using dynamic meshes'. V IBERO-AMERICAN SYMPOSIUM ON PATTERN RECOGNITION, LISBON, Portugal, September 11-13.

ROSENBLATT F, (1958). 'The Perceptron: A probabilistic model for information storage and organisation in the brain'. Psychological Review **65**, 386-408.

RUMLEHART D.E, HINTON G.E, WILLIAMS R.J, (1986). 'Learning internal representation by error propagation'. In Parallel Distributing Processing, 1

(D.E RumelHart, and J.L McClland Eds.) MA, MIT Press.

SPECHT D.F, (1990). 'Probabilistic Neural Networks and Polynomial Adaline as Complementary Techniques for Classification'. IEEE Transaction on Neural Networks. Vol. 1, No. 1, 111-121.

SANGER T. D, (1989). 'Optimal unsupervised learning in a single-layer feed-forward neural network', Neural Networks, 2, 459-473.

STOECKER W.V. MOSS R.H, MADSEN R.W, LEE H, C, ERCAL F, and UMBAUGH S.E. , (June 1997). 'Which Computer-detected features of pigmented lesions best predict a diagnosis of malignant melanoma.' Melanoma Research. Vol. 7, Supplement 1.

STOECKER W.V. MOSS R.H, (1992). 'Digital imaging in dermatology'. Compt Med imaging graphics, 145-150.

SWIERCZ M, LEWKO J, and MARIK Z, (1995), 'Application of neural networks to the prediction of intracranial pressure'. Biocybern, Biomed. Eng., 15, 93-104.

SWIERCZ M, MARIK Z, LEWKO J, CHOJNACKI K, and KOZLOWSKI PIEKARSKI P, (1998). 'Neural Network for detecting emergency states in neurological patients'. Medical & Biological Engineering & Computing, Incorporating CELLULAR ENGINEERING, Vol.36, No6. 717-722.

TOMATIS S, BARTOLI C, and BONO A, CASCINELLI N, CLEMENTE C, FARINA B, RADAELLI S, TRAGNI G, and MARCHESIN R, (1977). 'Advances in computer-diagnosis of Melanoma: the telespectrophotometric approach'. Melanoma Research, Vol. 7, Suppl. 1.

TUMER K, RAMANUJAM N, GOSHT J, RICHARDS-KORTUM R, (1998). 'Ensembles of Radial Basis Function Networks for Spectroscopic Detection of Cervical Precancer'. IEEE Transaction on Biomedical Engineering, Vol. 45, No 8. 953-961.

WERBOS P.J, (1974). 'Beyond Regression: New Tools for Prediction and Analysis in Behavioural Science'. PhD Thesis. Harvard University.

Biography



Stuart Andrews has over 35 years computing expertise covering many different domains. This expertise was gained both in industry and academia. He has also run his own business specialising in VLSI design tools research. Over the last two years he was a Technology manager assisting SME's solving many different types of problems related to embedded systems. He is now an Independent consultant. His main interests cover real-time systems, DSP as applied to communications systems, imaging systems, intelligent systems as applied to medical procedures. Currently his main topic of research is intelligent recognition & categorisation techniques in the form of hybrid neural networks.

A MULTISCALE METHOD FOR AUTOMATED INPAINTING

R.J.CANT, C.S.LANGENSIEPEN School of Computing and Mathematics, Nottingham Trent University

Abstract: We present a novel, simple and general method for image inpainting. Current methods may be crudely divided into those that aim to continue edges by various energy minimisation techniques, and those that perform texture synthesis from local information, but both have their weaknesses. Our method searches the image for areas of similarity and uses these to inpaint. By analysing the image at multiple resolution scales we can find similar features and textures from anywhere in the image at a reasonable speed. We present results using some challenging images where both features (edges) and textures from non-local information are used to achieve plausible restoration. Keywords: Inpainting, image restoration

1 INTRODUCTION

Image manipulation has a long and not very illustrious history. Stalin not only had people executed, but had their images removed from photographs as if they had never existed [1]. These days, there are more reasonable reasons why images may be edited before publication, from the ‘improvement’ of someone’s appearance to the clarification of a scene for journalistic effect. Usually this involves a substantial amount of work by the user, to make the changes blend in with the remainder of the image. This paper proposes a method of inpainting an image after the removal of a feature, which requires the minimum of human intervention. Its aim is to be able to reconstruct pertinent features and textures to generate a plausible image without explicitly searching for artefacts such as edges.

2 RELATED WORK

There has been considerable work in the field of digital inpainting, approaching the problem from the directions of noise removal (due to compression and transmission), film and video artefact removal, and texture synthesis. [11] provides a comprehensive summary of work done in this area. Early work such as [5] concentrated on image reconstruction after effects caused by the scanning device characteristics. Work of particular relevance includes the texture synthesis method [15] based on Heeger and Bergen’s [13] pyramid texture analysis. This produced convincing inpainting of large areas of small-scale texture e.g. grass, but no features or edges were included. [14] removed image noise while retaining line continuity, but at the cost of requiring manual choice of regions for spatial and frequency samples. De Bonet [8] used multiple resolution methods to reconstruct texture, showing that such a method could synthesise more difficult textures where overall feature directionality had to be maintained. More recently, [20] provided a multi-resolution method of impressive speed for infilling areas of texture. However, this method was purely on a single texture, and is not appropriate to more general inpainting, since image segmentation would also be needed. Methods which concentrated on edge, line, curvature continuity included work on the TV model [6] and the PDE approach of [3]. The latter produced impressive results, though the authors concede that, because it concentrates on achieving isophote continuity, it would have trouble with areas of texture. Their later work [2][4] includes what appears to be even more impressive inpainting of the standard Lena image. However, their method includes the use of field information directly from the undamaged image to direct the inpainting of the damaged area, so can really only be considered appropriate where there is such an image available e.g. film frames.

3 THE METHOD

Our method is based on work first done as an undergraduate student project [16] on black and white images. This method used pixels from other similar areas of the image to inpaint the area. If a region was to be inpainted, the process was started with the

outermost pixels – those adjacent to parts of the image that were to be retained. The pattern of levels of pixels in the patch surrounding it was then compared with the remainder of the image to find the closest equivalent pattern match. ‘Closest’ was simply defined by the sum of the grey values for the patch. The pixel in the equivalent position within this patch was then used to replace the pixel to be reconstructed.

Although this method appeared quite successful, a few problems were evident. It was quite good at generating ‘plausible’ reconstructions of amorphous areas such as foliage, and of simple repeating structures such as identical windows in a distant building. However it could not cope with more distinct edges as in individual leaves, or repeating structures with ranges of scales within them. It was also extremely slow, and experiments had to be limited to small images. An apparently identical technique was later used by [9], who also commented on its slowness.

We modified the technique by handling features in a way that seems to match human perception. If one reduces the resolution of an image by a factor of 2, 4 etc., one can still see the grossest features. One observes this effect when watching the ‘pixellated’ faces used to hide identities on screen – the facial shape, nose etc can still be seen if one ‘squints’ at it. By using a reduced resolution version of the image, the areas to be inpainted could be matched up with similar areas elsewhere on the image – where similarity is assessed at this reduced resolution. In other words, the method is finding features visible at this resolution which are relevant to the reconstructed area. Each resolution scale acts as a filter which selects out features at that scale for comparison. This gives a comparatively quick method of finding the best matches for the pixels to be reconstructed. Firstly, the picture resolution is reduced by a factor of 2^*N . As in the original process, the whole image (at this resolution) is scanned to find the best matches for the patch surround the pixel to be inpainted. The set of best matches (currently defined purely as a fixed number of matches) is then used as the starting set (rather than the whole image) for a similar matching process at a higher resolution. This process will match on features at a smaller image scale. Regions around each of the best matches are then examined at the higher resolution to again find the best match with the original. This limits the search space, yet allows for some variation in best match from one scale to another. The process is repeated until the final target image resolution, whereupon the best match is used to generate the reconstructed pixel (Figure 1). At each resolution level, each pixel in the candidate patch is compared with its equivalent in the original patch (in RGB), and the differences summed to give the overall measure for that candidate patch. Each candidate patch is also reflected about x and y axes and rotated by a range of angles for comparison, since a feature may be rotated/reflected elsewhere in the image relative to the one being reconstructed.

This method has a number of advantages over the original method discussed and other work. Firstly: speed. The process of exhaustive comparison of each pixel in the image against the one to be filled (particularly as one has to look at a patch of pixels

surrounding it) can be prohibitively slow. By performing the exhaustive search only at a much lower resolution, the overall process becomes much faster, as the search space used for the higher resolutions are then heavily culled. However, even with this improvement, images of 640*480 and 24 bit colour (which we used for convenience) could take from a few minutes to a couple of hours on a reasonable PC, depending on the size of area to be filled. The method of [3] is apparently faster, because apart from the initial smoothing, it examines pixels local to the inpainting area only but does not handle texture.

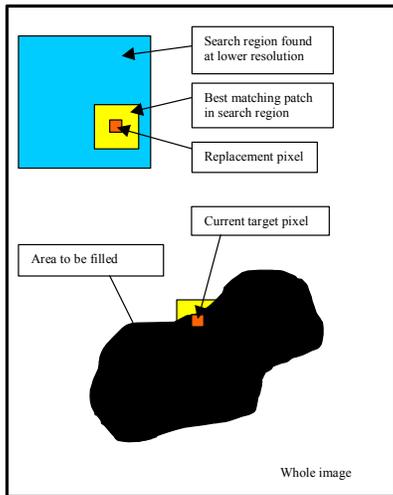


Figure 1 One stage in multiscale process

Secondly: feature identification. As discussed by DeBonet, human vision is very good at seeing features at a range of scales. That is why Gothic architecture is so appealing; it has repetition and variation at many different scales within a single building. By using a process that attempts to mimic the way we ‘pan and zoom’, the process has a better opportunity of finding the features that would affect the plausibility of the reconstruction. Thus this method, unlike [3] can also reconstruct texture, which is visible only at the higher resolutions.

Thirdly: generality. No explicit edges or lines are detected; the pixel colours in the patch are simply compared with those in the candidate patches. This means that one does not have to start looking for ‘special cases’ such as curved or straight edges. No explicit weighting is done to favour pixels closest to the one to be reconstructed, as otherwise some choice would have to be made as to how to weight. The effect of weighting is achieved by the use of lower resolutions, since each resolution scale essentially averages a different number of pixels.

The process does have some free parameters. As will be discussed later, choice of the size of patch to use and number of resolution scales can affect the quality of the reconstruction. Choice of region size mainly affects the process speed, since it is used to limit the search space. However a larger search space around the notional best match can help for local texture. A less important constraint, made solely for performance reasons, is that we limited the choice of rotation angles used for the comparison. Since the built landscape contains features that may vary in angle due to perspective, the matching process should rotate the comparison patches by small angles from the notional horizontal and vertical to find the best match. However, in the natural landscape, features may occur at any angle, so the rotation should be through a wider range of angles – we chose to use every 90 degrees (ie 8 rotations and tests for every patch including reflections). With more CPU speed, both sets of rotations could be tried for every image, and the comparison process itself would

eliminate the worst matches.

4 RESULTS

4.1 The Best

In Figure 2 of a building in Prague, there are a challenging range of features of differing scales and textures. Figure 2 shows the original lamppost in the scene, Figure 3 shows the image forming the starting point for the reconstruction, after removal of the lamppost, and Figure 4 shows the resultant reconstruction.



Figure 2 Original Image

Note that the process has managed to reconstruct the vertical glazing bars automatically, as well as the triangular shapes between the windows and the smaller scale pavement texture. We believe this capability in feature and texture is unique to our multiresolution method. The vertical glazing bar in the bottom window would not have been inpainted by the edge continuity techniques, because it was completely removed by the mask. It would not have been found by the texture method [16] without massive enlargement of the patch size, which would have ensured the pavement texture would have degenerated into garbage as discussed in [9]. [20] found that there tended to be discontinuities across the inpaint boundary in their multiresolution method unless they modified the process. However, they were only using multiple scales to enlarge the local region. In contrast, we are using the lower resolutions to sample and include non-local regions of the image, and find that continuity across the boundary is reasonable at the higher resolutions. Ashikhmin[1] who modified the Wei-Levoy method to achieve some elegant synthesis of natural textures, commented that multiresolution did not appear to assist much in his work on single textures – we have found it essential for images with a range of features and textures.

For this image, the most plausible reconstruction was obtained using a patch size of 9 pixels, search region of 4 pixels around the 4 best candidate patches, and 2 lower resolution scales before the final processing. Patches were compared at their original angles and at slight deviations from the horizontal and vertical as mentioned earlier. The latter process seems to have resulted in the best matches for the glazing bars that are not quite vertical – further work needs to be done in exploring whether the best matching pixel should be modified if the angle is not as the original. In the lower window, the left hand horizontal glazing bar is not reconstructed, though the right hand one is. This is

plausible because one of the other windows (bottom left) also exhibits only one horizontal bar, since the other half window is open. The process has used this area in its reconstruction of the rightmost window.



Figure 3 Area masked for infill

A problem involving natural rather than built artefacts is this dolphin image. Figure 5 shows the original image, Figure 6 the masked out dolphin, and Figure 7 the best infilling attempt. For this image, the most plausible infilling occurred with a patch size of 3 pixels, a search region of 8 pixels, and 2 lower resolution scales. In this case, patches were compared at a wide range of rotation angles rather than just slight variations off vertical. The wave shape and local texture appear to be plausible.

Most of the images upon which we have attempted the technique have provided the most plausible reconstructions when using 2 lower resolution scales. However, this is partially related to the image size. Higher resolution versions of the same images need one or more additional low resolution stages to attain the equivalent level of image inpainting.

4.2 The Worst

Figure 8 shows the problems associated with trying to inpaint a relatively large area within an image. The process begins by trying to find the best matches to those pixels on the perimeter of the reconstruction area that have the most neighbours in the original image. It then fills those with the next highest count of 'real' neighbours until the whole perimeter has been covered, before moving inwards. This means that as one gets closer to the centre of a large area for reconstruction, more and more of the pixels used for comparison have actually been reconstructed themselves. As commented by Wei and Levoy [20], this spiralling inwards is necessary to avoid directional bias caused by the obvious artificiality of following scan lines. Furthermore, because of the freedom of matching, adjacent pixels in the area to be filled may not be sourced from contiguous parts of the original image. This tends to cause a 'smoothing' of texture in the centre of a large fill area, and this phenomenon can be seen where the Mountie has been replaced in Figure 10. Potentially, 'noisiness' could be added to the interior of the region, perhaps by the method of [13] or [15] but would need automated analysis to identify the degree to which the texture encloses the region for inpaint. A better solution might be to allow the user to indicate that noisiness was required.

Note, however, that the process has constructed a plausible beach texture, and continued the edges at the lakeside and distant shoreline successfully. The 'Connectivity Principle' presented by [7] and discussed by [11] with regard to the Mumford-Shah model's failings is preserved with this technique, as well as the restoration of texture provided the area is not too large

Figure 13 shows the problems associated with the reconstruction of areas in distinct predictable lines. The human eye/brain combination is very good at pattern finding, and so the narrow sections where the wires have been removed can still be seen in the middle left of the image. As commented by Ashikhmin, the human eye will filter out discontinuities in an image with high frequency content – this is the manner in which the simple version of chaos mosaic texture synthesis [12] achieves plausibility. The bear image is not 'busy' and so artefacts would be comparatively visible. There are real vertical features in the right hand area of water, so to an observer who had not seen the original, the image would still be plausible. Interestingly, the horizontal 'replaced wires' are far harder to see, despite both inpainting areas being of similar dimensions. However the bear fur texture, body edge, stones and water are reasonably well reconstructed.



Figure 4 Lamppost filled in



Figure 5 Original Image



Figure 6 Dolphin masked out



Figure 7 Image after processing



Figure 9 Masked image



Figure 10 Best inpainting



Figure 8 Original image of Mountie



Figure 11 Original image of bear



Figure 12 Masked image

4.3 Analysis of results

Our experiments with a range of images showed that there were noticeable differences in the visual quality of reconstruction depending on the values of patch and region size chosen for a particular image and on the number of resolution scales used for the earlier stages. A small patch size means that the filled pixel correctly reflects small-scale variation i.e. texture, but may not achieve the correct overall shade. A large patch size may provide a better match to the local shade, but loses the opportunity to show texture, and ‘smoothes’ the image. As mentioned earlier, the optimum number of resolution scales tends to relate to the original image size, but some images did need more or fewer scales.



Figure 13 Best inpainting

These points can be illustrated by the example shown in Figure 14. This picture of lilies has its most plausible reconstruction following the removal of one stem (as shown in Figure 15) by using only 1 lower resolution scale (i.e. a factor of 2 in x and y) – see Figure 16. When using 3 lower resolution scales (Figure 17) the program failed to continue the horizontal wire correctly across the infill area. In both cases the patch and region sizes were the same.

It became apparent during the course of experiments that the experimenter could make a reasonable guess at the best parameters to use for the reconstruction from simply viewing the image. Further investigations are thus required to understand how this is occurring. For this inpainting method to work, parts of the area being reconstructed must be replicated (approximately) elsewhere in the image. The number of resolution scales used acts as a crude filter into frequency bands, whereas the patch size itself relates directly to the scale of the feature. The region size

used for the local best match relates to the ‘wave packet’ scale, the distance within which the highest frequencies are part of some common feature. Initial Fourier analyses of our test images have not yet led to any clear relationships between the image scales and the parameters.

5 FUTURE WORK AND CONCLUSIONS

In order to derive the relationship mentioned above, further experiments need to be conducted. Because at present plausibility is so difficult to define mathematically, we will have to use human testing to achieve at least a statistical measure. For example, although [19] calculate an error measure for their magnification process, they found images with similar error values could have markedly different visual appeal. In a similar vein [7] found that some algorithms generated solutions which caused lines to be terminated, while human perception much prefers continuity. As a result of these difficulties, much of the work discussed earlier presents the images to the reader without evaluation or objective comparison (as do we). As the processes for inpainting improve, we have to start measuring the plausibility of the resultant images, in order to assess whether any new technique is a genuine improvement.

Assuming that human tests can give us a relationship for our free parameters to the image feature scales, the process of digital inpainting can then be made more automated. Given an image with a masked region, the program could use an initial analysis to determine plausible values for the 3 parameters, perform the inpainting using these and some nearby values, and thus provide a small set of reconstructed images from which the user could make the final selection



Figure 14 Original image and enlarged region



Figure 15 Masked image and enlarged region

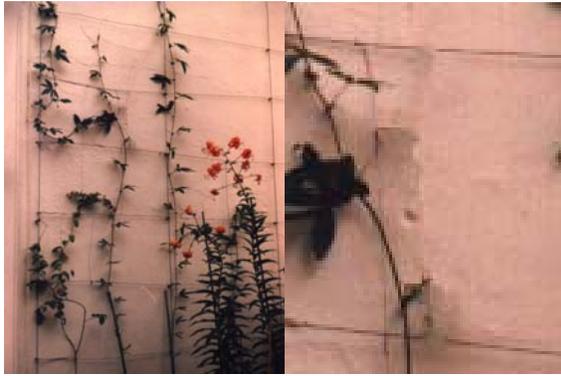


Figure 16 Best infill, and enlarged region

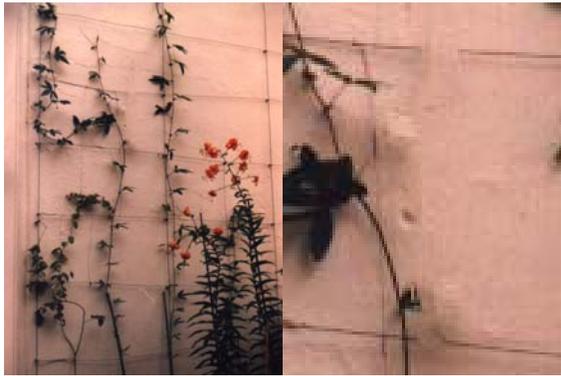


Figure 17 Infill using more lower resolution stages

If human tests could further assist in identifying which technique was most suitable for a given class of image, one could extend the process. Apart from simple Fourier or wavelet analysis, some segmentation could be carried out eg as in [18] to categorise the areas of texture to be inpainted. The most appropriate techniques could then be selected and used. Such a tool could free the user from the drudgery of manual editing, while still giving them the freedom to decide on the best image to use.

6 ACKNOWLEDGEMENTS

Thanks to .P. Bown, M.R.Cant for use of some images.

References

- [1] M. Ashikhmin. Synthesizing Natural Textures. Proceedings of 2001 ACM Symposium on Interactive 3D Graphics, Research Triangle Park, North Carolina pp. 217-226, March 2001.
- [2] C.Ballester, M.Bertalmío, V.Caselles, G.Sapiro and J.Verdera. Filling-In by Joint Interpolation of Vector Fields and Gray Levels. IEEE Trans. Image Processing, 10(8),pp1200-1211, August 2001.
- [3] M. Bertalmío, G. Sapiro, V. Caselles and C. Ballester. Image Inpainting. Proceedings of SIGGRAPH 2000, pp 417-424, New Orleans, USA, July 2000.
- [4] M. Bertalmío, A. Bertozzi, G. Sapiro. Navier-Stokes, Fluid-Dynamics and Image and Video Inpainting. IEEE CVPR 2001, Hawaii, USA, December 2001.
- [5] T.E.Boult and G.Wolberg. Local Image Reconstruction and Sub-pixel Restoration Algorithms. CVGIP, Graphical Models and Image Processing, 55(1), pp63-77, 1993.
- [6] T.F.Chan and J.Shen. Mathematical Models for local

non-texture inpainting. SIAM, J.Appl.Math,63(2) pp1019-1043,2001.

- [7] T.F.Chan and J.Shen. Non-texture inpainting by curvature driven diffusion (CDD). J.Visual Comm. Image Rep, 12(4), pp436-449, 2001.
- [8] J.S.DeBonet. Multiresolution sampling procedure for analysis and synthesis of texture images. SIGGRAPH 97, pp361-368, 1997..
- [9] A.Efros, T.Leung. Texture synthesis by non-parametric sampling. Proc. IEEE International Conference on Computer Vision, pp1033-1038, Corfu, Greece, September 1999.
- [10] A. Efros W.T. Freeman. Image Quilting for Texture Synthesis and Transfer, Proceedings of SIGGRAPH '01, pp 341-346 Los Angeles, California, August, 2001.
- [11] S.Esedoglu, J.Shen. Digital inpainting based on the Mumford-Shah-Euler model. European J.Appl.Math, 13, pp353-370, 2002.
- [12] B. Guo, H. Shum, Y.-Q. Xu. Chaos Mosaic: Fast and Memory Efficient Texture Synthesis. Microsoft research paper MSR-TR-2000-32.
- [13] D.Heeger, J.Bergen. Pyramid based texture analysis/synthesis. SIGGRAPH 95 pp229-238, 1995.
- [14] A.N.Hirani, T.Totsuka. Combining frequency and spatial information for fast interactive image noise removal. SIGGRAPH '96, pp269-276, New Orleans, LA, 1996.
- [15] H.Igehy and L.Pereira. Image replacement through texture synthesis. Proceedings of 1997 IEEE International Conference on Image Processing, pp 186-189, Santa Barbara, Oct 1997.
- [16] J.Keen. Image reconstruction after object removal. BSc thesis, Nottingham Trent University 1997.
- [17] D. King. The Commissar Vanishes, Holt, Henry & Co. ISBN: 0805052941.
- [18] J. Malik, S.Belongie, T.Leung, J.Shi. Contour and Texture analysis for image segmentation. International Journal of Computer Vision, 43(1), pp7-27, 2001.
- [19] B.S.Morse, D.Schwartzwald. Image Magnification and Level-Set Reconstruction. Computer Vision and Pattern Recognition 2001 (CVPR'01).
- [20] L.-Y.Wei and M. Levoy. Fast texture synthesis using tree-structured vector quantization. Proceedings of SIGGRAPH 2000, pp479-488.



Richard Cant started life as a theoretical physicist, then moved into industry, where he spent 9 years as a system designer working on computer generated imaging for military training systems. His current research interests continue this theme and include use of neural nets for evaluation of image algorithms, including their validity for training.



Caroline Langensiepen started as a theoretical physicist, moved into industry as a nuclear physicist, then spent 8 years on system/software design of high reliability comms. systems & training simulators. She followed this with a period as an independent consultant for very large mission-critical systems. Her current interests include image processing and real-time sensor information analysis.

A FORMEL LANGUAGE FOR SOFTWARE REUSE

ZINA HOUHAMDI

*Computer Science Department, University of Biskra
BP 145, Biskra RP, 07000, Algeria.
E-mail: z_houhamdi@yahoo.fr*

Abstract : Software reuse has been claimed to be one of the most promising approaches to enhance programmer productivity and software quality. One of the problems to be addresses to achieve high software reuse is organizing databases of software experience, in which information on software products and processes is stored and organized to enhance reuse. This paper presents a new approach to define and construct such databases called the Reuse Description Formalism (RDF). The formalism is a generalization of the faceted index approach to classification. Unlike the faceted approach, objects in RDF can be described in terms of different sets of faceted and in terms of other object descriptions. This allows a software library to contain different classes of objects, to represent various types of relations among these classes, and to refine classification schemes by adding more detail supporting a growing application domain and reducing the impact of initial domain analysis. In particular, *RDF* provides a specification language based on concepts of set theory capable of representing a rich variety of software and non-software domains; it provides a retrieval mechanism based on exact matches and similarity metrics which can be customized to specific domains; and it provides a mechanism for defining and ensuring certain semantic relations between attribute values.

Keywords. Software reuse library, classification system, taxonomy, similarity, retrieval process, specification language.

1. INTRODUCTION

Complex computer programs have placed a growing demand on the talents of software engineers as well as on existing technologies for software development. In order to keep up with the increasing complexity of today's software systems, productivity must be increased and cost reduced in all phases of the software construction process [2]. An important aspect of the projected solution to this growing demand for new software is the development of support technologies to help increase software reuse, that is, the reapplication of knowledge about one system to other similar systems [1,3]. Rather than starting from scratch in new development efforts, the emphasis must be placed on using already available software assets (e.g., processes, documents, components, tools). This approach avoids the duplication of work and lowers the overall development cost associated with the construction of new software applications. One important characteristic common to most approaches to software reuse is that they rely, either explicitly or implicitly, on some kind of software repository or library from where the "basic building blocks" are extracted. The fact that software libraries are such an important aspect of most reuse systems, has made software reuse library systems (i.e., systems for designing, building, using, and maintaining software libraries) a very important research topic in the area of software reuse [7].

Although these classification models provide the basis for a useful software reuse library system, they have significant limitations and, therefore, can only be regarded as a first step towards a more complete system. They all suffer from one or more of the following problems [4]:

Restricted domain. Some reuse library systems have been designed with the purpose of improving reuse at code level. Their representation language usually does not have the expressive power to model more abstract or complex software domain (e.g. software project, defect, or processes).

Poor retrieval mechanism. One essential characteristic of any software reuse library system is to allow the retrieval of candidate reuse components based on partial or incorrect specifications. This functionality requires the ability to perform similarity-based comparisons, but most systems only provide retrieval based on partial keyword matches or predefined hierarchical structures.

Not flexible. Software reuse library systems must evolve as the level of expertise in an organization evolves. Because of this, a software reuse library system must be flexible enough to allow the incorporation of new classification schemes or new

retrieval patterns, yet this is not the case in most systems.

No consistency verification. Most software reuse library systems are based on representation models, which must satisfy certain basic predicates for the library to be in a consistent state. Yet, most of these systems do not provide a mechanism for ensuring this consistency.

This paper proposes a classification system for software reuse called the Reuse Description Formalism (RDF) which addresses the limitations of current software reuse library systems. RDF is based on the principles of faceted classification, which have proven to be an effective mechanism for creating such systems [8,11]. RDF is capable of representing a rich variety of software (and no-software) domains; provides a powerful and flexible similarity-based retrieval mechanism; and provides facilities for ensuring the consistency of the libraries.

2. FOUNDATION OF RDF

The Reuse Description Formalism uses a generalization of the faceted classification approach proposed by Diaz [10] to represent and classify software objects. The faceted index approach relies on a predefined set of facets defined by experts. Facets and associated sets of terms form a classification scheme for describing components. Component descriptions can be viewed as a records with a fixed number of fields (facets), where each field have a value selected among a finite set of values (terms). Faceted classification scheme has proven to be an effective technique to create libraries of reusable software components. Yet, it suffers from various shortcomings, which limit its usefulness and applicability. The RDF approach to classification overcomes these limitations by extending the representation model as follow [5]: Components are replaced by instances that belong to several different classes. Instances and classes are defined in terms of attributes and other classes, supporting multiple inheritance.

Facets are replaced by typed attributes. Possible types are: integers, string, enumerations, classes, and sets of the above. Having instances as attribute values allows a library designer to create relations among different instances (e.g., that **push** is a component of **stack**).

The concept of similarity is extended to account for the richer type system, including comparisons of instances of different classes and comparisons of set values.

Semantic attribute relations can be defined and checked using the **assertion** construct. This facility simplifies the process of maintaining the consistency of the definitions in a software library.

An integrated language describes attributes, terms, classes, instances, distances, and their dependencies. Descriptions are type checked. The language is based on a formal mathematical model, which makes it both coherent and analyzable.

2.1. Representation model

To understand the representation principles of RDF, it is useful to consider descriptions of objects of a particular class as point in a multidimensional space, where each dimension is represented by an attribute. Attributes have a name and a list of possible values defined by their associated type (i.e., set of values). If a is an attribute name, and v belongs to the a 's type, the assignment " $a = v$ " represent the set all objects whose attribute a is v . Assignment can be combined in expression to define other sets of objects. In particular, if A_1 and A_2 are two assignments, the expression " $A_1 \& A_2$ " represents the intersection of the sets A_1 and A_2 . Similarly, " $A_1 | A_2$ " represents the union of these sets. In addition, the set of objects that have been defined in terms of a particular attribute a , independently of the value associated with a , is denoted by "**has** a ". The set of objects defined by the "has" operator is a short form of the expression " $a = v_1 | a = v_2 | \dots | a = v_n$ " where the value v_i are the elements of the type of a . A set of objects is called a class in RDF. Classes can be given a name they are denoted as **class** (E) where E is an expression; i.e., unions and intersections of other sets of objects. If c is a class name, the set of objects it represents is denoted by "**in** c ", and can be combined with other sets of objects in an expression. An object description is called an instance in RDF. Instances can be given a name and they are denoted as **instance** (E) or $[E]$ where E is an expression. Semantically, an instance must have only one set of attributes, therefore we say that instance (E) is well defined if and only if: (1) E is not a contradiction (i.e., $\text{class}(E) \neq \emptyset$), (2) E defines a mapping from attributes to values, that is, E can be simplified into a consistent conjunct of assignments.

Expressions can also be used to characterize particular sets of instances defined in a RDF library. We denote by **set** (E) the set $\{i | i \in D \cap \text{class}(E)\}$, where D is the set of instances in the library. In other words, the **set** operator defines the set of instances in the library that belong to the class defined by E .

2.2. Similarity Model

The goal of any Reuse Library System is to facilitate the process of finding suitable objects for reuse. RDF supports two criteria for selection candidate

objects: by exact match and by similarity. For exact matches the construct set (E) already described is used. Similarity-based queries are performed using the construct "query E", which denoted the list of instances in the library sorted by decreasing similarity to the target object define by E. That is, the first element of the "query E" is the best reuse candidate for [E], the following element the second best, and so on.

As mentioned earlier, similarity is quantified by a non-negative magnitude called similarity distance, which is used as an estimator of the amount of effort required to transform one object into another. Because of this, distances between two object descriptions, A and B, are not symmetric, because the effort to transform A into B is not necessarily the same as the one required to transform B into A. For this reason, whenever a distance is computed, it is important to define which object is the source and which the target.

Let Z be an object class defined by the set of attributes $Z' = \{A_1, \dots, A_n\}$, and S and T be two instances in this class. Also, let $S' \subseteq Z'$ be the actual set of attributes used to define S, and similarly for T'. The distance from S to T is denoted by D(S,T) and is computed as follows:

$$D(S,T) = \sum_{A \in S' \cap T'} K_A T_A(SA, TA) + \sum_{A \in S' - T'} K_A R_A(SA) + \sum_{A \in T' - S'} K_A C_A(TA)$$

Where I.A denotes the value of an attribute A on an instance I. The set $S' \cap T'$ represents the attributes shared by S and T, while $S' - T'$ is the set of attributes found in S but not in T, and similarly for $T' - S'$. These three sets are disjoint. In addition, each constant K_A is called the relevance factor of attribute A. Their values fall in the range 0 to 1., and must satisfy the relation $\sum_{A \in Z'} K_A = 1$.

Functions T_A , R_A and C_A are called comparators, and are explained later in this section.

The expression for distance D(S,T) is based on the assumption that the overall transformation effort from S to T can be computed using a linear combination of the differences between their respective attributes. In other words, attributes are considered independent of each other when computing similarity. This is a strong assumption that limits the types of domains that can be handled by RDF's similarity model.

Relevance factors. In general, the distance between two RDF objects is given by the sum of the distances between their corresponding attributes. This default scheme gives equal importance to all attributes. In our particular situation, this is not a reasonable assumption. For example, one would

consider that the difference between component subsystems is more important than the difference between their number of lines of source code. Therefore, the first step required to design comparators is to assign a relevance factor to each attribute in the representation model, that is, to define the amount of influence they have in the computation of similarity distances.

Comparators. Explained earlier, each attribute has three associated functions T_A , R_A and C_A called comparators. T_A is the transformation comparator and is used to qualify the amount of effort required to transform one value of attribute into another. R_A is the removal comparator and is used to estimate the amount of effort required to eliminate a source attribute value not required in the target specification. Finally, C_A is the construction comparator and estimates the amount of effort required to supply a target value not specified in the source specification. The set of all attribute comparators plus their associated relevance factors define a specific similarity model for reuse library. These functions and values must be specified using a process called domain analysis [9] which, among other thing, defines the criteria for similarity for objects in a particular domain. Nonetheless, RDF provides default comparators for each type of attribute. These default comparators can be used as a starting point from which to refine the similarity model of a library. This refinement is normally done by assigning attributes non-default comparators using "foreign" functions specified in some conventional programming language.

RDF defines default comparators for each different kind of RDF type. Although default comparators are well suited for certain domains, sometimes it is necessary to define alternative comparators to be able to capture the semantics and relations of specific objects and attributes. For this purpose, RDF allows the library designer to define arbitrary comparators, which can be assigned to any attribute or type using the "distance" clause.

2.3. RDF Specification language

This section presents a formal definition of the syntax of the RDF language. Syntax is presented in a variation of the BNF using the following conventions: Keywords and symbols occurring literally are written in bold; non-terminals are written in italics; type-name, attribute-name, instance-name, term, and class-name all denote identifiers; symbol, ... means one or more occurrences of symbol, separated by commas; and keyword_{opt} means that the keyword may or may not occur, without affecting the semantics.

Declarations: A RDF library consists of a sequence of declarations. Each declaration either defines a

name (of a type, an attribute, an instance, or a class) or describes an assertion that must be true of all instances in the library.

Library ::= declaration

Declaration ::= type-declaration | attribute-declaration | instance-declaration | class-declaration | assertion

Attributes and types: Software components and other objects are described in terms of their attributes. We can think of attributes as fields of a record describing the object. The declaration of an attribute specifies the type of the values for the attribute. RDF supports the following types: number, string, term enumerations, object classes, and homogeneous sets of the above.

Attribute-declaration ::= **attribute** attribute-name : type;

Type-declaration ::= **type** type-name = type;

Type ::= simple-type distance-clause | **set** distance-clause of type

Simple-type ::= number | string | {term, ...} | class | type-name

Distance-clause ::= **distance**_{opt} | **no distance** | **distance** {triplet,...} | **distance** *{triplet,...}

triplet ::= term_{opt} → term_{opt} : number-literal

The keyword **distance** by itself is optional and assigns default distance functions. The case “**no distance**” indicates that the distance between values of the associated type is always zero. In the third and the fourth forms of the distance clause, the triplet $t1 \rightarrow t2: n$ means that the distance from $t1$ to term $t2$ is n . If $t1$ is omitted the unspecified value is assumed (i.e., n is creation distance of $t2$). If both $t1$ and the arrow are omitted, the previous $t1$ is assumed. If the keyword **distance** is followed by the character “*”, then the distances between terms not mentioned in a triplet will be set to infinity. If “*” is not specified, distances between all terms will be adjusted by computing the shortest path between them.

Expressions: Expression are formed from attribute assignments, the unary operators has and in, and the binary operators & (intersection) and | (union).

Expression ::= attribute-name = value | **has** attribute-name | **in** class-name | expression & expression | expression | expression | (expression)

The expression “attribute-name = value” means that the value of attribute-name for the instance being defined is value. The expression “**in** class” means that the instance defined belongs to the class; it is similar to a macro-expansion of the expression that defines the class. The expression “**has** attribute-name” denotes the condition that the instance being defined has some value for attribute-name.

Values: Values are used in assignment expressions. Values are either simple values or set values. A

simple value is either a literal (number or string), a term, an instance, or the value of an attribute of an instance. Set values must denote homogenous sets; they are described either by extension or by intention, using the

set construct. Only sets of instances can be described by intention.

Value ::= simple-value | {simple-value, ...} | **set** (expression) | **set** (instance-name | expression)

Simple-value ::= number | string | term | instance | **self** | Instance.attribute-name | **self**.attribute-name

The construct **set** (E) represents the set of all instances in the library that satisfy the expression (i.e., that belong to **class** (E)). If the optional instance-name is used, the name is bound within E to each instance in the library. The dot notation “instance.attribute-name” is used to refer to the value of the attribute attribute-name of an instance. This notation is similar to that used in other languages to access record fields. The keyword **self** is a reference to the instance defined by the expression in which the value is used. Within an **instance** construct, **self** is bound the instance defined. Within an **assertion**, **self** is bound to every instance in the library in turn. Within nested **instance** construct, **self** is bound to the innermost instance.

Classes: A class is defined by giving the corresponding expression; the class denotes the set of all objects for which the expression holds. Classes are used to abstract properties of instances and also as abbreviations for the corresponding expressions. Classes are also used as types of attributes whose values are instances.

Class-declaration ::= class-name = class;

Class ::= **class** (expression) | class-name

Instances: Instances are defined in terms of an expression. An instance defined by an expression E is a representative of the class of instances defined by “**class** (E)”

Instance-declaration ::= instance-name = instance;

Instance ::= **instance** (expression) | [expression]

An instance may not exist either because the class is empty (i.e., the expression is a contradiction) or because the class is not specific enough (i.e., it defines more than one valid set of attributes) a sketch of a possible simplification and verification algorithm is as follows.

Expand all “**in**” propositions with the expressions of the corresponding classes.

Transform the expression into disjunctive normal form, as follows:

Restructure the expression using associativity laws so that no disjunction occurs within a conjunction. Represent each conjunct as a set of assignments and has propositions.

Represent the expression as a set of these conjuncts. For each conjunction do the following:
Delete redundant assignments.
If there are still two assignment to the same attribute, or there are unsatisfied has propositions, delete the conjunction.

Else, delete has propositions (not needed anymore). Delete conjunctions that imply another conjunction. If there no conjunctions left, fail (E is a contradiction)
If there are more than one conjunction left, fail (E is not specific enough)

Assertion: An assertion specifies a semantic constraint that must be true of all instances in the library. Expressions are used to represent dependencies between attributes, to constrain data types and classes, and to enforce correct typing.
Assertion ::= **assertion** expression \Rightarrow expression;

The meaning of “**assertion** $E_1 \Rightarrow E_2$ ” is similar to **set** (E_1) \subseteq **set** (E_2). This definition does not capture subtleties with respect to the binding of **self**. RDF signals false assertions

Queries and distance computations: Queries are used to examine a RDF library; they are not part of the library itself. A query command computes a list of instances in the library sorted by decreasing similarity (increasing distance) to the implicit target instance define by an expression. The syntax of queries is:
Query ::= **query** expression | **query** expression : identifier

If specified, identifier must be the name of an attribute or a type, and distances are computed using the distance functions associated with the type or the attribute. If identifier is not specified, distances are computed using the default distance functions provided by RDF. The distance command is used to compute similarity distances between a pair of values. This command is useful for verifying the definition of distance functions and the results they produce.

Distance ::= **distance** source-value_{opt} \rightarrow target-value_{opt} | **distance** source-value_{opt} \rightarrow target-value_{opt} : identifier
The source -value and target-value must be values of the same type (e.g., instance names). In case of terms, they must belong to the same enumeration. If both names are specified, the command computes their transformation distance. If only the source value is given, its destruction distance is computed. Finally, if only the target is specified, its construction distance is computed. The identifier

has the same use as in the case of the query command.

3. CONTRIBUTION OF THIS WORK.

As explain earlier, current software reuse systems based on the faceted index approach to classification suffer from one or more of the following problems: they are applicable to a restricted set of domains; they posses poor retrieval mechanisms; their classification schemes are not extensible; and/or they lack mechanisms for ensuring the consistency of library definitions. The primary contribution of this dissertation is the design and implementation of the Reuse Description Formalism [6], which overcomes these problems.

RDF is applicable to a wide range of software and non-software domains. The RDF specification language is capable of representing not only software components at the code level, but it is also capable of representing more abstract or complex software entities such as projects, defects, or processes. What is more, these software entities can all be made part of one software library and can be arranged in semantic nets using various types of relations such as "is-a", "component-of", and "members-of".

RDF provides an extensible representation scheme. A software reuse library system must be flexible enough to allow representation schemes to evolve as the needs and level of expertise in an organization increases. The RDF specification language provides several alternatives to extend or adjust a taxonomy so as to allow the incorporation of new objects into the library without having to classify all other objects.

RDF has a powerful similarity-based retrieval mechanism. One essential characteristic of any software library system is to allow the retrieval of candidate reuse components based on partial or incorrect specification. RDF provides a retrieval mechanism that selects candidate components based on the degree of similarity of their associated library descriptions. This mechanism is based on an alternative refinement process in which components at different levels of granularity can be retrieved. It also includes facilities that allow a library designer to customize the retrieval process by including domain specific function.

In short, RDF addresses the main limitations of current faceted classification systems by extending their representation model.

SUMMARY AND FUTURE WORKS

The RDF is a general system for creating, using, and maintaining libraries of object descriptions with the purpose of improving reusability in software and non-software organizations. RDF overcomes the limitations of the actual systems by extending their representation model and incorporating a retrieval mechanism based on asymmetric similarity distances. In summary, we have presented a software reuse library system called RDF and show how its representation model overcome the limitations of current reuse library systems based on faceted representations of objects. Although the RDF reuse system has to be an effective reuse tool, its performance and usefulness can be enhanced. Several areas that need more research were identified:

Domain analysis. In general, to create a library for software reuse it is necessary to perform a domain analysis, the process of identifying, collecting, organizing, analyzing, and representing a domain model and software architecture from the study of existing systems, underlying theory, emerging technology, and development histories within the domain of interest. Domain analysis is currently done by human expert, but several proposals for formalizing and automating this process have been presented in the literature.

Semi-automatic classification. A method is needed to classify components in terms of a given representation model. In a general, this involves analysis of the different parts of a component (e.g., source code, documentation, etc.), and the use of heuristics to extract attributes based on this analysis.

Similarity distances. A method is needed to test whether the reuse candidates proposed by the system are truly best ones available in the software library. For example, if we classify a new component A know to be similar to a previously classified component B, we would expect the library system to propose B as a reuse candidate for A. failure to do this could arise due to errors in classification of components A or B, or because of errors in the definition of relevance factors and/or distance comparators.

REFERENCES

K.J. Anderson, R.P. Beck, and T.E. Buonanno. The full computing reviews classification scheme, Computer review, 29 January 1988.

B.H. Barnes and T.B. Bollinbger. Making reuse cost-effective, IEEE Software Engineering, January 1991, 13-24.

T.J. Biggerstaff and A.J. Perlis. Software reusability, Volume I: Concepts and Models, ACM Press Frontier Series. September 1989, 474-476,.

Z. Houhamdi and S. Ghoul. A Reuse Description Formalism, ACS/IEEE international conference on computer systems and applications AICCSA01, Lebanese American University, Beirut, Lebanon. 2001.

Z. Houhamdi. Describing and Reusing Software Experience. The international Conference on Computer Science, Software Engineering, Information Technology, e-Business, and Applications CSITeA'02. Foz do Iguazu, Brazil, June 2002.

Z. Houhamdi. A Classification Scheme for Software Reuse. SCS/IEEE 2002. The third Middle East Symposium on Simulation and Modelling, MESM'2002, Dubai, Emirate united, September 2002.

Z. Houhamdi. Building and Managing Software Reuse Library. The international Journal of Computing and Informatics Informatica (accepted), 2003.

R. Prieto-Diaz. A Software Classification Scheme, Ph.D. thesis, Department of Information and Computer Science, University of California at Irvine, 1985.

R. Prieto-Diaz. Domain analysis for software reusability, In proceedings of the 11th international Computer Software and applications Conference COMPSA'98. IEEE Computer Society Press, 1987.

R. Prieto-Diaz. Implementing Faceted Classification for software reuse, IEEE Transaction on Software Engineering. 1991, 88-97.

R. Prieto-Diaz and P. Freeman. Classifying software for reusability, IEEE Transaction on Software Engineering, January 1987, 6-16.

**ANALYTICAL
&
STOCHASTIC MODELLING
TECHNIQUES
&
APPLICATIONS**

A Preconditioning Method for Stochastic Automata Networks

ABDEREZAK TOUZENE

Department of Computer Science, Sultan Qaboos University

P.O. Box 36, Al-khod 123, Oman.

email: touzene@squ.edu.om

Abstract

In this paper we extend the methodology presented in [1] to develop a preconditioning method to solve Markovian models issued from Stochastic Automata Network modeling (SAN). This method is also based on grouping terms and factorization of the SAN descriptor. Stochastic automata networks have gained a high interest because of the ease of modeling parallel systems and also because of the compact structure of the SAN generator. In this paper, we propose a preconditioning method that uses the compact structure of the SAN.

Keywords: Markovian models; Stochastic automata networks; Preconditioning.

1 Introduction

Stochastic Automata Networks (SAN) is a very powerful modeling tool for complex systems [2] [3] [4], they are particularly useful to model parallel activities, such as concurrent and communicating processes. A stochastic automata network consists of a collection of automaton that may interact each others. An automaton models a specific activity or a component of the whole system under study. Each automaton is represented by a finite number of states and a set of probabilistic transition rules, which define the moves from a state to another. At any given time t , the global state of a stochastic automata network consists of the current state of each one of its compound automaton. In the general case, the SAN descriptor Q has the following form:

$$Q = \bigoplus_i A_i + \sum_j \otimes_i Q_i^{(j)}, \quad (1)$$

where the first part of Q is called local term, and the second part corresponds to synchronization events see [4]. Computing the steady states probability vector needs to solve the following system of equations:

$$\pi Q = 0. \quad (2)$$

where π is the steady states probability vector of size n and Q the SAN descriptor. Because of the compact form of the descriptor, only iterative methods can be applied to solve the above system of equations. Fortunately, all iterative methods involve the multiplication of the SAN descriptor by a vector which can be efficiently computed without expanding the matrix Q [5]. Iterative methods such as Arnoldi, GMRES [6] [7] can be used in solving the system of equations.

Preconditioning is a very important step for solving the system of equations using iterative methods. This step will improve considerably the convergence rate of the original iterative method. Let us consider the general case, where the system to be solved is

$$Ax = b. \quad (3)$$

Where A is an $n \times n$ matrix and x of size n . The aim of preconditioning is to modify the original system of equations in order to have an equivalent system with a better distribution of its eigenvalues. In fact, it is well known that the convergence of iterative methods and also the stability of direct methods depends on the distribution of the eigenvalues of the system [8] [9]. Preconditioning the system (3) leads to the modified system

$$M^{-1}Ax = M^{-1}b. \quad (4)$$

The matrix M^{-1} is called the preconditioner and should approximate the inverse of the matrix A . The problem is how to compute the matrix M^{-1} in an inexpensive way. Traditional approaches to calculate the matrix M^{-1} is to obtain an incomplete factorization of the matrix A ,

$$A = LU + E \quad (5)$$

where L is a lower triangular matrix, U is an upper triangular matrix, and E is the error or a residual matrix. For more details on how the factorization is performed ($ILU(0)$, $ILU(k)$, $ILUTH$) see [9].

Let us focus now on the SAN descriptor, which has a tensor form. It is clear that the traditional methods cannot be applied to build a preconditioner. We will solve the SAN descriptor using the

system of equations (2). The system of equations (2) is equivalent to

$$\pi P = \pi, \quad (6)$$

where $P = (I + Q)$, and I is the identity matrix of dimension n . This system can be solved using an iterative method. Preconditioning the system (6) leads to the following iterative formulation:

$$\pi^{(t+1)} = \pi^t (I - (I - P)M^{-1}). \quad (7)$$

And then,

$$\pi^{(t+1)} = \pi^t - \pi^t (I - P)M^{-1}, \quad (8)$$

where M^{-1} is an approximation of the inverse of the matrix

$M = (I - P)$. The problem is the computation of this inverse taking into account the particular structure of the matrix $(I - P)$. In [10], the inverse of $(I - P)$ is computed as a polynomial series,

$$M^{-1} = \sum_{k=0}^K P^k, \quad (9)$$

where K is a given order. This Preconditioner gave a good convergence rate, but unfortunately, the cost of powering of P induces a huge amount of time. In this paper we propose a method which avoid the direct computation of the inverse M^{-1} and transforms this problem to solve a linear system of equations [1]. Indeed, if we denote $y^t = \pi^t (I - P)$, the result x^t of multiplying y^t by the matrix M^{-1} can be provided by solving the following system of equations:

$$x^t M = y^t, \quad (10)$$

where M is kept in its compact form (tensor sums).

Our paper is organized as follows: In section 2 we present the methodology. Section 3 describes the SAN models to be tested. In section 4, we summarize our results and we conclude in section 5.

2 Methodology

To simplify the presentation of our method, we will present first the case where the SAN descriptor is a pure tensor sum (it contains only local terms). Then we will show how to extend the method to the case where the SAN is not a pure tensor sum, which is in fact the more general case.

2.1 SAN Descriptor with Pure Tensor Sums

For clarity reasons, let us consider the case where the SAN descriptor consists only on tensor sum with only two factors : $M = (A_1 \oplus A_2)$, where A_1 is of

dimension $m \times m$, and A_2 is of dimension $n \times n$. The generalization of this method is very simple and will be discussed later on. We recall that our aim is to solve the system of equations (10). As described in the basic algorithm of Bartels and Stewart [11], the system (10) is equivalent to the system:

$$A_1^T X + X A_2 = Y \quad (11)$$

where X is an $m \times n$ matrix. The matrix X can be seen as a matrix of columns : $X = (x_1, x_2, \dots, x_n)$, where x_j denotes the j th column of X . The matrix Y which represents y_t is structured in a similar way. The algorithm is as follows:

Algorithm

1. Compute a unitary transformation U and V such that

$$\bar{A}_1 = U^T A_1 U \quad \text{and} \quad \bar{A}_2 = V^T A_2 V, \quad (12)$$

where \bar{A}_1 and \bar{A}_2 are upper triangular matrices obtained using a QR algorithm.

2. Transform the original system (11) using step 1 as follows:

$$\bar{A}_1^T \bar{X} + \bar{X} \bar{A}_2 = \bar{Y} \quad (13)$$

where

$$\bar{X} = U^T X V$$

and

$$\bar{Y} = U^T Y V.$$

3. The first column of \bar{X} is given by solving the following lower triangular system:

$$(\bar{A}_1^T + \bar{A}_{211} * I) \bar{x}_1 = \bar{y}_1. \quad (14)$$

4. The computation of the k^{th} column of the system is given by:

$$(\bar{A}_1^T + \bar{A}_{2kk} * I) \bar{x}_k = \bar{y}_k - \sum_{i=1}^{k-1} \bar{A}_{2ik} * \bar{x}_i. \quad (15)$$

5. Form the solution X using \bar{X} : $X = U \bar{X} V^T$.

The generalization of this method is based on a recursive solving process using the previous steps as follows:

Any system $x_t (A_1 \oplus A_2 \dots \oplus A_N) = y_t$, where the matrices A_i are of dimension $n_i, i = 1..N$, can be decomposed as $x_t (A \oplus A_N) = y_t$. This leads to solve the system of equations (10) including the matrix A which is a tensor sum. We apply the recursion till

reaching a matrix with only two matrices. It is easy to see that the cost of solving the above system of equations (excluding the QR decomposition of the matrices) is given by

$$Cost = \frac{5}{2}(n_1 n_2 \dots n_N)(n_1 + n_2 + \dots + n_N).$$

In general, matrices $A_i, i = 1..N$ are small matrices. The cost of their QR decomposition is negligible and hopefully they are calculated only one time for all iterations. The cost to solve the preconditioned system is then equal to 2.5 time the cost of the multiplication vector by the tensor sum.

Let us focus now on practical implementation issues. Since M is singular (generator), the inverse of M does not exist. The idea is to alter slightly the system (13) to overcome the singularity problem. One sure way to alter M lies within the following property of the tensor sum: $(A_1^T + \alpha I)X + X(A_2 - \alpha I) = Y$, for any real number α . The resulting system is equivalent to the first one, but the singularity is still present. The second way is to apply a shift to $M = (A_1 \oplus A_2 + \alpha I)$. The new system is not equivalent, but fortunately any shift combined with the power method converge to the right solution.

2.2 SAN Descriptor with Synchronization Terms

We recall that the SAN descriptor has the form given in formula (1). In general, it has two parts, the local terms, which are pure tensor sums, plus the synchronization terms, which are tensor products. In this case our algorithm cannot be applied directly, since it handles only pure tensor sums. One solution is to consider only the local terms as a preconditioner. This leads to lose the effects of the synchronizations in the preconditioner. We recall that in this case according to formula (5), the residual matrix E of the preconditioner will be the sum of all synchronization terms. A more accurate preconditioner is to add as much as possible the synchronization terms, in such a way that the preconditioner remains a pure tensor sums and in this case the residual matrix E will be of smaller magnitude. Intuitively, we will capture the effect of some synchronizations in the preconditioner. This may be possible only if the synchronization events affect only few automata in the stochastic automata network. This assumption is valid for modeling distributed and parallel systems.

Now we focus on how to add the effects of the synchronization in the preconditioner resulting in a pure tensor form preconditioner: The idea is to group the automata that are affected by the same synchronization events using tensor algebra. This

will result in reducing the number of terms but within the terms, matrices will be of higher size.

In the next section we describe the models to be tested and we will show using a specific example how to group and factorize the preconditioner using the tensor algebra properties in order to have an accurate preconditioner.

3 Examples of SAN Models

3.1 Resource Sharing Problem

In this model, N processes share P resources, $P \leq N$. Each process $i = 1..N$, has two states: *using* the resource or *idle*. We denote λ_i the resource acquisition rate and μ_i the freeing rate of a resource from the process i . Notice that the number of processes which may access concurrently to the resource is limited to P . When a process is willing to acquire a resource and find that P processes have already got the resource, it will fail and stay in its *idle* state. In case $P = 1$, the model is equivalent to the usual mutual exclusion problem. The SAN descriptor will have the following form: $Q = \bigoplus_{i=1}^N A_i$ (pure tensor sums), where A_i is

$$A_i = \begin{pmatrix} -\lambda_i f & \lambda_i f \\ \mu_i & -\mu_i \end{pmatrix}$$

Each automaton contains a functional variable f evaluated to 0 or 1 depending on the availability of a resource or not. For our experiments (Matlab), the descriptor matrix Q is factorized in two factors $Q = Q_1 \oplus Q_2$. Concerning the preconditioner, for simplicity reasons we evaluate all the functions $f = 1$.

3.2 Altered Bit Protocol

This model concerns the well-known network protocol named altered bit protocol. This protocol is modeled by a SAN in [3] as follows: It contains four automata :

- Sender automaton, named S , which have four states namely :PrepareMess0, WaitAck0, PrepareMess1, WaitAck1.
- Message automaton, named M , which have four states namely :WaitMess0, Mess0Inline, WaitMess1, Mess1Inline.
- Acknowledge automaton, named A , which have four states namely :WaitAck0, Ack0Inline, WaitAck1, Ack1Inline.
- Receiver automaton named R , which have four states namely :WaitMess0, PrepareAck0, WaitMess1, PrepareAck1.

The descriptor of this model contains nine terms, one term called local term and eight terms called synchronized terms. The local term consists of the local views of each automaton. It does not contain the interaction between all automata. The synchronization terms represent the effect of a synchronization event on the different automaton of the SAN. For each Synchronization event correspond one term. In the following we construct each term of the descriptor of SAN.

1. The local term consists of the matrices L_S, L_M, L_A and L_R , where

$$L_S = \begin{pmatrix} 0 & 0 & 0 & 0 \\ tm & -tm & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & tm & -tm \end{pmatrix},$$

$$L_M = \begin{pmatrix} 0 & 0 & 0 & 0 \\ lm & -lm & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & lm & -lm \end{pmatrix},$$

$$L_A = \begin{pmatrix} 0 & 0 & 0 & 0 \\ la & -la & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & ta & -ta \end{pmatrix},$$

$$L_R = \begin{pmatrix} -ta & 0 & 0 & ta \\ 0 & 0 & 0 & 0 \\ 0 & ta & -ta & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

$$termloc = L_S \oplus L_M \oplus L_A \oplus L_R.$$

2. The term Send-Message1 contains the matrices $Sm1^{(S)}$ and $Sm1^{(M)}$:

$$TermSendMess1 = Sm1^{(S)} \otimes Sm1^{(M)} \otimes I_4 \otimes I_4.$$

This synchronization affects only the automata S and M .

3. The term Send-Mess0 contains the matrices $Sm0^{(S)}$ and $Sm0^{(M)}$:

$$TermSendMess0 = Sm0^{(S)} \otimes Sm0^{(M)} \otimes I_4 \otimes I_4.$$

This synchronization affects only the automata S and M .

4. The term Send-Ack1 contains matrices $Sack1^{(A)}$ and $Sack1^{(R)}$:

$$TermSendAck1 = I_4 \otimes I_4 \otimes Sack1^{(A)} \otimes Sack1^{(R)}$$

This synchronization affects only the automata A and R .

5. The term Send-Ack0 contains matrices $Sack0^{(A)}$ and $Sack0^{(R)}$:

$$TermSendAck0 = I_4 \otimes I_4 \otimes Sack0^{(A)} \otimes Sack0^{(R)}$$

This synchronization affects only the automata A and R .

6. The term Receive-Mess1 contains the matrices $Srm1^{(M)}$ and $Srm1^{(R)}$

$$TermRecMess1 = I_4 \otimes Srm1^{(M)} \otimes I_4 \otimes Srm1^{(R)}$$

This synchronization affects only the automata M and R .

7. The term Receive-Mess0 contains the matrices $Srm0^{(M)}$ and $Srm0^{(R)}$

$$TermRecMess0 = I_4 \otimes Srm0^{(M)} \otimes I_4 \otimes Srm0^{(R)}$$

This synchronization affects only the automata M and R .

8. The term Receive-Ack1 contains the matrices $Srack1^{(S)}$ and $Srack1^{(A)}$

$$TermRecAck1 = Srack1^{(S)} \otimes I_4 \otimes Srack1^{(A)} \otimes I_4$$

This synchronization affects only the automata S and A .

9. The term Receive-Ack0 contains the matrices $Srack0^{(S)}$ and $Srack0^{(A)}$

$$TermRecAck0 = Srack0^{(S)} \otimes I_4 \otimes Srack0^{(A)} \otimes I_4$$

This synchronization affects only the automata S and A .

The SAN descriptor of this model is the sum of all the above terms. For our preconditioner, we will factorize and group as much term as possible in order to obtain the preconditioning matrix M of the form $M = Q_1 \otimes Q_2$ and hence our method will be directly applicable. First, let us group all synchronized terms that have a synchronized effect on the same automata. In the second step, add them if possible to the pure tensor sum using some simple tensor factorization operations (for more details on tensor algebra see [12]). According to the effect of the synchronization on the automata network as described above, we group automata S, M to form a macro-automaton and we group automata A, R to form another macro-automaton. This will result in the following:

$$Q_1 = (L_S \oplus L_M) + (Sm1^{(S)} \otimes Sm1^{(M)}) \\ + (Sm0^{(S)} \otimes Sm0^{(M)})$$

and

$$Q_2 = (L_A \oplus L_R) + (SAck1^{(A)} \otimes SAck1^{(R)}) \\ + (SAck0^{(A)} \otimes SAck0^{(R)}).$$

The other terms:

$$E = TermRecMess1 + TermRecMess0 + \\ TermRecAck1 + TermRecAck0,$$

cannot be added to the preconditioner (without a loss of its pure tensor sums property), their effect will be lost. We may think of them as a residual matrix of an incomplete factorization see equation (5), and we hope that their effect is very small.

4 Experimental Results

For testing our preconditioning method, we chose for simplicity reasons the power iterative method. In our experiments we considered a shift $\alpha = 0.25$ to ensure the non-singularity of the preconditioner. The following tables summarizes the number of iterations using the power method and the preconditioned power method. The solution precision is 10^{-10} decimals. For the first table, all the parameters $\lambda_i, i = 1..8$ are equal to a given value λ and similarly $\mu_i = \mu, i = 1..8$. We recall that P is the number of resources.

P	Param.	Iter. Power	Iter. Precond.
1	$\lambda = 1, \mu = .5$	18	10
	$\lambda = 1, \mu = .9$	13	3
4	$\lambda = 1, \mu = .5$	91	19
	$\lambda = 1, \mu = .9$	109	25
6	$\lambda = 1, \mu = .5$	77	22
	$\lambda = 1, \mu = .9$	319	30
8	$\lambda = 1, \mu = .5$	97	32
	$\lambda = 1, \mu = .9$	196	33

The second experimental study concerns the model of the alterned bit protocol. For this model we consider four cases as follows:

- 1: $tm = 2, lm = 2, la = .1, ta = .1, us = 2, ur = 1, ue = 2, uc = 1.$
- 2: $tm = 4, lm = 4, la = .1, ta = .1, us = 4, ur = 1, ue = 4, uc = 1.$
- 3: $tm = .1, lm = .1, la = .1, ta = .1, us = .1, ur = 1, ue = .1, uc = 1.$
- 4: $tm = .1, lm = .1, la = .1, ta = .1, us = .1, ur = .1, ue = .1, uc = 1.$

us is a transition rate in the matrix $Sm0^{(S)}$ and $Sm1^{(S)}$. ur is a transition rate in the matrix $Srm0^{(R)}$ and $Srm1^{(R)}$. ue is a transition rate in the matrix $Srack0^{(R)}$ and $Srack1^{(R)}$. uc is a transition rate in the matrix $Srack0^{(S)}$ and $Srack1^{(S)}$.

Case.	Iter. Power	Iter. Precond.
1	1240	185
2	2189	293
3	77	22
4	787	146

These two tables show clearly the good performance of our preconditioning method. Concerning the number of iterations to achieve convergence, this method is nine time faster than the simple power method. If we consider the time cost, our method will be almost 4 times faster than the simple power method.

5 Conclusion

In this paper we contributed in developing a preconditioning method for solving stochastic automata network models. This method has shown a very good results on the tested models and it can be easily adapted to other iterative methods, known to converge very quickly such as the iterative GMRES for SAN models. In our future work we will investigate a general and automatic procedure for grouping automata (affected by the same synchronization events) and factoring the preconditioner as a pure tensor sum with the smallest residual matrix possible.

Acknowledgment

The author thanks Prof. B. Plateau and Prof. W. J. Stewart for their help and guidance in conducting this work.

References

- [1] G.W Stewart. Stochastic Automata, Tensors Operation, and Matrix Formulas. *Technical Report, University of Maryland, UMIACS TR-96-11 CMSC TR-3598, Jan. 1996.*
- [2] B. Plateau. On the Stochastic Structure of Parallelism and Synchronization Models for Distributed Algorithms. *Proc. ACM Sigmatics Conference on Measurement and Modeling of Computer Systems, Austin, Texas, Aug. 1985.*
- [3] K. Atif. Modelisation du Parallelism et de la Synchronisation. *These de Docteur de l'Institut National Polytechnique de Grenoble, France, Sep. 1991.*

- [4] K. Atif, B. Plateau. Stochastic Automata Network for Modeling Parallel Systems. *IEEE Trans. On Soft. Eng.*, 17, 10, Oct. 1991 .
- [5] P. Fernandes, B. Plateau, W. J. Stewart. Efficient descriptor-vector multiplication in stochastic automata networks. *J. ACM*, (3), 381-414, 1998.
- [6] Y. Saad. Krylov Subspace Methods for Solving Unsymmetric Linear Systems. *Mathematics of Computation*, 37, 105-126.
- [7] Y. Saad, M.H. Shultz. GMRES: A generalized minimal residual algorithm for solving non-symmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7, 856-869, 1986 .
- [8] R. S. Varga. Matrix Iterative Analysis. *Printice Hall, Englewood Cliffs, N.J*, 1963.
- [9] W. J. Stewart An Introduction to the Numerical Solution of Markov Chains. *Princeton University Press, NJ*. 1994.
- [10] K. Atif, B. Plateau, and W. Stewart. The numerical solution of stochastic automata network. *European Journal of Operation Research*, 86(3), Nov. 1995.
- [11] R. H. Bartels and G. W. Stewart. Algorithm 432: The solution of the matrix equation $AX-BX=C$. *Communication of ACM*, 8:820-826, 1972.
- [12] M. Davio. Kronecker Products and Shuffle Algebra. *IEEE Trans. Comp.*, Vol C-30, No 2, Feb. 1981.

Biography

ABDEREZAK TOUZENE received an undergraduate degree in computer science from university USTHB of Algiers in 1987 ,M.Sc. from university Paris sud in 1988 and a Ph.D. degree in computer science from Institut Polytechnique de Grenoble (France) in 1992. Dr. Touzene is currently assistant professor in the department of computer science, Soltan Qaboos university in Oman. His area of interest include performance evaluation, interconnection networks and parallel computing. He is a member of IEEE.

HIERARCHICAL STOCHASTIC ACTIVITY NETWORKS

MOHAMMAD ABDOLLAHI AZGOMI AND ALI MOVAGHAR

*Department of computer engineering,
Sharif university of technology,
Tehran 11365, Iran.*

E-mail: azgomi@mehr.sharif.edu and movaghar@sharif.edu

Abstract: *Stochastic activity networks* (SANs) are a powerful and flexible extension of Petri nets. These models can be used for the modeling and analysis of various kinds and different aspects of distributed real-time systems. Similar to other classical extensions of Petri nets, SANs have some limitations for modeling complex and large-scale systems. In order to remove these limitations and provide some high-level modeling facilities, we have defined a new extension for SANs, which is called *hierarchical stochastic activity networks* (HSANs). HSAN models provide facilities for composing a hierarchy of submodels, incremental modeling and the reusability of submodels. HSAN models encapsulate hierarchies and a key benefit of these models is the possibility of automatic employment of composition techniques by their modeling tool. In this paper, we present the graphical notations, formal definition, and methods for the analysis of HSAN models. We also present an example of the application of these models in the design of a high-capacity packet-based telecommunication switch.

keywords: Stochastic Petri Nets, Stochastic Activity Networks, High-Level Petri Nets, Hierarchical Modeling.

1. INTRODUCTION

Stochastic activity networks (SANs) [Movaghar and Meyer, 1984] are a stochastic generalization of Petri nets (PNs). These models are more powerful and flexible than most other stochastic extensions of Petri nets such as *stochastic Petri nets* (SPNs) [Molloy, 1982] and *generalized stochastic Petri nets* (GSPNs) [Marsan, Balbo and Conte 1986]. SANs permit the representation of concurrency, timeliness, fault-tolerance and degradable performance in a single model [Movaghar, 1985].

Similar to other classical extensions of Petri nets, SANs have some limitations for modeling complex and large-scale systems, such as the lack of compositionality, incremental modeling and the reusability of submodels. In order to remove these limitations, we have defined a new extension for SANs called *hierarchical stochastic activity networks* (HSANs). HSAN models encapsulate hierarchies and a key benefit of these models is the possibility of automatic employment of composition techniques, such as Replicate/Join [Sanders and Meyer, 1991] or Graph Composition [Stillman, 1999] formalisms, by their modeling tools.

In this paper, we present properties, graphical notation, formal definition, and applications of HSANs. The paper is organized as follows. In Sec. 2, some limitations of SANs and related works on hierarchical PNs are presented. In Sec. 3, the informal and formal definitions and graphical notations of HSANs are presented. Methods for the transformation and analysis of HSANs will be introduced in Sec. 4. And, in Sec. 5, an example of the application of these models is presented.

2. MOTIVATION

2.1 Limitations of SANs

SAN models have been employed in a lot of applications and are supported with a few powerful modeling tools such as *UltraSAN* [Sanders et al, 1995] and *Mobius* [Deavours et al, 2002]. But, both the ordinary definition [Movaghar and Meyer, 1984] and the new definition of SANs [Movaghar, 2001] have the following limitations:

1. *SANs are rather flat:* People manage complexity by having hierarchies. Programming languages, especially object-oriented languages, have simple, intuitive ways of encapsulating hierarchies [Deavours, 2003]. The Replicate/Join formalism has been proposed to alleviate this problem. Unfortunately, There is a set of issues with this construct [Deavours, 2003]. It is limited to a tree-like structure, and an arbitrary symmetric model structure, such as ring or mesh, cannot be expressed with this construct. However, the use of the Replicate/Join or the Graph Composition formalisms is specific to modeling tools and not SANs in general.
2. *The lack of facilities for compositionality:* There is no facility in the definition of SANs for constructing models for complex systems with small and simple submodels.
3. *The lack of facilities for reusability:* Since SAN models are flat, using a part of an existing model as a component for constructing a new one is difficult.
4. *The lack of facilities for incremental modeling:* There is a lack of facilities for constructing

complex models incrementally, by starting with abstract components and easily replacing them with detailed and enhanced components.

2.2 Related Works

There are a few numbers of extensions for PNs, which provide facilities for composing hierarchical models. Some of them are as follows:

- *Hierarchical Petri nets (HPNs)*: The absence of compositionality has been one of the main critiques raised against Petri net models [Jensen, 1990]. Because, PN models are a flat network of *places* and *transitions*. To remove this drawback, *hierarchical Petri nets (HPNs)* were introduced. Hierarchical Petri nets are a structural method of describing concurrent processes. They enable to design more complex systems through abstracting some parts of the net.
- *Hierarchical coloured Petri nets (HCPNs)*: Hierarchical extensions have also been introduced for coloured Petri nets (CPNs) [Jensen, 1990]. *Hierarchical coloured Petri nets (HCPNs)* provide facilities for constructing a hierarchy of CPNs [Huber, Jensen and Shapiro, 1990]. In HCPNs, CPN models can be structured into *pages* (subnets) using *substitution transitions*. HCPNs also offer the concept of *place fusion*. *Fusion places* make it possible to specify that a set of places represent a single conceptual place, which has been drawn as several copies. When tokens are removed or added to a fusion place all places in the fusion set will have their marking correspondingly changed [Lakos, 1995].
- *Modular coloured Petri nets (MCPNs)*: Allows invariant analysis in the context of modular nets incorporating both place and *transition fusion* [Christensen and Petrucci, 1992]. MCPNs therefore extend hierarchical coloured Petri nets (HCPNs) by supporting the notion of *substitution places* as well as *substitution transitions*. These constructs are also referred to as *super places* and *super transitions*, respectively.
- *Replicate/Join construct* [Sanders and Meyer, 1991]: This construct has been introduced to provide model composition in the *UltraSAN* modeling tool. Replicate and Join operations are used to compose models by sharing states. *Subnet* of SAN model is the unit of composition. These subnets can be composed in a hierarchical manner to construct a SAN model. This approach has been extended and generalized in the work of [Stillman, 1999] on the *Graph Composition* formalism in the *Mobius modeling framework* [Deavours, 2001].

3. Definitions of HSANs

To remove some limitations of SANs for modeling and analysis of complex and large-scale systems, we introduced *hierarchical stochastic activity networks (HSANs)*. These models provide constructs to build a hierarchy of submodels rather than viewing SANs as a flat collection of primitives. The structure of the composition in HSANs is a hierarchy of submodels. Each level of the hierarchy represents a different level of abstraction. This will facilitate the comprehension of the whole model of a large system.

3.1 The Elements of HSAN Models

An HSAN model is composed of five primitives of the ordinary SANs, (including, *place*, *input gate*, *output gate*, *instantaneous activity*, *timed activity*) and *macro activity (MA)*. MA is an HSAN submodel, which is composed of a finite number of SAN primitives and lower-level macro activities.

Place fusion provides a mechanism for interfacing macro activities to other parts of an HSAN model. A MA may have zero or more *input* and *output fusion places*. Fusion places are a subset of normal places. A MA has a well-defined interface, which is similar to high-level programming languages. The places surrounding a MA are *formal fusion places*. When a MA is used in an HSAN model, these formal places will be bound by *actual places*, which are normal places of SANs. This relation is similar to formal parameters of a procedure, which will be bound by actual parameters of the caller program in high-level programming languages. A formal fusion place has always the same marking as the related actual place. The two places are different views of the same place.

3.2 Graphical Notation

In graphical representation of HSANs, a fusion place is depicted as \odot . A graphical representation of a MA has been depicted in Fig. 1. In this figure, a MA is drawn as a rectangle, which is surrounded by input and output fusion places. An example of an HSAN model is presented in Fig. 2 and the definition of *TMAI* in Fig. 3.

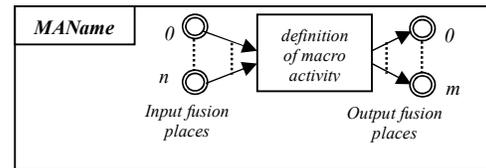
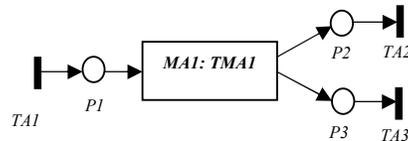


Fig. 1. Definition of macro activity

Fig. 2. An example of the usage of a MA named *TMAI*



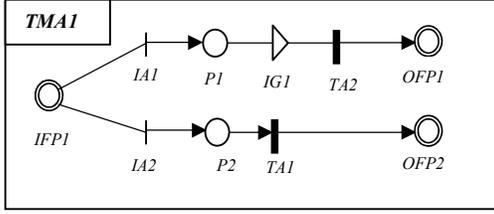


Fig. 3. Definition of a macro activity named *TMA1*

3.3 Some Rules and Properties

There are the following rules and properties on HSAN models:

1. In order to interface with other models or submodels, a MA should have at least one input fusion place or one output fusion place. The maximum number of input or output fusion places is not limited.
2. Inside a MA, all input fusion places are *read-only* and all output fusion places are *write-only*.
3. All predicates and functions of a MA are dependent to the marking of the local or fusion places.
4. A MA can be used in the hierarchy of an HSAN model more than once and with different names.
5. In each level of the hierarchy of an HSAN model, naming is local. For example, in Fig. 2, there are three timed activities *TA1*, *TA2* and *TA3*. In Fig. 3, which is the definition of *TMA1*, there are other timed activities named *TA1* and *TA2*. However, in the same level (root HSAN or in each MA), names should be unique.
6. MAs can be defined separately. Previously defined MAs may be used to compose a new HSAN model.

3.4 Formal Definition of HSANs

Formal definition of HSANs is the basis for analytic solution, simulation, and for the formal verification over state spaces. Based on these definitions and well-defined syntax and semantics, HSAN tools will facilitate the process of constructing HSAN models. Modeling tools for HSAN will automatically check for the correct use of these syntax and semantics.

Now, we formally define *hierarchical stochastic activity networks (HSANs)*, based on a new definition of SANS [Movaghar, 2001]. In the following definitions, N denotes the set of natural numbers and R_+ represents the set of non-negative real numbers.

Definition 1. *Hierarchical stochastic activity network (HSAN)* is defined as a 12-tuple $HSAN = (P, IA, TA, MA, IG, OG, IR, OR, C, F, \Pi, \rho)$ where:

- P is a finite set of *places*,
- IA is a finite set of *instantaneous activities*,
- TA is a finite set of *timed activities*,
- MA is a finite set of *macro activities*, which will be defined later,
- IG is a finite set of *input gates*. Each input gate has a finite number of inputs. To each $G \in IG$, with m inputs, is associated a *function* $f_G : N^m \rightarrow N^m$, called the function of G , and a *predicate* $g_G : N^m \rightarrow \{true, false\}$, called the *enabling predicate* of G ,
- OG is a finite set of *output gates*. Each output gate has a finite number of outputs. To each $G \in OG$, with m outputs, is associated a *function* $f_G : N^m \rightarrow N^m$, called the function of G ,
- $IR \subseteq P \times \{1, \dots, |P|\} \times IG \times (IA \cup TA \cup MA)$ is the *input relation*. IR satisfies the following conditions:
 - for any $(P_1, i, G, a) \in IR$ such that G has m inputs, $i \leq m$,
 - for any $G \in IG$ with m inputs and $i \in N$, $i \leq m$, there exist $a \in (IA \cup TA \cup MA)$ and $P_1 \in P$ such that $(P_1, i, G, a) \in IR$,
 - for any $(P_1, i, G_1, a), (P_1, j, G_2, a) \in IR$, $i = j$ and $G_1 = G_2$,
- $OR \subseteq (IA \cup TA \cup MA) \times OG \times \{1, \dots, |P|\} \times P$ is the *output relation*. OR satisfies the following conditions:
 - for any $(a, i, G, P_1) \in OR$ such that G has m outputs, $i \leq m$,
 - for any $G \in OG$ with m outputs and $i \in N$, $i \leq m$, there exist $a \in (IA \cup TA \cup MA)$ and $P_1 \in P$ such that $(a, G, i, P_1) \in OR$,
 - for any $(a, G_1, i, P_1), (a, G_2, j, P_1) \in OR$, $i = j$ and $G_1 = G_2$,
- $C : N^n \times IA \rightarrow [0, 1]$ is the *case probability function*, where $n = |P|$.
- $F = \{F(\cdot, \mu, a); \mu \in N^n, a \in TA\}$ is the set of *activity time distribution functions*, where $n = |P|$ and, for any $\mu \in N^n$, and $a \in TA$, $F(\cdot, \mu, a)$ is a probability distribution function,
- $\Pi : N^n \times TA \rightarrow \{true, false\}$ is the *reactivation predicate*, where n is defined as before,
- $\rho : N^n \times TA \rightarrow R_+$ is the *enabling rate function*, where n is defined as before.

Definition 2. Macro activity (*MA*) is defined as a 14-tuple $MA = (P, IA, TA, MA, IG, OG, IR, OR, C, F, \Pi, \rho, IFP, OFP)$ where:

- $P, IA, TA, MA, IG, OG, IR, OR, C, F, \Pi,$ and ρ are defined as before,
- $IFP \subseteq P$ is a set of *input fusion places*,
- $OFP \subseteq P$ is a set of *output fusion places*,
- $|IFP \cup OFP| > 0.$

4. ANALYSIS OF HSAN MODELS

An HSAN model is solvable by analytic or simulative methods, *iff* the dependency graph of the HSAN model is *acyclic*. In this graph, nodes are macro activities and the composition relation between MAs and their parents determines arcs. The root of this graph is the HSAN model.

If the dependency graph is acyclic, it is possible to employ an algorithm to transform an HSAN model to an equivalent flat SAN model. Also, it is possible to automatically employ methods for the analysis of composed models. Some of these techniques are described in this section.

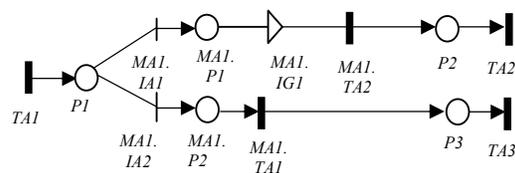
4.1 Transformation to a Flat HSAN Model

Each HSAN model has a behaviorally equivalent SAN model. For the analysis of an HSAN model, it is possible to transform it to a behaviorally equivalent SAN model. A step-by-step substitution algorithm does transformation. In each step, MAs on leaves of the graph will be substituted by their implementations. This will be repeated until the only node on the graph is the root node.

As we mentioned before, naming in HSAN models is local. To resolve the problem of duplicate names in substitution, the name of each element of a MA will be preceded by the name of its parent MA. For example, in transformation of HSAN model of Fig. 2, which is shown in Fig. 4, $IA1, IA2, P1, P2, IG1, IG2, TA1, TA2$ are preceded by $MA1$ and a dot. Also, all marking dependent predicates and functions of MAs should be substituted.

The resulting model is a flat SAN model, without any MA. If the resulting model is Markovian, the reachability graph for this model can be generated and methods for solution of Markov chains can be used to solve it. If the model is not Markovian, the simulation method may be employed.

Fig. 4. An equivalent SAN model for the model of Fig. 2.



4.2 Using Replicate/Join Construct

The main disadvantage of the ad-hoc transformation of HSAN models to flat SANs is the explosion of state-space. There are a few techniques for constructing SAN models in a way, which avoid state-space explosion problem. One of these techniques is the Replicate/Join construct [Sanders and Meyer, 1991], which we briefly described in Sec. 2.2.

A key benefit of HSAN models is the possibility of automatic employment of composition formalisms by modeling tools. Therefore, a possible way of the analysis of HSAN models is their automatic transformation to a Replicate/Join construct.

For this purpose, the modeling tool can check for a tree-like structure in the dependency graph of an HSAN model. After that, it can automatically find shared states based on the fusion places and organize the model using Replicate and Join operations. For the solution of the resulting model, the technique proposed in [Sanders and Meyer, 1991] can be employed.

The model of a telecom switch, which is presented in the next section, is an example of such kind of HSAN models.

4.3 Using Graph Composition Formalism

Another composition formalism, which has been proposed in the Mobius modeling framework for SANs and other models, is the Graph Composition formalism [Stillman, 1999]. This formalism does not limit the model hierarchy to a tree-like structure and any arbitrary is possible.

A modeling tool for HSANs can also check for the possibility of the use of this formalism. Then, it can automatically transform the HSAN model to this construct and then use its method of solution, which is proposed in [Stillman, 1999].

4.4 Simulation of HSAN Models

If an HSAN model or one of its macro activities includes non-exponential timed activities or its state-space is infinite, it may not be solved analytically. In such cases, discrete-event simulation may be employed to solve the model.

For the efficient simulation of HSAN models, methods proposed for SANs, such as procedures presented in [Sanders and Freire, 1993], can be used. These methods are used with Replicate/Join formalism. Therefore, after the transformation of HSAN model to this construct, these methods of simulation may be employed.

If an HSAN model is transformed to the Graph Composition formalism, the methods of simulation proposed in the Mobius modeling framework can be used.

5. AN HSAN MODEL FOR A PACKET-BASED TELECOM SWITCH

In this section, we present an HSAN model for a high-capacity telecommunication switch (HCTS). In contrast to traditional class 5 TDM switches, HCTS is a packet-based switch based on gigabit Ethernet technology. In this switch, analog voice signals will be converted to voice packets in line-cards and will be switched by Ethernet switches to output ports. We have constructed a model for this switch in three levels: *card level*, *unit level*, and *rack level*. The purpose of this modeling is to show the application of HSANs in this area.

HCTS has a modular structure, including one *control rack* (CR) and a few numbers of *subscriber/trunk racks* (STRs). CR includes a *control unit* (CU) and a few numbers of *subscriber units* (SUs) or *trunk units* (TUs). In CU, there is a two modules redundancy (TMR) of *control and switching module card* (CSMC), which executes *call-processor* program of the switch. An HCTS can operate if at least one of these two modules is working properly. STRs include a few number of SU or TU. In this example, we concentrate on modeling SUs, which include line-cards. The goal of this modeling is to evaluate some design parameters, which are related to the *quality of service* (QoS) of the switch to the analog/digital subscribers. Three important parameters in our study are the *blocking probability*, the *maximum delay of voice packets in switching system* (from line-cards to output ports), and the *optimal size of input buffers* of switches in different parts of the system.

Each STR includes 6 SUs. The connection of these units to the whole system is through a gigabit Ethernet switch multiplexer. Each SU includes 19 *line-cards* (LCs). Star topology is used to connect LCs to an Ethernet switch multiplexer with 100 *Mbps* input ports and 1 *Gbps* output port. Each LC includes 30 subscriber ports. LC packetizes 4 *ms* analog voice signals to 84 bytes packet. Output multiplexer of LC is a 100 *Mbps* switch.

A simplified HSAN model for an HCTS is presented in Fig. 5. This model is composed of n rack macro activities (*rma*), input buffer places (*ibs*), and Ethernet switching-hub macro activity (*eshma*). In Fig. 6, *rma* is depicted, which is composed of 6 unit macro activities (*uma*), *ibs*, *esma* and the output buffer fusion place (*ob*). In Fig. 7, *uma* is depicted, which is composed of 19 line-card macro activities (*lcma*), *ibs*, Ethernet switch macro activity (*esma*) and *ob*. In Fig. 8, *lcma* is depicted. An *lcma* is composed of 30 subscriber port macro activities (*spma*), *ibs*, *esma* and *ob*. In Fig. 9, *spma* is depicted. For this purpose, *off-hook* timed activity models the process of hooking off a phone set. *release* timed activity

models the holding time. *busy* is a place, which presents the busy state of line. *isbusy* is an input gate for *packetize* timed activity. *isbusy* checks whether the line is busy or not. *packetize* models the conversion of 4 *ms* analog voice signals to 84 bytes packets by digital signal processor (DSP) of LC. *ob* is an output fusion place of *spma*. Finally, in Fig. 10, *esma* is a simple model for an Ethernet switch multiplexer with n input ports and a single output port. The model for an Ethernet switching-hub (*eshma*) is similar to *esma*.

Fig. 5. HSAN model of an HCTS (top level)

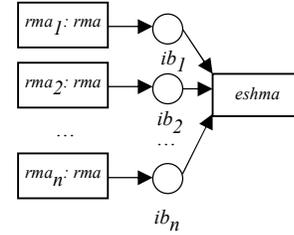


Fig. 6. Rack macro activity (*rma*)

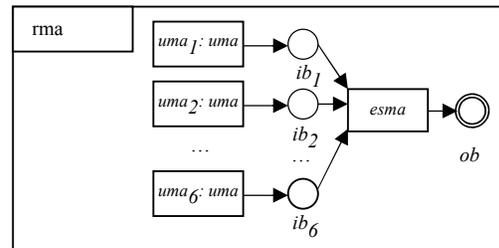


Fig. 7. Unit macro activity (*uma*)

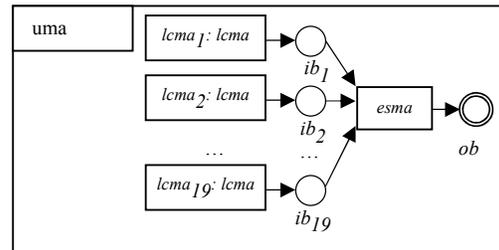


Fig. 8. Line-card macro activity (*lcma*)

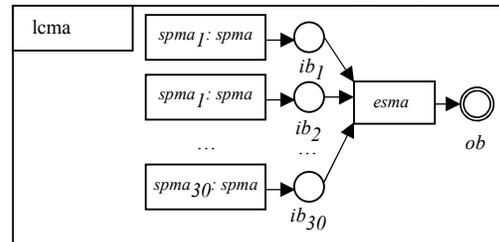
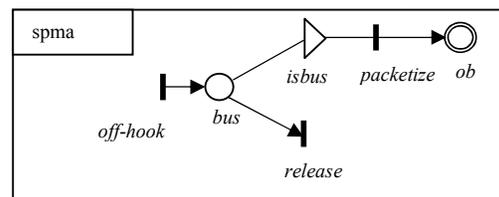


Fig. 9. Subscriber port macro activity (*spma*)



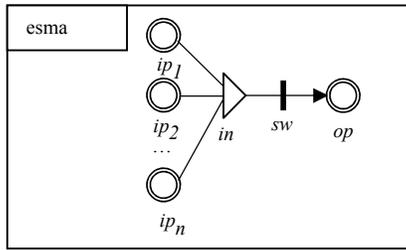


Fig. 10. Ethernet switch macro activity (*esma*)

6. TOOL SUPPORT

For modeling and analysis with HSANs, we have developed a modeling tool called *SANBuilder*. This tool is an enhanced and completely redesigned version of our existing tool for SANs called *SharifSAN* [Abdollahi and Movaghar, 2001]. *SANBuilder* enables modelers to construct HSAN models in a graphical editor, define and store separate macro activities and retrieve them to reuse in a new HSAN model. It also provides features for interactive simulation (i.e. token-game animation), automatic simulation, and analytic solution of HSAN models.

7. CONCLUDING REMARKS

In this paper, we introduced *hierarchical stochastic activity networks* (HSANs). HSAN models provide facilities for composing a hierarchy of submodels, incremental modeling and the reusability of submodels. HSANs have the same power as ordinary SANs, but HSAN models are more flexible and high-level. HSANs have formal definition and well-defined syntax and semantics. HSAN models encapsulate hierarchies and a key benefit of these models is the possibility of automatic employment of composition techniques, such as Replicate/Join or Graph Composition formalisms.

Similar to ordinary SANs, HSANs can be used for the modeling and analysis of various kinds and different aspects of computer and communication systems. HSANs can be used to build both Markovian and non-Markovian models, and have nondeterministic, probabilistic, and stochastic settings. For functional analysis (i.e., verification), the nondeterministic setting of HSAN models and a method for specifying the properties of system (e.g. *temporal logic*) may be employed. For the analysis of operational aspects (i.e., performance, dependability or performability evaluation), the stochastic setting of these models and methods for their steady state or transient analytic solution or simulation can be used.

We have developed *SANBuilder* tool for the modeling and analysis of HSAN models. To make this tool more useful for the application on large-scale systems, we are working on methods for efficient solution and simulation of HSAN models.

REFERENCES

- Abdollahi Azgomi M. and Movaghar A. 2001, "SharifSAN: A Tool for Verification and Performance Evaluation Based on a New Definition of SANs," In *Proc. of the 13th IASTED Int. Conf. on Parallel and Distributed Computing and Systems (PDCS'01)*, Anaheim, CA, Pp667-672.
- Christensen S. and Petrucci L. 1992, "Towards a Modular Analysis of Coloured Petri Nets," In *Proc. of Int. Conf. on Application and Theory of Petri Nets (LNCS 616)*, K. Jensen (ed.), Springer-Verlag, Pp113-33.
- Deavours D.D. 2001, *Formal Specification of The Mobius Modeling Framework*, Ph.D. Dissertation, University of Illinois at Urbana-Champaign.
- Deavours D.D. 2003, Personal Correspondence, Apr 2003.
- Deavours, D.D. et al 2002, "The Mobius Framework and Its Implementation," *IEEE Trans. on Soft. Eng.*, Vol. 28, No. 10, Pp956-969.
- Huber P., Jensen K. and Shapiro R.M. 1990, "Hierarchies of Coloured Petri Nets," In *Proc. of 10th Int. Conf. on Application and Theory of Petri Nets (LNCS 483)*, Springer-Verlag, Pp313-341.
- Jensen K. 1990, *Coloured Petri Nets: A High Level Language for System Design and Analysis (LNCS 483)*, Springer-Verlag.
- Lakos C.A. 1995, "From Coloured Petri Nets to Object Petri Nets," In *Proc. of 16th Int. Conf. on the Application and Theory of Petri Nets (LNCS 935)*, Torino, Italy, Springer-Verlag, Pp278-297.
- Marsan M.A., Balbo G. and Conte G. 1986, *Performance Models of Multiprocessor Systems*, MIT Press.
- Molloy M.K. 1982, "Performance Analysis Using Stochastic Petri Nets," *IEEE Trans. on Computers*, C-31, Pp913-917.
- Movaghar A. and Meyer J.F. 1984, "Performability Modeling with Stochastic Activity Networks," In *Proc. of the 1984 Real-Time Systems Symp.*, Austin, TX, USA, Pp215-224.
- Movaghar A. 1985, *Performability Modeling with Stochastic Activity Networks*, Ph.D. Dissertation, University of Michigan.
- Movaghar A. 2001, "Stochastic Activity Networks: A New Definition and Some Properties," *Scientia Iranica*, Vol. 8, No. 4, Pp303-311.
- Sanders W.H. and Freire R.S. 1993, "Efficient Simulation of Hierarchical Stochastic Activity Network Models," *Discrete Event Dynamic Systems: Theory and App.*, Vol. 3, no. 2/3, Pp271-300.
- Sanders W.H. and Meyer J.F. 1991, "Reduced Base Model Construction Methods for Stochastic Activity Networks," *IEEE J. on Selected Areas in Comm.*, Special Issue on Computer-Aided Modeling, Analysis, and Design of Comm. Net., Vol. 9, No. 1, Pp25-36.
- Sanders W.H. et al 1995, "The UltraSAN Modeling Environment," *Performance Evaluation*, Vol. 24, Pp1-33.
- Stillman A.J. 1999, *Model Composition in the Mobius Modeling Framework*, M.S. Thesis, University of Illinois at Urbana-Champaign.

AN $M^X/G/1$ RETRIAL QUEUE WITH UNRELIABLE SERVER AND VACATIONS

AMAR AISSANI

*Department of Computer Sciences
USTHB, BP 32 El Alia, Bab-Ez Zouar, 16111, Algeria.*

Abstract: In this paper, we consider retrial queueing systems with batch arrivals in which the server is subject to controllable interruptions (called vacations) and random interruptions (breakdowns). No specific assumption is taken regarding probability distributions of parametric random variables. The purpose of this work is to show effect of above mentioned parameters (in particular retrial and breakdown parameters) upon main performance measures of interest. Next, we study some optimal control problems of vacation and retrial policies.

Keywords- Retrial queues, vacations, batch arrivals, breakdowns, cost model, optimal control, reliability.

1. INTRODUCTION:

Queueing (or service) systems arise in modelling of many practical applications related to Computer Sciences: Communication, Production, Human-Computer Interactions, and so on. In this paper we consider queueing systems which take into consideration additional phenomena:

- (i) Batch arrivals of customers: in many computer systems, the message is transmitted by packets (frames).
- (ii) Repeated attempts of unsatisfied customers: see the bibliographical paper of Artalejo 1997
- (iii) Idle time use of servicing device through introducing periods (for example maintenance actions in order to prevent the risk of failure): see the survey of Doshi (1986)
- (iv) Random interruptions due server breakdowns or other priority tasks.

Similar models have been used in concrete applications as the modelling of Digital Cellular Mobile Networks [Sun Jong Kwon, 2001], Local Area Networks with star topology [Janssens,1997]and so on. However all the models used there, neglected breakdown process.

In this work we also study the optimal control of vacation and retrial policies. Some attempts have been made in this direction by Artalejo (1997) in the case of Poisson arrival process and by Aissani (2000) in the case of batch arrivals. However, in both

papers they considered the case of constant retrial policy with an absolutely reliable server.

Next section is meant for description of the model. Section 3 concerns the analysis part of the problem where we obtain probability distribution of the system state. These results are obtained by the method of supplementary variables which is well known in Queueing Theory. So, we will provide only the necessary elements. This study confirms some decomposition properties showing the effect of vacations and retrials. Finally, we consider the problem of the optimal control of vacation policy (section 4) and retrial policy (section 5). We conclude the study by some numerical examples.

2. MODEL DESCRIPTION:

We consider a single server queue where batches of customers arrive according to a compound Poisson process. If an incoming batch finds the server idle, one of the batch members immediately begins service and the rest of customers in that batch join the retrial group (a sort of queue with infinite capacity also called: the orbit) and seek for service individually after a random amount of time. The server is subject to random breakdowns with rate θ . Whenever the server fails, it is immediately repaired. If an incoming batch finds the server unavailable (i.e. busy by the service of a certain customer, out of order or in vacation), then all customers in the batch join the orbit. Any customer accepted for service upon arrival or on retrial leaves the system forever after service completion. We consider the following policy to access the server from orbit. If the orbit is not idle at some instant, then a random customer is chosen to occupy the server after a random amount of time Λ . We assume that the server takes a random vacation

each time the system is empty. If the server returns from vacation to find one or more customers waiting in orbit, he works until the system empties, then begins another vacation. If the server returns from a vacation to find no customers in orbit, he begins another vacation immediately. We assume that all the considered variables are mutually independent.

Acronyms

PDF Probability Distribution Function
PGF Probability Generating Function
LST Laplace-Stieltjes Transform
NBUE New Better than Used in Expectation

Notation

λ =the batch arrival rate
 X =size of an arrival batch
 $G(z)$ =PGF of X
 $g_k=E(X^k)$ =kth order moment of X
 S =service time random variable
 Λ =Retrial time
 V =vacation time random variable
 θ = rate of breakdowns
 W =repair time of a breakdown
 $H(x), R(x), V(x), W(x)$ The PDF of the random variables S, Λ, V, W .
 $h(s), r(s), v(s), w(s)$ = The LST of the PDF $H(x), R(x), V(x), W(x)$
 h_k, r_k, v_k, w_k =the kth order moments of the PDF $H(x), R(x), V(x), W(x)$
 $M(t)$ =system size at time t
 $E(t)=0$ if server is operative at t
 $=1$ if the server is down
 $S(t)=0$ if the server is free at t
 $=1$ if it is busy
 $=2$ if it is on vacation
 $\xi(t)$ =the remaining retrial time if
 $E(t)=S(t)=0$
 $=$ the remaining service time if
 $E(t)=0, S(t)=1$
 $=$ the remaining repair time if
 $E(t)=0, S(t)=2$

$$Q(z) = \lim_{t \rightarrow \infty} E(z^{M(t)})$$

\mathbb{R}^+ =the set of non negative real numbers

3. ANALYSIS OF THE MODEL:

First, we develop some analytical properties of the system under study.

3.1.Fundamental Process.

The process $\zeta(t) = \{E(t), C(t), M(t), \xi(t)\}$ is a Markov process defined on the state space $\mathbf{E} = \{0, 1\} \times \{0, 1, 2\} \times \mathbb{1}\{0, 1, 2, \dots\} \times \mathbb{R}^+ \setminus \{y: y=(0, 0, x), x \geq 0\}$ which can be studied by using a method similar to that of [Aissani, 2000]. Thus we restrict analysis to the description of obtained results. First we have that a condition for the system to be stable is

$$\rho = \frac{\lambda}{\delta} (g_1 - 1) + \frac{\lambda g_1 [\theta w_1 + (\delta + \lambda) m_1] - \theta}{\delta} < 1$$

$$\text{where } \delta = \frac{(\lambda + \theta)r(\lambda + \theta)}{1 - r(\lambda + \theta)}$$

and [Aissani & Artalejo, 1998]:

$$m_1 = (1 - h(\theta))(w_1 + \theta^{-1});$$

It is not surprising that this condition depends on reliability parameters. However, we note that it is independent of the vacation parameter and contrary to the case of linear retrial rate (see [Aissani and Artalejo, 1998]) it depends on retrial time distribution. We assume from now that $\rho < 1$, so the stationary probabilities:

$$P_{ij}(m, x) = \lim_{t \rightarrow \infty} P\{E(t)=i, C(t)=j, M(t)=m; \xi(t) < x\}$$

exists. Now, define by $Q_{ij}(z, x)$ the PGF in m , and the Laplace transform $f_{ij}(z, s)$ in x . By usual way, we obtain:

$$f_{00}(z, s) = \frac{(\lambda + \theta)[r(\lambda + \theta) - r(s)]}{s(s - \lambda + \theta)[1 - r(\lambda + \theta)]} Q_{00}(z, \infty)$$

$$f_{10}(z, s) = \theta Q_{00}(z, \infty) \frac{w(\varepsilon(z)) - w(s)}{s(s - \varepsilon(z))}$$

$$f_{01}(z, s) = \frac{v + \lambda G(z)}{z} \frac{h(\varepsilon(z)) - h(s)}{s(s - \varepsilon(z))} Q_{00}(z, \infty)$$

$$f_{02}(z, s) = \alpha \frac{v(\varepsilon(z)) - v(s)}{s(s - \varepsilon(z))} \text{ where}$$

$$\alpha = (1 - \rho) \frac{\delta}{(\lambda + \delta)v_1}, \quad \varepsilon(z) = \lambda - \lambda G(z).$$

3.2. System Size Distribution:

The PDF of the number of customers in the system at an arbitrary point can be derived from previous section as in [Aissani, 2000]:

$$Q(z) = (1 - \rho) \frac{r(\lambda + \theta)}{v_1} \frac{(1 - z)[1 - v(\varepsilon(z))]}{\varepsilon(z)} \times$$

$$\frac{(\lambda G(z) + \delta)h(\varepsilon(z) + \theta w(\varepsilon(z)))}{[(\lambda G(z) + \delta)h(\varepsilon(z)) + \theta w(\varepsilon(z)) - (\lambda + \delta + \theta)z]}$$

3.3. Decomposition Result:

Using the above formula, we can obtain an interesting decomposition result. More precisely, the number of customers in our model can be expressed as a sum of three independent random variables representing the number of customers in: (i) an unreliable system with FIFO queue without vacation; (ii) our model given that the server is on vacation; (iii) an unreliable retrial queue without vacation given that the server is idle. Such a result is useful when computing higher order moments.

3.4. Reliability and Service Metrics:

Since the breakdown and repair processes are independent of the servicing processes, then the hardware reliability and availability metrics are defined in the usual way. Next, we can define:

Servicing availability: It is the probability that the server is available (in the hardware sense) and free

$$\text{of customers: } p_{00} = Q_{00}(1, \infty) = \frac{\lambda g_1}{\lambda + \delta + \theta}.$$

Probability that the server is available and busy by the service of a customer

$$p_{01} = Q_{01}(1, \infty) = \lambda g_1 h_1 \frac{\lambda + \delta}{\lambda + \delta + \theta}$$

Probability that the server is on vacation: $p_{02} = Q_{02}(1, \infty) = (1 - \rho)r(\lambda + \theta)$

Average number of customers in the system: This characteristic can be obtained directly using formula of §3.2.:

$$E\{M\} = \frac{\lambda g_1 v_2}{v_1} + \Psi,$$

$$\text{where } \Psi = \frac{\beta \delta + \gamma}{2\delta(1 - \rho)} + \frac{\lambda g_1 h_1 \delta + \alpha}{\lambda + \theta + \delta}$$

$$\alpha = \lambda g_1 + \lambda^2 g_1 h_1 + \theta \lambda g_1 w_1$$

$$\beta = \lambda g_2 h_1 + (\lambda g_1)^2 h_2$$

$$\gamma = \lambda g_2 + 2(\lambda g_1)^2 h_1 + \lambda \alpha + \theta \lambda g_1 w_1 + \theta (\lambda g_1)^2 w_2$$

Mean waiting time: From Little's formula we have:

$$E(W) = \lambda g_1 E(M).$$

4. CONTROL OF VACATION POLICY:

This section illustrates usefulness of the results of previous sections by giving applications to optimal control of the vacation policy.

4.1. Cost Function: Let us consider the following costs.

C_s =setup cost per cycle (each time the server is reopened).

C_h =holding cost per unit time (incurred for each customer present in the system).

C_d =breakdown cost per unit time for a failed server.

C_0 =cost per unit time for keeping the server on and in operation.

As usual, the expected exploitation time costs per unit can be expressed as

$$C = \frac{C_s}{E(L)} + C_h E(M) + C_0 \frac{E(A)}{E(L)} + C_d \frac{E(B)}{E(L)}$$

where A and B are the sojourn times in the corresponding states during a cycle L with mean

$$E(L) = \frac{E(V)}{(1 - \rho)r(\lambda + \theta)}. \text{ We consider the policy}$$

under which the server is turned off when system becomes empty and it is turned on again when the number of customers reaches the threshold N. In this case, the cost function C(N) is expressed as

$$C(N) = \frac{C_s \lambda g_1 (1 - \rho) \delta}{N(\lambda + \theta + \delta)} + C_h \left\{ \frac{N - 1}{2} + \Psi \right\}$$

up to a fixed cost which is independent of N.

4.2. Optimal Threshold. We are now able to find the optimal value N^* which minimizes the cost function C(N). Since this cost is a convex function, then the optimal value is one of integers adjacent to the value

$$N^* = \sqrt{\frac{2C_s \lambda g_1 (1 - \rho) r(\lambda + \theta)}{C_h}}.$$

4.3. Effect of Retrials Upon The Optimal N^* .

We consider here effect of retrial distribution $R(\cdot)$ upon the optimal values for both vacation policies. Consider the class \mathfrak{T}_m^σ of PDF with mean m and finite variance σ^2 , and $\mathfrak{T}_m^{\text{NBUE}}$ the class of all PDF on $[0, \infty)$ with mean m that are NBUE. Recall that a PDF on \mathbb{R}^+ is NBUE if and only if

$$\int_x^\infty F(y) dy \leq m F(x) \text{ for } x \geq 0. \text{ Let } \theta_m = 0 \text{ if } x < m,$$

and $\theta_m = 1$ if $x \geq m$.

(1) If retrial time distribution $R(x)$ belongs to the class \mathfrak{T}_m^σ , then the optimal value of N^* is bounded

as follows: $N^*_L < N^* < N^*_U$ where upper and lower bounds are given respectively by

$$N^*_L = \theta_0(\eta) \sqrt{\frac{2S\lambda g_1(\eta)}{h}}$$

$$\eta = e^{-\lambda r_1} g_1 - (g_1 - 1) - \lambda g_1 h_1$$

$$N^*_U = \theta_0(\chi) \sqrt{\frac{2S\lambda g_1(\chi)}{h}}$$

$$\chi = r_U(\lambda) g_1 - (g_1 - 1) - \lambda g_1 h_1$$

$$\text{and } r_U(\lambda) = \frac{r_2 - r_1^2}{r_2} + \frac{r_1^2}{r_2} e^{-\lambda \left(r_1 + \frac{r_2 - r_1^2}{r_1^2} \right)}$$

(2) If $R(x)$ belong to $\mathfrak{S}_m^{\text{NBUE}}$ then $N^*_L < N^*_{\text{NBUE}} < N^*_{\text{EXP}}$ where N^*_{NBUE} is the optimal value for an $M^X/G/1$ vacation queue with **NBUE** retrial time distribution, and N^*_{EXP} is the optimal value for the model with vacation and constant retrials [Artalejo, 1997].

Remark. For sake of space, we have considered here only the case of a reliable server ($\theta=0$). The first inequality gives approximations (in fact lower and upper bounds) on the optimal threshold N^* when the retrial time distribution is unknown, but we have a partial information about the first two moments. The second one tell us about the case when the partial information concerns an ageing class of retrial time distribution.

5. CONTROL OF RETRIAL POLICY.

We now investigate the problem of optimal control of retrial parameter when the system operates under the N -policy. Note that in fact the cost function depends only on the real value δ , and not on the concrete aspect of the retrial time distribution. So, we consider the problem of choice of an optimal value δ which minimizes the cost function C .

$$\text{Differentiate wrt: } \delta \text{ gives } \frac{C_S \lambda g_1}{N C_h} = F(\delta)$$

$$\text{where } F(\delta) = 1 - \frac{1}{2} \left(\frac{\lambda + \theta + \delta}{\delta(1 - \rho)} \right)^2 \times \frac{(\lambda + \theta - \alpha)\beta - \gamma(1 - \lambda g_1 h_1)}{\alpha - (\lambda + \theta)\lambda g_1 h_1}$$

The function $F(\delta)$ satisfies the following properties:

(i) Its domain's value is $F: [\delta_e, +\infty) \rightarrow \mathbb{R}^+$ where

$$\delta_e = \frac{\alpha - \lambda - \theta}{1 - \lambda g_1 h_1}$$

$$(ii) \lim_{\delta \rightarrow \delta_e} F(\delta) = +\infty$$

$$(iii) \lim_{\delta \rightarrow +\infty} F(\delta) = \Pi$$

where

$$\Pi = \frac{1}{1 - \lambda g_1 h_1} \times \left(1 + \frac{(1 - \lambda g_1 h_1)(\gamma - 2\lambda g_1 h_1 \xi) - (\lambda + \theta - \alpha)\beta}{2\xi(1 - \lambda g_1 h_1)} \right)$$

Note finally that $C'(\delta) < 0$ if and only if $F(\delta) > C_S \lambda g_1 / C_h N$. So the optimal value of δ^* is:

$$(i) \text{ If } \frac{C_S \lambda}{C_h N} \leq \Pi, \text{ then } \delta^* = +\infty.$$

$$(ii) \text{ If } \frac{C_S \lambda}{C_h N} > \Pi, \text{ then } \delta^* \text{ is solution of the}$$

$$\text{equation } F(\delta) = \frac{C_S \lambda g_1}{C_h N}, \text{ for } \delta > \delta_e.$$

6. NUMERICAL ILLUSTRATIONS:

In this section we illustrate the effect of parameters (retrial, vacation and breakdowns) on system performances. In the remainder of this section we take the basic data of [Artalejo, 1997]: $\lambda=1$, $g_1=1$, $g_2=0$, $h_1=0.25$, $h_2=1$. Concerning the maintenance parameters we take $w_1=0.1$ and $w_2=1$.

First, we show effect of failure rate on the retrial parameter δ . In figure 1 we have plotted the function $\delta(\theta)$ for different retrial PDF with mean $r_1=1$:

- (i) Hyperexponential (H_2).
- (ii) Exponential (Exp):
- (iii) Determinist (D):

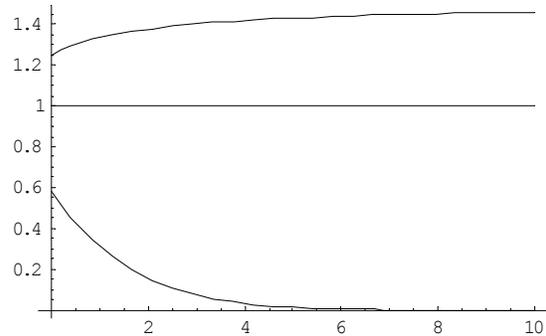


Figure 1. Effect of failure rate θ on δ .

We observe that the parameter δ increases in the case (i) and decreases in the case (iii) as the failure rate increases. In the case (ii) the parameter δ is independent of the failure rate. This can be easily understood from exponential nature of retrial time.

Figure 2 plots expectation $E(M)$ versus failure rate θ and ratio v_2/v_1 . We note that $E(M)$ decreases when θ and v_2/v_1 increases and increases otherwise.

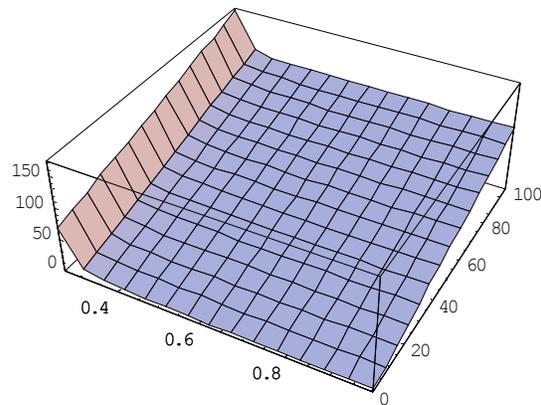


Figure 2. Effect of breakdowns and vacations on Mean system size .

Figure 3 shows effect of failure rate on the optimal threshold for different values of $C_s/C_h=10, 50$ and 100 . We have considered a 2-Erlangian retrial distribution (E_2) with mean $r_1=0.5$; We note that the optimal threshold increases with the ratio C_s/C_h .

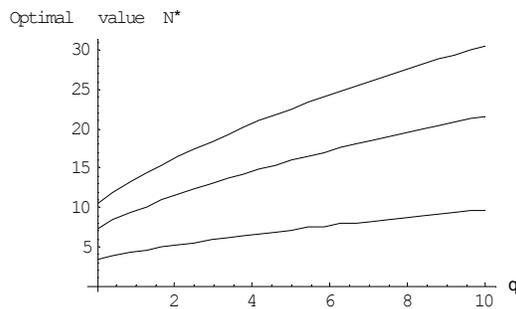


Figure 3. Effect of θ on the optimal N^* .

Table 1 compares lower and upper bounds on the optimal value N^* for different parametric (Exp, D, H_2) and non parametric (NBUE) retrial PDF which typify some PDF observed in practice. For each of these choices we varied the ratio C_s/C_h from 0.5 to 10^5 .

Figure 4 illustrates behaviour of the bounds as a function of the mean retrial time for different values of $C_s/C_h=10, 1, 0.1$. For a given value of

this ratio, the dot-dashed curve corresponds to a lower bound and the continuous curve to an upper bound. The lowest pair of curves corresponds to the case $C_s/C_h=0.1$. We see that lower bound tends to be more closed to the upper bound curve for small values of r_1 and C_s/C_h .

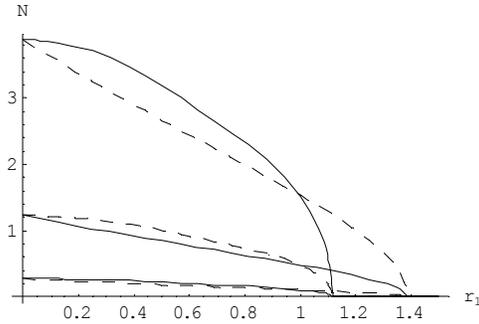


Figure 4. Bounds on the optimal threshold.

Finally, table 2 shows the joint effect of retrials and breakdowns upon the optimal value N^* and its corresponding minimum expected cost. The optimal value N^* increases and the cost decreases when both δ and θ increases (see also figure 5).

7.CONCLUSION: In this work we studied the effect of retrials, vacations and breakdowns on the performance metrics of queueing service systems. We have showed how to control the vacation and retrial mechanisms. A similar study can be provided to control the maintenance actions.

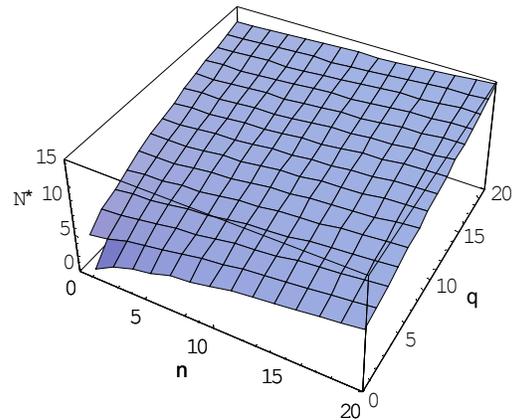


Figure 5. Effect of retrial rate δ and failure rate on the optimal threshold N^* .

8.REFERENCE:

[1].Aissani A. and Artalejo J.R. 1998, "On the single server retrial queue subject to breakdowns", *Queueing Syst.* 30, 309-321.

[2].Aissani A. 2000, "An $M^X/G/1$ retrial queue with exhaustive vacations", *J. of Statist. and Mangmnt Systems*, 3, N°3, 269-2863].

[3]Artalejo, J.R.1997. "Analysis of an M/G/1 queue with constant repeated attempts and server vacations", *Computers Ops. Res.* ,24, N°6, 493-504.3].

[4]Artalejo, J.R. 1999, "A classified bibliography of research on retrial queues: Progress in 1990-1999", (*TOP*), vol. 7, N°2, pp.187-211.

[5]Doshi, T. 1986, "Queueing systems with vacation-A survey", *Queueing Syst.*, 129-66.]

[6] Janssens G. K. 1997, "The quasi-random input queueing system with repeated attempts as a model for a collision-avoidance star local area network", *IEEE Trans. on Commun*; vol. 45, N°3, 361-364.

[7] Sun Jong Kwon & al, 2001. "Performance analysis of CDPD Sleep Mode for power conservation in Mobile End Systems", *IEIC Trans. Commun.*, vol. E84, N°10.

Table 1. Lower and Upper bounds on the optimal value N^*

S/h	0.5	1	10	24	10^5
Lower bound Determinist retrials	0.18708	0.37416	0.83666	1.29614	83.666
Exponential m=2; $\sigma^2=4$ m=1; $\sigma^2=1$	0.48370 0.63245	0.68318 0.894427	2.16023 2.82842	3.3466 4.38178	216.02314 282.8427
2-Erlang m=2; $\sigma^2=4$ m=2; $\sigma^2=2$	0.2626 0.38729	0.3714 0.54772	1.1747 1.732	1.8198 2.68328	117.473 173.205
NBUE m=2; $\sigma^2=1$	0.32989	0.46654	1.47523	2.28525	147.533
Upper bound m=2; $\sigma^2=4$ m=1; $\sigma^2=1$	0.64800 0.67820	0.916500 0.959100	2.8982 3.0331	4.48998 4.6989	289.8275 303.3150

Table 2. Optimal Thresholds N^* and its corresponding minimum cost.

$C_s=5, C_h=1, \lambda=1, g_1=1, g_2=0, w_1=0.1, w_2=1.$

$\theta=0; \delta \rightarrow$	0.35	0.4	0.5	1	10	20	50	∞
ρ	0.9642	0.875	0.75	0.5	0.275	0.2625	0.255	0.25
N^*	0.3042	0.5976	0.9128	1.5811	2.5672	2.6502	2.7025	2.7386
$C(N^*)$	7.5223	6.4904	5.3295	3.8311	3.1878	3.1705	3.1611	3.1552
$\theta=0.5; \delta \rightarrow$	0.85	1	10	20				∞
ρ	0.01	0.05	0.23	0.24				0.25
N^*	1.887	1.9493	2.5876	2.6589				2.7386
$C(N^*)$	3.9636	3.8340	3.2075	3.1804				3.1552
$\theta=1; \delta \rightarrow$	5	10	20	50	100			∞
ρ	0.12	0.18	0.185	0.23	0.24			0.25
N^*	2.5070	2.6060	2.6671	2.7086	2.7233			2.738
$C(N^*)$	3.2988	3.2256	3.1898	3.1689	3.1820			3.1552
$\theta=10; \delta \rightarrow$	50	100						∞
ρ	0.03	0.16						0.25
N^*	2.7564	2.7482						2.738
$C(N^*)$	3.2457	3.1946						3.1552

EXPONENTIALLY FAST MONTE CARLO SIMULATIONS FOR NON-EQUILIBRIUM SYSTEMS

A. BANDRIVSKYY¹, S. BERI¹, D. G. LUCHINSKY¹,
R. MANNELLA² AND P. V. E. McCLINTOCK¹

¹*Department of Physics, Lancaster University, Lancaster LA1 4YB, UK*

²*Dipartimento di Fisica and INFN, Università di Pisa, Italy*

Abstract: A new numerical technique is demonstrated and shown to reduce exponentially the time required for Monte Carlo simulations of non-equilibrium systems. The quasi-stationary probability distribution is computed for two model systems, and the results are compared with the asymptotically exact theory in the limit of extremely small noise intensity. Singularities of the non-equilibrium distributions are revealed by the simulations.

Keywords: *stochastic, non-equilibrium, probability distribution, Monte Carlo simulation*

1. INTRODUCTION

Fluctuations in systems away from thermal equilibrium represent a problem of long standing in statistical physics [Onsager and Machlup, 1953]. Well known examples of systems in which non-equilibrium fluctuations play a particularly important role include e.g. lasers [Keay et al., 1995], proteins [Serpersu and Tsong, 1983], Josephson junctions [Kautz, 1996], and chemical reactions [Smelyanskiy et al., 1999b]. In particular, activated processes are of big importance. Noise induced escape means a transition from one state to another, which e.g. in chemical system describes a reaction [Smelyanskiy et al., 1999b; Huber and Kim, 1996]. In non-equilibrium systems, where symmetries of detailed balance are broken, no general methods exist for the calculation of even basic quantities like the probability distribution. This is a case where numerical and asymptotic theoretical methods for investigating the probability distribution are of particular importance.

In the limit of small noise intensity, $D \rightarrow 0$ [Ventcel and Freidlin, 1970; McKane, 1989; Dykman, 1990; Smelyanskiy et al., 1999b], asymptotic theoretical approaches, such as WKB-like or path-integral methods, can be used. The theory suggests that a solution to the problem of non-equilibrium fluctuations requires an understanding of the dynamics of deviations from the steady state [Onsager and Machlup, 1953] and an analysis of singularities in the non-equilibrium potential [Graham and Tel, 1984; Smelyanskiy et al., 1997]. Some ideas have recently been proposed for how to extend the existing ($D \rightarrow 0$ limit) theory to encompass the case

of still small but finite noise intensity [Smelyanskiy et al., 1999a; Lehmann et al., 2000; Bandrivskyy et al., 2003].

Monte Carlo simulation provides the main numerical technique used to verify such theoretical predictions, and to analyse the behavior of the dynamical system under study. The theory gives an asymptotically exact solution only in the $D \rightarrow 0$ limit. In contrast, D in the numerical simulations is necessarily finite. Typically, the time required for Monte Carlo simulations grows exponentially as $D \rightarrow 0$. This meant that theoretical predictions of interesting singular structures, and of the non-equilibrium probability distribution [Graham and Tel, 1984; Jauslin, 1987], for long remained untested either experimentally or by numerical simulation. Furthermore, there has been no clear understanding of how the picture changes as the noise intensity becomes finite.

Earlier attempts to speed up the simulations focussed mainly on finding optimal fluctuational paths and rates of transition in between stable states of a system (e.g. efficient transition path sampling [Dellago et al., 1999] and dynamics importance sampling [Woolf, 1998], which follow the earlier suggestion of [Pratt, 1986]). In [Crooks and Chandler, 2001] the path sampling method was adapted to non-equilibrium systems. Based on the same idea, the umbrella sampling technique was suggested to estimate the probability to reach any point of phase space of an equilibrium system starting from a fixed initial state [Dellago et al., 1999]. An idea how to improve sampling by splitting up the probability packets was suggested in [Huber

and Kim, 1996]. So far, however, no general algorithm has been suggested for non-equilibrium systems, able to provide both the whole probability distribution and also dynamical information, e.g. optimal fluctuational paths, for small noise intensities.

We now introduce a numerical method that enables the time required for Monte Carlo simulations to be reduced by an exponentially large factor. It is applicable to generic two-dimensional non-equilibrium systems, does not require any *a priori* knowledge about the system apart from its dynamical equations of motion, and it allows the quasi-stationary probability distribution to be computed directly over the whole phase space. Using this method, we reveal singular behavior of the non-equilibrium distribution and show that it is in quantitative agreement with the asymptotic theory. The central idea is to perform the simulations in sequential steps.

We construct the quasi-stationary distribution in stages, patching together intermediate results, starting from the minimum of the potential and gradually moving away from it. We find that the time required for the simulations at each step is reduced by an exponentially large factor as compared to the standard technique: if the time necessary for a conventional Monte Carlo simulation technique is T , our modified method would require only time

$$T_m \approx NT \exp^{-(N-1)\frac{\Delta\Phi}{D}},$$

where N is the number of steps involved and $\frac{\Delta\Phi}{D}$ is distance in logarithm of the probability between them. Given that T is exponentially large, the benefit in reduced processing time can be very substantial. The result of simulations for Duffing system (Fig.3, for $D = 0.02$) can be practically directly compared with a result given by an ordinary technique. It took us around 15 minutes to simulate the whole distribution shown in Fig.3 (for $D = 0.02$) with our fast technique, and it takes around four days of standard Monte Carlo simulation to obtain comparable statistics close to the boundary of attraction.

We explain the underlying principle of the method in Sec. 2, testing it by application to a very simple equilibrium stochastic system where all the results are already known. Then, in Sec. 3 we apply it to two much-studied non-equilibrium systems and compare the numerical results with the corresponding theoretical predictions. Finally, we summarise our conclusions in Sec. 4.

2. THE FAST MONTE CARLO SIMULATIONS TECHNIQUE

To illustrate the technique, we consider an over-

damped Brownian particle moving in a bistable Duffing potential $U(x) = -x^2/2 + x^4/4$. The corresponding Langevin equation is

$$\dot{x} = -U'(x) + \xi(t), \quad (1)$$

where $\xi(t)$ is zero-mean white Gaussian noise with intensity D and moments

$$\langle \xi(t) \rangle = 0, \quad \langle \xi(t)\xi(0) \rangle = 2D\delta(t).$$

The form of the probability distribution is completely defined by the potential $U(x)$, and is of the Boltzmann type $\rho(x) \propto \exp(-U(x)/D)$. As in the case of a non-equilibrium system (where the probability distribution is not defined by a potential) a standard Monte Carlo technique can be used to deduce $\rho(x)$. Numerical integration of the Langevin equation (1), assuming the system to be located initially at one of the potential minima x_m , gives the discrete probability distribution $\rho(x)$, peaked at x_m . The potential can be deduced as $\Phi(x) \propto -D \ln \rho(x)$. If the noise intensity is very small, the system fluctuates in a close vicinity of x_m and large deviations from it are extremely rare. The conventional Monte Carlo method cannot be used to study the dynamics of optimal escape paths, or the properties of the probability distribution far from the potential minima: the statistics required cannot in practice be collected within a realistic time when the noise intensity is within the range of interest, i.e. small but finite. We have overcome this problem by starting from the distribution already obtained near x_m .

We fix two probability levels ρ_i and ρ_f , lying well within the region where the numerical ρ is accurate, with $\rho_f < \rho_i$ corresponding to two levels in the potential Φ_i and Φ_f , and two coordinates x_i and x_f , as shown in Fig. 1. We require the levels ρ_i and ρ_f to be fairly different, such that the corresponding x_i and x_f are sufficiently separated: the distance between them must exceed \sqrt{Dh} , where h is the integration time step used in the Monte Carlo simulation, and must also exceed the discretization step Δx in the discrete probability distribution. All simulations were carried out following the procedure described by Mannella [2000].

The next simulation step is started from the level Φ_i (putting the system at $x = x_i$ as its initial condition). If the system starts to evolve along a fluctuational trajectory (towards the boundary of attraction) we just follow its natural dynamics according to (1) and collect the statistics for building the probability distribution in a usual way. If the system starts with a relaxation trajectory (towards x_m), or when it crosses the boundary x_i due to relaxation some time later, we stop the simulation and reinject the system back to the initial state x_i .

In this way we simulate the full dynamics of the system at higher levels of the potential $\Phi(x) > \Phi_i$ (in the region of coordinate space $x > x_i$ for this particular case). Thus, in the subsequent simulation step we follow only those fluctuations that have already attained a certain level in the potential Φ_i , without waiting for this exponentially slow event to happen. In this way, a new piece of the probability distribution is built with a time saving $\sim \exp \Phi_i/D$ compared to a simulation starting from the potential minimum x_m . The upper curve of Fig.1 shows the new piece of the potential $\Phi_2(x)$, as computed.

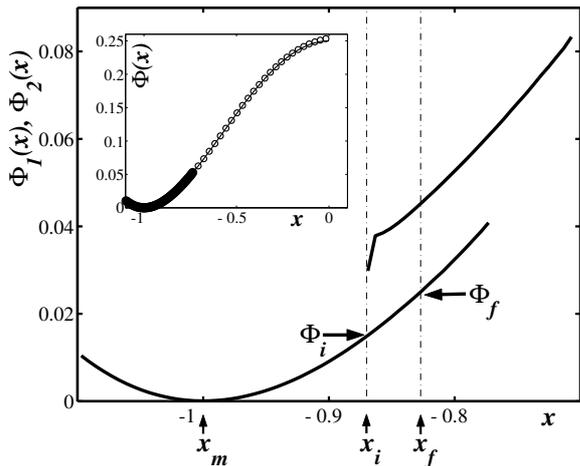


Fig. 1: The first ($\Phi_1(x)$, lower curve) and second ($\Phi_2(x)$, upper curve) pieces of the inferred potential $\Phi(x)$ for the system (1) with $D = 0.005$. The discontinuity in the gradient of $\Phi_2(x)$ near x_i is an artefact due to a boundary effect in the calculation of the discrete probability distribution. To avoid this problem $\Phi_1(x)$ and $\Phi_2(x)$ are merged at the point x_f and the initial part of $\Phi_2(x)$ is discarded. We normalize $\Phi_1(x)$ choosing $\Phi_1(x_m) = 0$, and each successive piece of $\Phi(x)$ is normalized in order to match with the previous one at the point where they join. Inset: The inferred potential $\Phi(x)$ for the system (1) with $D = 0.005$. The new technique (circles) is compared with standard Monte Carlo simulations (bold line) and with the Duffing potential $U(x)$ (thin line).

The merging of the two pieces of the inferred potential (the original $\Phi_1(x)$ and the new $\Phi_2(x)$) at x_f can be effected in a unique way. Continuing this procedure, the probability distribution and the corresponding potential can be built, step by step, for the whole region of interest. The inset in Fig. 1 shows the resultant potential, built from 13 such pieces between the minimum at $x_m = -1$ and the maximum at $x = 0$. It coincides closely with the Duffing potential $U(x)$ itself. The potential $\Phi(x)$ is thus inferred within a region of coordinate space that is inaccessible in a conventional

simulation (shown as bold curve for comparison). The advantage of our new technique is immediately evident. We stress that, in the simulations, no *a priori* knowledge of the dynamics was required.

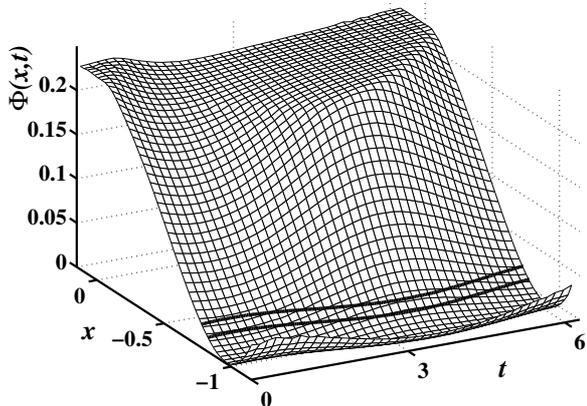


Fig. 2: The whole inferred $\Phi(x,t)$ for the system (2) for $A = 0.1$, $\Omega = 1$, $D = 0.005$. Two lines are the lines of constant probability found after the first step of simulations. The corresponding levels of probability were chosen as $\Phi_i = 3D$ and $\Phi_f = 5D$.

Essentially the same procedure can be applied to a two dimensional system. The main difference is that, instead of identifying two boundary points x_i and x_f , we need to identify two boundary lines of constant probability. One line is for starting simulations from, and another one is a reference line for matching together different pieces of the probability distribution (see Fig.2 for clarification). In turn, this implies that we should consider the reinjection location probability (RLP) along the “lower” boundary line corresponding to ρ_i . Starting from the second step of the simulations, the system should be reinjected back according to the RLP after it relaxes across the boundary. We emphasize that the RLP is not the same as the probability distribution $\rho(\mathbf{x})$, which is constant on the boundary line. The RLP is an additional important measure which describes local discrete dynamics in the neighborhood of the boundary line. It is a distribution along the boundary of how often the system crosses the boundary.

The principle of detailed balance that applies in equilibrium systems provides a symmetry that can be used to reinject the system back at the boundary level, without any need to compute the RLP. For non-equilibrium systems, however, detailed balance does not apply and so the procedure cannot be used. The RLP must be considered separately (and calculated explicitly) for the particular system being investigated. It can be obtained from an analysis of the finite difference equation corresponding to the model. In the limit of small integration time

step the probability to cross the boundary is proportional to the diffusion-related term in the finite difference equation. Then the RLP is simply proportional to the projection of the vector orthogonal to the boundary onto the coordinate affected by the noise ξ . It can also be computed numerically.

3. APPLICATION TO NON-EQUILIBRIUM SYSTEMS

As a first example of a non-equilibrium system, consider the periodically-driven overdamped Duffing oscillator

$$\dot{x} = -U'(x) + A \cos \Omega t + \xi(t). \quad (2)$$

We infer $\Phi(x, t)$ as $-D \ln \rho(x, t)$. This quantity corresponds to the theoretical “global minimum of the modified action” in the Hamiltonian theory of large fluctuations [Bandrivskyy et al., 2003] and, in the limit $D \rightarrow 0$, it becomes the non-equilibrium potential.

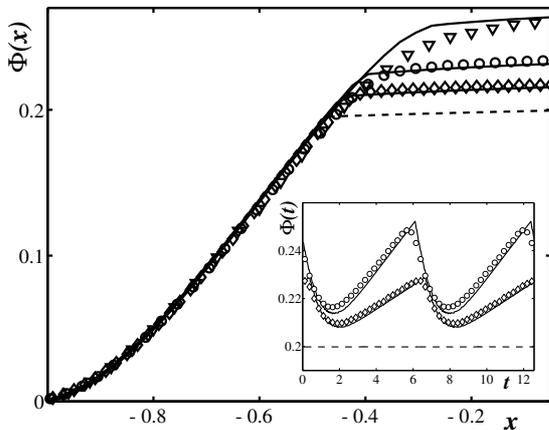


Fig. 3: A time section of the inferred $\Phi(x, t = 4.1)$ for the system (2) with $A = 0.1$, $\Omega = 1$, and different noise intensities: $D = 0.005$ (diamonds); $D = 0.01$ (circles); and $D = 0.02$ (triangles). The theoretical predictions are shown by full lines for finite noise intensities, and by dashed line for $D = 0$. Inset: oscillations of $\Phi(x, t)$ at the boundary of attraction for different noise intensities.

The limit of small noise intensity is of particular interest and importance in the case of non-equilibrium systems. A sufficiently small D gives rise to the possibility of revealing the non-equilibrium potential

$$\Phi(\mathbf{x}) = \lim_{D \rightarrow 0} -D \ln \rho(\mathbf{x}),$$

directly through a numerical experiment. Observations of the predicted singular shape of $\ln \rho(\mathbf{x})$, and of its dependence on D , are thus of considerable interest.

Fig. 2 shows the complete $\Phi(x, t)$, constructed from 12 such pieces, and a time section of $\Phi(x, t)$ calculated for different noise intensities together with the results of theoretical calculations (Hamiltonian theory including the prefactor) [Bandrivskyy et al., 2003] is shown in Fig. 3. The RLP in the simulations can be taken as constant if a small enough integration time step is used in the scheme. A small difference between the theory and the simulations results appears for the larger noise intensities, and then the asymptotic theory starts to break down and becomes inapplicable.

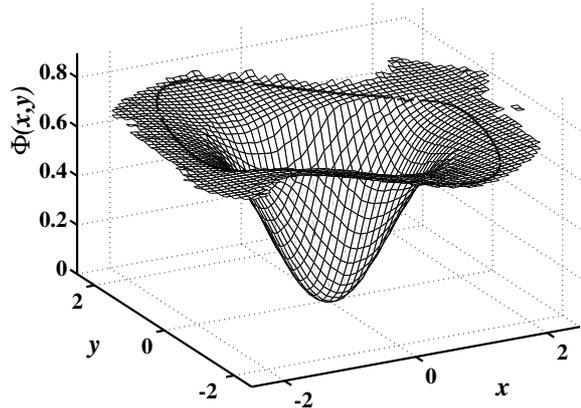


Fig. 4: Inferred $\Phi(x, y)$ for the system (3) with $\omega_0 = 1$, noise intensity $D = 0.01$, and $\eta = 0.5$. The boundary of attraction (unstable limit cycle) is shown by a bold curve.

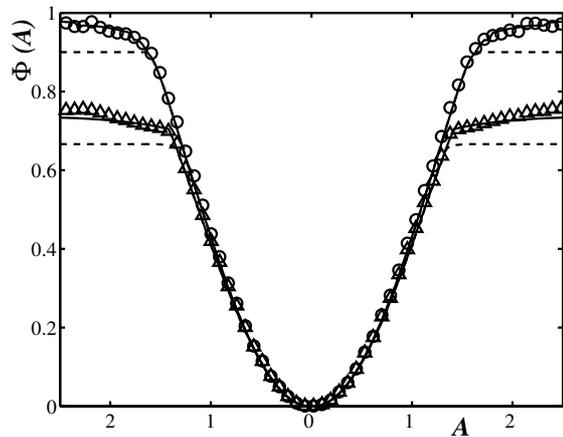


Fig. 5: A section ($x = y$) of the inferred $\Phi(A)$ for the system (3) with $\omega_0 = 1$, noise intensity $D = 0.01$ and $\eta = 0.25$ (circles); and $\eta = 0.5$ (diamonds). Theoretical predictions are shown in each case for $D = 0$ (dashed curves) and $D = 0.01$ (full curves).

We now consider, as a second more complicated non-equilibrium example, the inverted Van-der-Pol oscillator

$$\ddot{x} + 2\eta(1 - x^2)\dot{x} + \omega_0^2 x = \xi(t) \quad (3)$$

In order to be able to merge more easily the different pieces of $\Phi(x, y)$, we apply a coordinate transformation from x and $y = \dot{x}$ to amplitude A and phase ϕ

$$x = A \cos(\phi), \quad y = -A\omega_0 \sin(\phi).$$

It is then possible to analyse the probability $\rho(x, y)$ in the (A, ϕ) coordinate space. This makes the problem very similar to the periodically driven Duffing oscillator: the only difference is the RLP which, in the case of the Van der Pol oscillator, turns out to be strongly modulated. It is essential for this modulation to be taken into account when reinjecting the system back to the boundary of constant probability. The complete $\Phi(x, y)$ built by the Fast Monte Carlo simulations is shown in Fig.4. Two sections of $\Phi(x, y)$, obtained from the simulations for two different parameters η , are compared with the theory in Fig. 5. Here again, excellent agreement is obtained between numerics and theory.

4. CONCLUSIONS

The same structure of singularities is found in both of the non-equilibrium systems considered in this paper. Using the fast Monte Carlo simulations we reveal plateaus, the essentially flat regions in the probability distribution, which can be observed close to boundaries of attraction. They result from a purely dynamical effect that is not associated with the flatness of any potential. We have shown that its origin is related to switching between different types of optimal fluctuational path, and it is a general feature of non-equilibrium systems with metastable states [Bandrivskyy et al., 2003, 2002]. The switching lines [Smelyanskiy et al., 1997] are revealed as lines along which the “global minimum of the modified action” $\Phi(\mathbf{x})$ exhibits sharp bends – corresponding to the predicted line at which the non-equilibrium potential is non-differentiable. In the boundary region we found the oscillations of the probability distribution and their dependence on noise intensity (see the inset in Fig.3) discussed in the recent publications [Smelyanskiy et al., 1999a; Lehmann et al., 2000; Maier and Stein, 2001]. The noise-induced shift of the singularities and the optimal escape path revealed by the simulations has stimulated a new step in the development of the theory [Bandrivskyy et al., 2003].

It is only in the limit of extremely small noise intensity that the singularities can be confidently observed, so that the use of our new fast technique

is crucial to their investigation. In addition to being fast, it preserves dynamical information. It can be further extended to encompass higher dimensional systems and maps, and it can readily be modified to analyse optimal fluctuational paths, including those that arise in the energy-optimal control problem [Khovanov et al., 2000].

ACKNOWLEDGEMENTS

The work was supported by the Engineering and Physical Sciences Research Council (UK), the Joy Welch Trust (UK), the Russian Foundation for Fundamental Science, and INTAS.

REFERENCES

- Bandrivskyy A., Beri S. and Luchinsky D.G. (2003). Noise-induced shift of singularities in the pattern of optimal paths. *Physics Letters A, in press*.
- Bandrivskyy A., Luchinsky D.G. and McClintock P.V.E. (2002). Simple approximation of the singular probability distribution in a nonadiabatically driven system. *Phys. Rev. E*, **66**, 021108.
- Crooks G.E. and Chandler D. (2001). Efficient transition path sampling for nonequilibrium stochastic dynamics. *Phys. Rev. E.*, **64**, 026109.
- Dellago C., Bolhuis P.G., Csajka F.S., and Chandler D. (1999). Transition path sampling and the calculation of rate constants. *J. Chem. Phys.*, **111**(5), 1964–1977.
- Dykman M.I. (1990). Large fluctuations and fluctuational transitions in systems driven by colored gaussian noise – a high frequency noise. *Phys. Rev. A*, **42**, 2020–2029.
- Graham R. and Tel T. (1984). Existence of a potential for dissipative dynamical systems. *Phys. Rev. Lett.*, **52**, 9–12.
- Huber G.A. and Kim S. (1996). Weighted-ensemble Brownian dynamics simulations for protein association reactions. *Biophysical Journal*, **70**, 97–110.
- Jauslin H.R. (1987). Nondifferentiable potentials for nonequilibrium steady states. *Physica A*, **144**, 179–191.
- Kautz R.L. (1996). Noise, chaos, and the Josephson standard. *Rep. Prog. Phys.*, **59**, 935–992.
- Keay B.J., Allen S.J., Galan J., Kaminski J.P., Campman K.L., Gossard A.C., Bhattacharya U., and Rodwell M.J.W. (1995). Photon-assisted electric field domains and multiphoton-assisted

- tunneling in semiconductor superlattices. *Phys. Rev. Lett.*, **75**, 4098.
- Khovanov I.A., Luchinsky D.G., Mannella R., and McClintock P.V.E. (2000). Fluctuations and the energy-optimal control of chaos. *Phys. Rev. Lett.*, **85**, 2100.
- Lehmann J., Reimann P. and Hänggi P. (2000). Surmounting oscillating barrier. *Phys. Rev. Lett.*, **84**(8), 1639–1642.
- Maier R.S. and Stein D.L. (2001). Noise-activated escape from a sloshing potential well. *Phys. Rev. Lett.*, **86**(18), 3942–3945.
- Mannella R. (2000). In *Stochastic Processes in Physics, Chemistry and Biology*, Freund J.A. and Pöschels T. eds. (Springer-Verlag, Berlin), pp 353–364.
- McKane A.J. (1989). Noise-induced escape rate over a potential barrier: Results for a general noise. *Phys. Rev. A*, **40**(7), 4050–4053.
- Onsager L. and Machlup S. (1953). Fluctuations and irreversible processes. *Phys. Rev.*, **91**(6), 1505–1512.
- Pratt L.R. (1986). A statistical method for identifying transition states in high dimensional problems. *J. Chem. Phys.*, **85**(9), 5045–5048.
- Serpensu E.H. and Tsong T.Y. (1983). Stimulation of a ouabain-sensitive RB⁺ uptake in human-erythrocytes with an external electric-field. *J. Membr. Biol.*, **74**, 191.
- Smelyanskiy V.N., Dykman M.I. and Golding B. (1999a). Time oscillations of escape rates in periodically driven systems. *Phys. Rev. Lett.*, **82**(16), 3193–3197.
- Smelyanskiy V.N., Dykman M.I. and Maier R.S. (1997). Topological features of large fluctuations to the interior of a limit cycle. *Phys. Rev. E*, **55**(3), 2369–2391.
- Smelyanskiy V.N., Dykman M.I., Rabitz H., Vugmeister B.E., Bernasek S.L. and Bocarsly A.B. (1999b). Nucleation in periodically driven electrochemical systems. *J. Chem. Phys.*, **110**(23), 11488–11504.
- Ventcel A.D. and Freidlin M.I. (1970). On small random perturbations of dynamical systems. *Uspehi Mat. Nauk*, **25**, 1–56.
- Woolf T. (1998). Path corrected functionals of stochastic trajectories: towards relative free energy and reaction coordinate calculations. *Chem. Phys. Letters*, **289**, 433–441.

BIOGRAPHY



Andriy Bandrivskyy was born in 1978 in Lviv, Ukraine. He graduated from the Physics Department of Lviv National University in 2000. Now he is a PhD student in Nonlinear Dynamics group, Lancaster University, UK.

e-mail: band@lancaster.ac.uk

SIMULATION METHODOLOGY FOR ASSESSING MxRAN ARCHITECTURE PERFORMANCE

ANJA WIEDEMANN

*University of Essen
Institute for Experimental Mathematics
Ellernstr. 29
D-45326 Essen, Germany
Phone: +49 201 183-7635
Fax: +49 201 183-7673
Email: wiedem@exp-math.uni-essen.de*

PETER SCHEFCZIK

*Lucent Technologies
Wireless Advanced Technology Lab
Thurn-und-Taxis-Strasse 10
D-90411 Nuremberg, Germany
Phone: +49 911 526-4604
Fax: +49 911 526-3183
Email: pschfczik@lucent.com*

GEORGIOS NIKOLAIDIS

*University of Athens
Communication Networks Laboratory
Panepistimiopolis, Ilisia
157 84 Athens, Greece
Phone: +30 2 10 727 5654
Fax: +30 2 10- 727 5601
Email: nikolaid@di.uoa.gr*

Abstract¹: Product life cycles in communications and other technology related industries are getting shorter and shorter. Consequently the performance of such systems must be evaluated early in the architecture design stage. In the wireless networks area new radio access technologies are applied and form a multistandard radio access network (MxRAN) which denotes an evolution of the current UMTS (Universal Mobile Telecommunications System) architecture defined by standards bodies. The MxRAN is a main part of the IPonAir [IPonAir homepage] project. For this project several network architecture options and protocol stacks must be assessed. This makes the availability of an easy to use performance evaluation tool indispensable. In this paper the requirements and the modeling concept of a flexible signaling performance model are depicted. Thereby a use case approach starting with a description in Message Sequence Charts (MSCs) is applied. This set of MSCs is then converted automatically to a generic model that can be executed in an event driven environment.

keywords: Network Performance Models, Queueing Systems

1. INTRODUCTION

In the IPonAir project a new flexible radio access network (RAN) architecture supporting existing and future IP-based protocols is conceived. Starting from the UMTS Release 5 (R5) RAN architecture new architecture options of the above mentioned type are evolved. Several different solutions must be compared to derive an optimal architecture with respect to protocol processing cost and signaling delay. Therefore a performance assessment in an early stage of the software design is needed to detect the most advantageous architectures and avoid capital investment in technologies which are not efficient. This is why a flexible signaling performance assessment methodology for decision support is provided and implemented in a tool chain. Thus the methodology can be applied easily for the comparison of different MxRAN architectures.

The paper is organized as follows. First the requirements for the envisaged performance tool are identified. Then after a short overview on

UMTS R5 the main scenarios (use cases) of the MxRAN control plane are listed. MSCs [SDL Forum 2001] are used for the specification of the use cases. The MSCs are annotated with additional processing related information. From the set of MSCs a table related representation is derived automatically. This representation of the use cases is transferred into an event driven simulation tool. Predefined generic modules are used in the network nodes of the event driven system. The modeling concept and the node internal generic modules applied are described. Then the realization within an OPNET event driven environment is explained by means of two architecture examples. The paper closes with a summary of the simulation methodology and some remarks concerning the future development and implementation of this simulation methodology.

2. REQUIREMENTS

In this paragraph the requirements that have to be met by the envisaged simulation methodology are described.

¹The research reported in this paper has been partly conducted within the IPonAir program funded by the German Ministry of Education and Research (BMBF) under grant 01BU161. The responsibility for contents of this publication is with the authors.

- The focus is to evaluate MxRAN architecture alternatives by simulation with regard to performance in terms of signaling load, processing and memory capacity as well as delays assuming different traffic loads and mobility models.
- The evaluation of architecture variants shall be performed easily and efficiently.
- The bundling of protocol entities to network elements and the mapping of protocol entities to processors shall be possible in a flexible manner.
- The modeling concept shall support the separation of functions belonging to different planes, e.g. user plane and control plane.
- Main focus of the intended investigations is the signaling traffic load.

In order to meet these requirements, a generic and flexible modeling approach for the intended simulation study is needed.

3. UMTS R5 MODEL

As the future MxRAN architecture evolves from the current UTRAN R5 architecture, a short overview of the latter architecture is given hereafter.

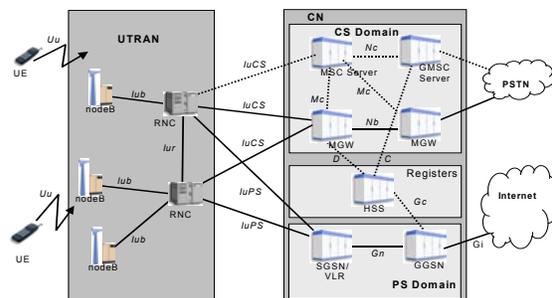


Fig. 3.1: UMTS Schematic R5 Architecture

Figure 3.1 depicts a simplified picture of the UMTS R5 architecture as defined by 3rd Generation Partnership Project (3GPP). Three different domains are depicted. First there is the UMTS Radio Access Network (UTRAN) consisting of base transceiver station (Node B) and Radio Network Controller (RNC) network elements. Second there exists the Circuit Switched (CS) domain of the Core Network (CN) consisting of a Media Gateway (MGW) for voice user data, the Mobile Switching Center (MSC) server for signaling purposes and the Gateway MSC (GMSC) server for connection to external networks. It must be noted that unfortunately inside the CN of the UMTS system the term MSC denotes Mobile Switching Center, but in the rest of this paper an MSC denotes a Message Sequence Chart. The third

domain is the Packet Switched (PS) domain within the CN comprising the Serving GPRS Support Node (SGSN) which is comparable to the Mobile Switching Center but for PS connections and the Gateway GPRS Support Node (GGSN) for connection to external packet data networks. The UTRAN handles all radio specific functionality. The CS domain handles all circuit switched sessions and connects the UTRAN to the Public Switched Telephone Network (PSTN) while the PS domain handles the packet switched sessions and connects the UTRAN to the public internet. Also depicted is the user equipment (UE) handling the user traffic in the cellular system and the Home Subscriber Server (HSS) storing the relevant user data. In UMTS R5 there are some more network elements performing call control and other signaling functions as well as multimedia resource functions. Those are not shown here for shortness sake. In Figure 3.1 dotted lines denote pure signaling links. Solid lines can transport signaling as well as user traffic.

The current UMTS architecture originates from the second generation (2G) GSM (Global System for Mobile communications) voice network. In GSM the PS domain was added later on in the form of the General Packet Radio Service (GPRS) subsystem. All this clarifies the fairly complex 3G architecture today and also in R5 as well as the permanent process of enhancing and optimizing these architectures. The further development of such 3rd generation networks and the finding of new improved architectures with regard to performance is a topic of active research just now [Uskela, 2003], [Yungsoo et al, 2003].

Certainly such rather complex networks impose strict delay requirements both on user and signaling traffic handling. Those requirements have to be met by the architectures under consideration and are one criterion for assessing these architecture options by simulation. In order to investigate this criterion, traffic has to be imposed on the simulation system. The traffic modeling concept and the traffic flows of interest as well as their derivation and adaptation to the overall simulation modeling concept are the main focal point now. To impose load on the modeled network elements, traffic must be generated inside the system. Within the traffic modeling concept the focus is on the signaling traffic, i.e. the control plane of the MxRAN. With regard to the signaling traffic the following use cases are of interest:

- CS Mobile Originated (MO) and Mobile Terminated (MT) Call Setup and Release
- PS MO and MT Call Setup and Release
- Intra- and Intersystem Handover
- Location Update procedure

- Paging procedure.

The first two procedures indicate the main activities of a user in a cellular network. Those are starting or receiving a voice call and using packet switched services by reading e-mail or surfing the web. The others are mobility related procedures involved when a user transits to a new cell or cell area or when the UE has to be located by the system when receiving a call.

These use cases can be described in the form of high level MSCs as specified in [3GPP TSG RAN, 2002] and [Kaaranen et al, 2001] where those MSCs are used to describe important UTRAN functions. The dominant signaling load is generated by these MSCs. In Fig. 3.2 the “RRC Connection Setup” procedure which is part of the CS MO Call Setup scenario is taken as an example. A Radio Resource Control (RRC) Connection is a point-to-point bi-directional connection between RRC peer entities on the UE and the UMTS RAN side, respectively. An UE has either zero or one RRC connection. The “RRC Connection Setup” procedure is used in several basic signaling traffic procedures to set up a signaling channel between a mobile terminal (UE) and the corresponding RNC via a base station (Node B).

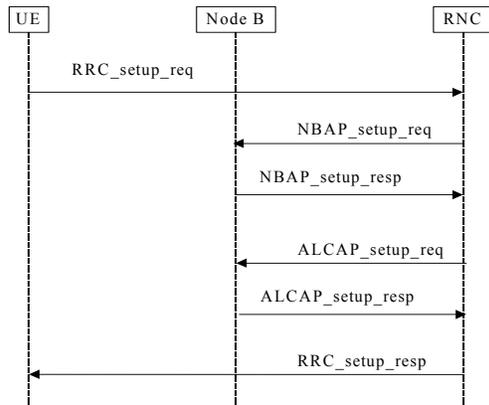


Fig. 3.2: High Level MSC for “RRC Connection Setup”

In the developed modeling concept the relevant network elements and protocol entities are identified from the MSCs given in the appropriate standards documents, e.g. [3GPP TSG RAN, 2002]. Considering the “RRC Connection Setup” procedure as an example, the relevant network elements which can be identified from Fig. 3.2 are the

- UE with the protocol entity RRC
- Node B and the protocol entities Access Link Control Application Protocol (ALCAP) and Node B Application Part (NBAP)
- RNC with the protocol entities ALCAP, NBAP and RRC.

As a consequence of this identification procedure, the MSCs provided by standards documents are refined as illustrated in Fig. 3.3 where the MSCs are extended by additional trigger messages to model the exchange of messages between protocol entities.

In the refined MSC (see Fig. 3.3) the relevant protocol entities explicitly communicate with each other while this communication is only implicitly modeled by the high level MSC illustrated in Fig. 3.2.

In the modeling concept the relevant protocol entities are modeled as stateless Functional Entities (FEs). This implies that a full functional model based on extended finite state machines (EFSMs) for the protocols is not needed. Hence the presented modeling approach is not state based. The reason for this design decision is that the modeling approach has to meet the requirement of a quick and easy evaluation of MxRAN architecture options.

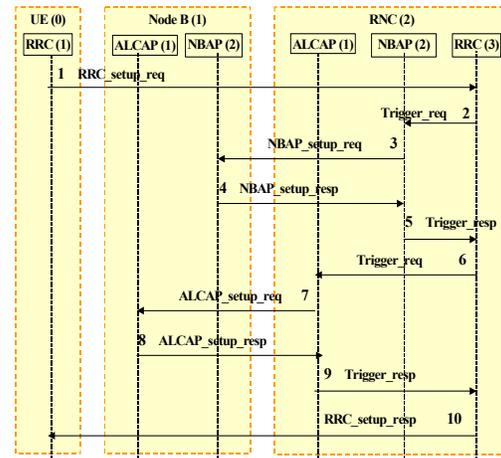


Fig. 3.3: Refined MSC

The context of FEs, MSCs and signaling traffic can be described as follows: FEs exchange specific sequences of signaling messages which are provided in the form of refined MSCs and model the use cases of interest described above.

4. MODELING CONCEPT

In order to set up network elements, a generic and modular node modeling concept is used. This is depicted in Fig. 4.1 and described subsequently.

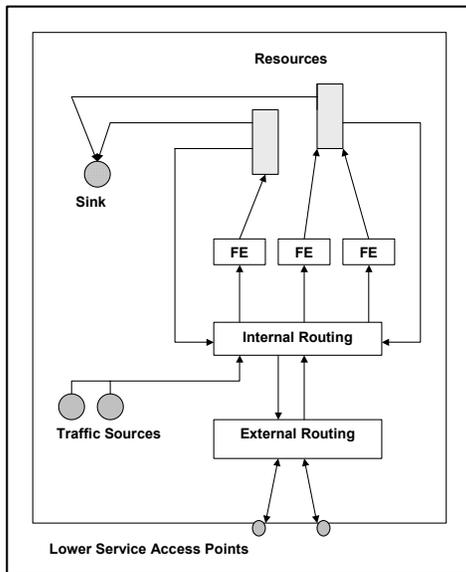


Fig. 4.1: Generic Node

Each network element consists of:

- One or more Lower Service Access Points (LSAPs)
- External Routing (ER) module
- Internal Routing (IR) module
- (Multiple) Functional Entities (FEs)
- One or more resource modules
- Sink module
- Optionally one or more traffic source modules.

Network elements which initiate signaling traffic contain *traffic sources* to model user (outgoing) and network generated (incoming) traffic. All sources starting a specific signaling sequence (e.g. all users in a cell starting an outgoing call) are aggregated. In a first step, the signaling sequences are triggered independently from each other and a statistical mix of signaling procedures derived from measurement data is used. One of the current activities is to refine the traffic model in order to model correlation among signaling procedures (e.g. setup and release of calls) without having to explicitly model every single traffic source.

An *LSAP* transparently interconnects the elements within the network. The LSAP can also be used as an interface to lower layer protocol stacks.

The *ER* and *IR* modules realize the routing functionality between network elements and within network elements, respectively. If a message reaches the ER module it is checked whether the message is addressed to the local node or not. If it is addressed to the local node the message is passed to the IR module, otherwise it is sent to the LSAP in direction to the addressed node which means that the message is just relayed by the ER module (e.g.

in Fig. 3.2 the “RRC Setup Request” within the “RRC Connection Setup” procedure is sent by the UE, addressed to the RNC and relayed by the ER module of Node B).

The *IR* module forwards an incoming message to the addressed FE.

FEs logically process incoming messages (e.g. convert them to another message type and provide new addresses) and pass the processed message to the *resource module* which models time consumption and resource contention. In reality FEs are located on physical processors which are modeled by the resource in this case. On a single processor one or more FEs can be located.

The resource forwards the last message of a signaling sequence to the *sink module* so that the End-to-End delay of the signaling procedure can be measured. Other messages are forwarded to the IR module which determines whether the message is addressed to a FE within the local node or not. If the message is addressed to a FE within the local node, the IR module sends the message to the addressed FE, otherwise the message is passed to the ER module. In this case the ER module sends the message to the LSAP in direction to the addressed node.

According to this generic node modeling concept and using the “RRC Connection Setup” procedure (see Fig. 3.2 and 3.3) as an example, the network element RNC can for example be modeled as illustrated in Fig. 4.2. Here the three FEs depicted in the RRC Connection Setup procedure are located on two resource modules. The ALCAP FE runs on the first resource while the NBAP and the RRC entities run on the second resource.

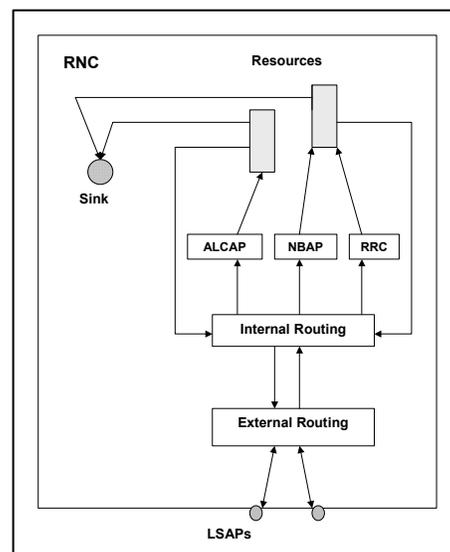


Fig. 4.2: Network Element RNC Modeled According To Generic Node Modeling Concept

5. REALIZATION

As far as an implementation of the simulation methodology is concerned, the OPNET Modeler environment was chosen to execute the simulations and analyze the results. OPNET is a quasi standard event-driven simulation environment provided by OPNET Technologies [OPNET homepage]. Despite the fact that OPNET provides a huge set of libraries for many well-known protocols, e.g. from the TCP/IP suite, the implementation of the methodology presented here does not necessarily make use of those OPNET built in protocol stacks. However, from the OPNET event driven environment the traffic generation package, queuing modules, links, graphical network editor and the analysis tool are used.

The refined MSCs (cf. section 3) that describe relevant signaling procedures are noted down in Microsoft Excel and automatically loaded into the OPNET simulation model via a Visual Basic for Applications (VBA) transformation algorithm. This VBA script generates the tables that can be imported directly into the OPNET environment. Extensible Markup Language (XML) or simple ASCII tables are used to specify and interchange the data. The network elements within the OPNET simulator are designed according to the generic node modeling concept (cf. Fig. 4.1). A simple RNC model with seven FEs mapped onto one resource is shown in Fig. 5.1 as an example for the OPNET node model of the network element RNC. The generic modules which build up the network elements (cf. section 4) are interconnected by packet streams, which is the mechanism for the exchange of messages between node internal modules provided by OPNET.

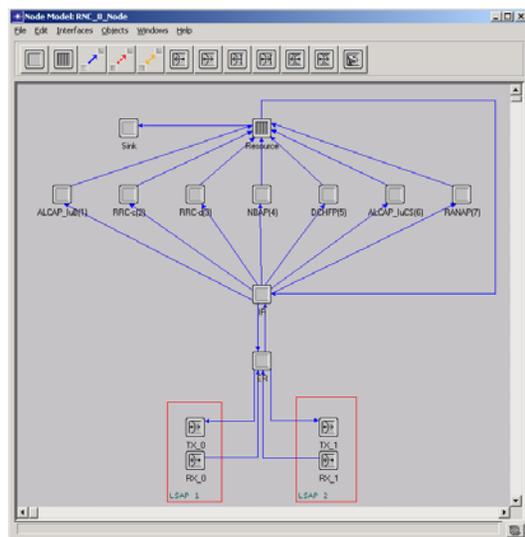


Fig. 5.1: Simple RNC Node Architecture Implemented in OPNET

Within the OPNET simulation model the MSCs are represented by a C code data structure within the FE modules. The FEs process incoming messages according to the MSC logic. Messages are annotated using complexity factors in order to specify the amount of processing time spent in the resource module. Thereby a higher complexity factor denotes more processing time in the resource. Moreover each message between two FEs is annotated with its message length. This determines the transport delay between two FEs. Within the current simulation model each signaling sequence is triggered by a single traffic source. Different traffic scenarios can simply be created by changing the simulation parameters of the traffic sources.

The OPNET implementation of the simulation concept at hand in particular fulfils the requirements presented in section 2. A flexible grouping of protocol entities to network elements and resources is easily possible by adding and removing FE modules and resources to and from network elements and re-interconnecting them by packet streams. For example the RNC illustrated in Fig. 5.2 contains seven FEs mapped onto two CPU (Central Processing Unit) resources. By doing so it is easily possible to set up different MxRAN architecture alternatives of interest for evaluation within simulation scenarios and assess them by means of performance figures (e.g. utilization of single resources, End-to-End delays of signaling sequences) provided by simulation runs.

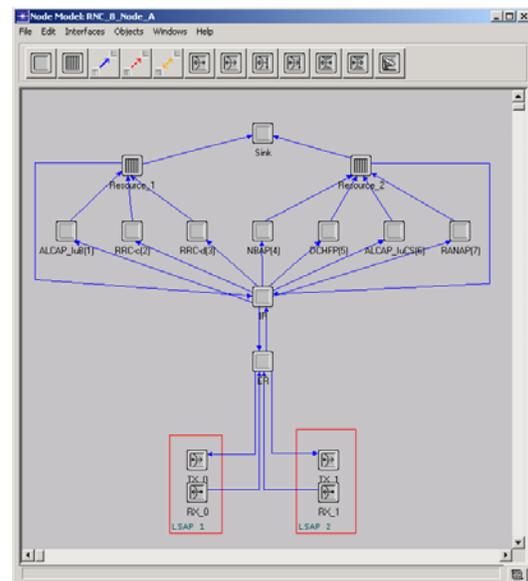


Fig. 5.2: Alternative RNC Node Architecture Mapping of FEs to Resources in OPNET

A detailed description about the implementation of the modeling concept using the OPNET Modeler simulation toolkit is provided in [Frangiadakis et al, 2002].

6. CONCLUSION

This paper introduces a general simulation methodology for the evaluation of different signaling architectures with respect to their performance. The approach is characterized by the automatic conveyance of the use cases under study to an event driven simulation environment.

There are many benefits of the proposed methodology. On the one hand the error prone transfer of MSCs into a simulation model is bypassed and a refinement of the use cases under study is assisted. On the other hand the approach is flexible and alternative MxRAN architectures under study can be modeled easily. Particularly the relocation of functional entities within one node is performed at the OPNET graphical user interface without difficulty. Thus the derivation of an optimized system architecture can be tackled in an iterative way and in a very early design stage so that the risk of capital investment into a suboptimal architecture is considerably reduced.

In the future the relocation of functional entities between different nodes is envisioned for the tool chain. Moreover it is intended to integrate and correlate all traffic sources within a Central Intelligent Traffic Control node. Inside this central traffic source node also additional information on occupied resources like radio channel elements or links can be kept. Besides the modeling of thousands of nodes will be incorporated as signaling background traffic by means of a stochastic process that switches ON and OFF the related resources. For this reason processing resources are available for processing foreground traffic only in ON mode. In the same way user traffic shall be supported as background traffic. Moreover the incorporation of resource specific state information for scarce resources like maximum number of channels is envisaged.

In general the devised approach and tool implementation helps to design complex next generation communication systems. This pertains to an optimized architecture choice in terms of signaling overhead, scalability and performance.

7. ACKNOWLEDGEMENTS

The work reported in this paper results from the cooperation of the Universities of Essen and Athens as well as Lucent Technologies.

The authors also want to acknowledge the contributions of Nikos Frangiadakis, Dimitra Kralli and Alexandra Tatsi (University of Athens) to the

overall modeling concept which have been provided within this project.

Part of the work presented here is funded by the German ministry of research and education (BMBF) within the IPonAir project.

8. REFERENCES

“IPonAir homepage”. <http://www.iponair.de>.

SDL Forum 2001, “MSC-2000 MESSAGE SEQUENCE CHART (MSC)”. (revised in 2001), SDL Forum Version of Z.120 (11/99) rev. 1(14/11/01).

3GPP TSG RAN 2002, "UTRAN Functions, examples on signaling procedures (Release 1999)". TS 25.931, v. 3.6.0, March 2002.

Frangiadakis N., Nikolaidis G., Schefczik P., Wiedemann A. 2002, “MxRAN Functional Architecture Performance Modeling”. *In Proc. OPNETWORK 2002 Conf.* (Washington D.C., USA, August).

“OPNET homepage”. <http://www.opnet.com>.

Kaarainen H., Ahtiainen A., Laitinen L., Naghian S., Niemi V. 2001, “UMTS Networks: Architecture, Mobility and Services”. John Wiley & Sons, Ltd., Chichester, England.

Uskela S. 2003, “Key Concepts for Evolution toward beyond 3G Networks”. *IEEE Wireless Communications Mag.* February 2003. Pp43-48.

Yungsoo K., Byung J., Jaehak C., Chan-Soo H., Ryu J. S., Ki-Ho K., Young K. 2003, “Beyond 3G: vision, requirements, and enabling technologies”, *IEEE Communications Mag.* March 2003. Pp120-124.

BIOGRAPHY:



Anja Wiedemann is a research assistant in the Computer Networking Technology Group at the Institute for Experimental Mathematics of the University of Essen, Germany. Her interests are in performance analysis, protocol modeling and simulation of wireless packet data networks. Mrs.

Wiedemann received her diploma degree in business informatics from the University of Essen and is now working towards her Ph.D.

A CHARACTERIZATION OF PRODUCT-FORM STATIONARY DISTRIBUTIONS FOR QUEUEING SYSTEMS IN RANDOM ENVIRONMENT

ANTONIS ECONOMOU

*University of Athens, Department of Mathematics
Panepistemioupolis, Athens 15784, Greece
aeconom@math.uoa.gr*

Abstract: We consider continuous-time Markov chains representing queueing systems in random environment and we obtain necessary and sufficient conditions for having product-form stationary distributions. The related topics of partial balance and the ESTA property are also studied. As an illustration, we apply the results to study the stationary distributions of Jackson networks in random environment.

For models that do not satisfy the product-form conditions, we develop a product-form approximation, which is proved to be very good for models evolving in a slowly changing random environment. We justify this fact and we propose a methodology for estimating the error of this approximation.

Keywords: queueing; Markov chains; random environment; stationary distribution; product form

1. INTRODUCTION

In many applications of queueing we find systems that evolve in random environment. The random environment may model the irregularity of the arrival process (for example when there are rush-hour phenomena), the irregularity of the service mechanism (due to servers' breakdowns, servers' vacations, availability of resources etc.) or both. Many authors have studied the properties of such models (see e.g. Neuts (1981), Gaver et al. (1984), O' Cinneide and Purdue (1986), Falin (1996) etc.). The reported results concern either qualitative properties or computational issues. The focus of the present work is on a computational issue and more specifically on the problem of computing the stationary distribution of a Markovian queueing system in random environment. Several authors have considered the same problem using various approaches. Matrix-analytic and transform methods have been used extensively. However, although in many cases the above methods give very satisfactory results, their implementation is computationally very demanding. The reason is that they require strong computational power to perform a great number of matrix operations. As the environmental state space grows large, the numerical complexity of the underlying algorithms increases rapidly and the efficient

implementation of these methods becomes very difficult.

To avoid the computational burden of the above methods, several authors have tried to identify some categories of models for which the stationary distributions assumes a simple product form. Although product-form stationary distributions and the related phenomenon of partial balance have been extensively studied within the framework of queueing networks, there are only few papers that apply these ideas to queueing systems in random environment. More specifically Sztrik (1987), Zhu (1994) and Falin (1996) have identified conditions that ensure product-form stationary distributions for several concrete classes of queueing systems in random environment. In the present paper we study the same problem within a general framework and we state necessary and sufficient conditions for product-form. Moreover, whenever these conditions fail, we develop a product-form approximation which is very good for queueing systems evolving in a slowly changing environment. We also study the relationship between the stationary distribution of a given Markovian queueing model and the stationary distributions of its embedded chains at environmental change epochs. Thus, we also study the Events See Time Averages (ESTA) property for the class of Markovian queueing systems in random environment.

To be concrete, we now define a general structure for a continuous-time Markov chain in a random environment. The model is an ergodic (i.e. irreducible and positive recurrent) Markov chain $\{(E(t), X(t)): t \geq 0\}$ with state space $\mathbf{E} \times \mathbf{X}$, where $\{E(t)\}$, $\{X(t)\}$ represent the random environment and the queueing process of interest respectively. We assume that $\{E(t)\}$ jumps from state to state according to an ergodic continuous-time Markov chain with transition rate matrix $\mathbf{Q}_E = (q_E(e, e'): e, e' \in \mathbf{E})$. In the meantime between two successive environmental transitions, the process $\{X(t)\}$ is governed by a transition matrix $\mathbf{Q}_{X|E}(e) = (q_{X|E}(x, x'|e): x, x' \in \mathbf{X})$ of an irreducible Markov chain on \mathbf{X} , where e is the current environmental state. More specifically the transition rates $q((e, x), (e', x'))$ of $\{(E(t), X(t))\}$ are given by

$$\begin{aligned} q((e, x), (e', x')) &= \begin{cases} q_{X|E}(x, x'|e), & \text{if } e' = e, x' \neq x \\ q_E(e, e'), & \text{if } e' \neq e, x' = x. \end{cases} \quad (1) \end{aligned}$$

Let $\bar{\pi} = (\pi(e, x): e \in \mathbf{E}, x \in \mathbf{X})$ be the joint stationary distribution of $\{(E(t), X(t))\}$ and $\bar{\pi}_E = (\pi_E(e): e \in \mathbf{E})$, $\bar{\pi}_X = (\pi_X(x): x \in \mathbf{X})$ its marginal distributions. The (full) balance equations are

$$\begin{aligned} &\pi(e, x) \left(\sum_{e' \neq e} q_E(e, e') + \sum_{x' \neq x} q_{X|E}(x, x'|e) \right) \\ &= \sum_{e' \neq e} \pi(e', x) q_E(e', e) \\ &+ \sum_{x' \neq x} \pi(e, x') q_{X|E}(x', x|e), \quad e \in \mathbf{E}, x \in \mathbf{X}. \quad (2) \end{aligned}$$

By summing these equation over x for every environmental state e we obtain after some easy manipulations that

$$\begin{aligned} &\pi_E(e) \sum_{e' \neq e} q_E(e, e') \\ &= \sum_{e' \neq e} \pi_E(e') q_E(e', e), \quad e \in \mathbf{E}. \quad (3) \end{aligned}$$

Hence, the marginal distribution $\bar{\pi}_E = (\pi_E(e))$ is the stationary distribution of the Markov chain with transition rate matrix $\mathbf{Q}_E = (q_E(e, e'))$.

Let $\mathbf{P}_{X|E}^{(t)}(e) = (p_{X|E}^{(t)}(x, x'|e): x, x' \in \mathbf{X})$ be the transition probability matrix at time t for the Markov chain with rate matrix $\mathbf{Q}_{X|E}(e)$ and $\bar{\pi}_{X|E}(e) = (\pi_{X|E}(x|e): x \in \mathbf{X})$ its stationary distribution (in the ergodic case in which it exists and is unique). We are interested in

determining $\bar{\pi}$, $\bar{\pi}_X$ and in examining their relationships with the transition rate matrices \mathbf{Q}_E and $\mathbf{Q}_{X|E}(e)$, $e \in \mathbf{E}$. We are also interested in studying the Palm (or embedded) distributions of $\{X(t)\}$ just after (or before) certain environmental transitions. For every $e \in \mathbf{E}$ let $X_{a(e)}(n)$ be the state of $\{X(t)\}$ just after the n -th environmental arrival to e and $X_{d(e)}(n)$ be the state of $\{X(t)\}$ just before the n -th environmental departure from e . Moreover, let $\bar{\pi}_{a(e)} = (\pi_{a(e)}(x): x \in \mathbf{X})$ and $\bar{\pi}_{d(e)} = (\pi_{d(e)}(x): x \in \mathbf{X})$ be the stationary distributions of $\{X_{a(e)}(n)\}$ and $\{X_{d(e)}(n)\}$ respectively. Define $q_E(e) = \sum_{e' \neq e} q_E(e, e')$. Because of (3) we have that

$$\sum_y \sum_{e' \neq e} \pi(e', y) q_E(e', e) = \pi_E(e) q_E(e).$$

Since the rate from state (e, x) to (e', x') is $\pi(e, x) q((e, x), (e', x'))$ we have that

$$\begin{aligned} \pi_{a(e)}(x) &= \frac{\sum_{e' \neq e} \pi(e', x) q_E(e', e)}{\sum_y \sum_{e' \neq e} \pi(e', y) q_E(e', e)} \\ &= \frac{\sum_{e' \neq e} \pi(e', x) q_E(e', e)}{\pi_E(e) q_E(e)}, \quad x \in \mathbf{X} \quad (4) \end{aligned}$$

and

$$\begin{aligned} \pi_{d(e)}(x) &= \frac{\sum_{e' \neq e} \pi(e, x) q_E(e, e')}{\sum_y \sum_{e' \neq e} \pi(e, y) q_E(e, e')} \\ &= \frac{\pi(e, x)}{\pi_E(e)}, \quad x \in \mathbf{X}. \quad (5) \end{aligned}$$

It is known that the Palm distributions of a process that correspond to different sets of transitions do not coincide with each other nor do they coincide with the stationary distribution of the process in general. In such cases it is important to study the relationships of these distributions and also to find conditions under which they do coincide (Events See Time Averages (ESTA) property).

2. CHARACTERIZATION OF PRODUCT-FORM DISTRIBUTIONS

Equations (2) are decomposed to the following partial balance equations that are not satisfied by $\bar{\pi}$ in general:

$$\begin{aligned} &\pi(e, x) \sum_{e' \neq e} q_E(e, e') \\ &= \sum_{e' \neq e} \pi(e', x) q_E(e', e), \quad e \in \mathbf{E}, x \in \mathbf{X} \quad (6) \end{aligned}$$

and

$$\begin{aligned} \pi(e, x) & \sum_{x' \neq x} q_{X|E}(x, x' | e) \\ & = \sum_{x' \neq x} \pi(e, x') q_{X|E}(x', x | e), \quad e \in \mathbf{E}, \quad x \in \mathbf{X}. \end{aligned} \quad (7)$$

The phenomenon of partial balance and its implications have been extensively studied in the literature (see e.g. Kelly (1979)). It has been generally noted that the presence of partial balance facilitates the study of a given model. First, it implies the equality of the Palm distributions at certain event epochs (see e.g. Kelly (1979) Ch. 9 and Fakinos and Economou (1998)). Second, under certain additional conditions, it implies that the stationary distribution assumes a certain product form. In characterizing the phenomenon of partial balance and product-form for the general model (1), we use the following result that gives the stationary distribution $\bar{\pi}$ in terms of the Palm distributions $\bar{\pi}_{a(e)}$, $e \in \mathbf{E}$. For a proof see Economou (2002).

Proposition 1 (*Inversion formula*) *Given the Palm distributions $\bar{\pi}_{a(e)}$, the stationary distribution $\bar{\pi}$ can be computed by*

$$\pi(e, x) = \pi_E(e) q_E(e) \int_0^\infty e^{-q_E(e)t} \pi_{a(e)}^{(t)}(x) dt \quad (8)$$

where $\bar{\pi}_{a(e)}^{(t)} = (\pi_{a(e)}^{(t)}(x) : x \in \mathbf{X})$ is the transient probability function at time t of a Markov chain with initial distribution $\bar{\pi}_{a(e)}$ and transition rate matrix $\mathbf{Q}_{X|E}(e)$.

We are now in position to investigate the phenomenon of partial balance within the framework of our model.

Theorem 2 *For the general model with transition rates given by (1) the following are equivalent:*

- (i) *The Palm distributions $\bar{\pi}_{a(e)}$ and $\bar{\pi}_{d(e)}$ coincide for every $e \in \mathbf{E}$.*
- (ii) *The stationary $\bar{\pi}$ satisfies the partial balance equations (6).*
- (iii) *The stationary $\bar{\pi}$ satisfies the partial balance equations (7).*

If moreover the transition matrices $\mathbf{Q}_{X|E}(e)$ are ergodic with stationary distributions $\bar{\pi}_{X|E}(e)$, $e \in \mathbf{E}$ then (i)-(iii) are also equivalent to:

- (iv) *The distributions $\bar{\pi}_{d(e)}$ and $\bar{\pi}_{X|E}(e)$ coincide for every $e \in \mathbf{E}$.*

(v) *The distributions $\bar{\pi}_{a(e)}$ and $\bar{\pi}_{X|E}(e)$ coincide for every $e \in \mathbf{E}$.*

(vi) *The stationary distribution $\bar{\pi}$ is given by the product-form formula*

$$\pi(e, x) = \pi_E(e) \pi_{X|E}(x | e), \quad e \in \mathbf{E}, \quad x \in \mathbf{X}. \quad (9)$$

Proof. (i) \Leftrightarrow (ii) The probabilities $\pi_{a(e)}(x)$ and $\pi_{d(e)}(x)$ given by (4) and (5) are respectively equal to the right and the left side of the partial balance equations (6) divided by $\pi_E(e) q_E(e)$.

(ii) \Leftrightarrow (iii) Immediate, in light of the full balance equations (2).

(iii) \Rightarrow (vi) Consider a fixed $e \in \mathbf{E}$. Because of (7) we have that the vector $(\pi(e, x) : x \in \mathbf{X})$ satisfies the balance equations of the Markov chain with transition rate matrix $\mathbf{Q}_{X|E}(e)$. Due the ergodicity of $\mathbf{Q}_{X|E}(e)$, we have that $(\pi(e, x) : x \in \mathbf{X})$ is a scalar multiple of the stationary distribution $\bar{\pi}_{X|E}(e)$, i.e. $\pi(e, x) = c(e) \bar{\pi}_{X|E}(e)$, $x \in \mathbf{X}$. By summing over x we obtain $c(e) = \pi_E(e)$; hence $\pi(e, x)$ assumes the form (9).

(iv) \Rightarrow (vi) Immediate using (5).

(v) \Rightarrow (vi) Since $\bar{\pi}_{a(e)}$ and $\bar{\pi}_{X|E}(e)$ coincide, the Inversion formula (8) assumes the form

$$\begin{aligned} \pi(e, x) & = \pi_E(e) q_E(e) \\ & \cdot \int_0^\infty e^{-q_E(e)t} \pi_{X|E}^{(t)}(x | e) dt, \quad e \in \mathbf{E}, \quad x \in \mathbf{X}. \end{aligned} \quad (10)$$

But $(\pi_{X|E}(x | e) : x \in \mathbf{X})$ is the stationary distribution of $\mathbf{Q}_{X|E}(e)$ and we have that $\pi_{X|E}^{(t)}(x | e) = \pi_{X|E}(x | e)$, $t \geq 0$, $e \in \mathbf{E}$, $x \in \mathbf{X}$. Equation (10) is reduced to (9).

(vi) \Rightarrow (iii), (iv), (v) If the stationary distribution $\bar{\pi}$ is given by (9) then we have obviously that the partial balance equations (7) hold, i.e. (iii) is valid. Moreover, by (5) we have that $\pi_{d(e)}(x) = \pi_{X|E}(x | e)$, $x \in \mathbf{X}$, i.e. (iv) is valid. We have also that (i) holds because of the implication (iii) \Rightarrow (i) that has already been proved. Hence $\pi_{a(e)}(x) = \pi_{d(e)}(x) = \pi_{X|E}(x | e)$, $e \in \mathbf{E}$, $x \in \mathbf{X}$, i.e. (v) is valid.

The above result characterizes completely the partial balance, the product form and the ESTA properties for the model (1). However, we see that the conditions that imply a product-form stationary distribution are very

restrictive. We now consider a 'perturbed' model that has always a product-form distribution. We have the following.

Theorem 3 Consider a continuous-time Markov chain with state-space $\mathbf{E} \times \mathbf{X}$ and transition rates

$$\tilde{q}((e, x), (e', x')) = \begin{cases} q_{X|E}(x, x'|e), & \text{if } e' = e, x' \neq x \\ q_E(e, e')\pi_{X|E}(x'|e), & \text{if } e' \neq e, x' \in \mathbf{X} \end{cases} \quad (11)$$

where $q_{X|E}(x, x'|e)$, $q_E(e, e')$ and $\pi_{X|E}(x|e)$ are the same as in the model (1). Then its stationary distribution is given by the product-form formula

$$\tilde{\pi}(e, x) = \pi_E(e)\pi_{X|E}(x|e), \quad e \in \mathbf{E}, x \in \mathbf{X}.$$

Proof. The balance equations of the model are

$$\begin{aligned} \tilde{\pi}(e, x) & \left(\sum_{e' \neq e} \sum_{x' \in \mathbf{X}} q_E(e, e')\pi_{X|E}(x'|e) \right. \\ & \quad \left. + \sum_{x' \neq x} q_{X|E}(x, x'|e) \right) \\ & = \sum_{e' \neq e} \sum_{x' \in \mathbf{X}} \tilde{\pi}(e', x')q_E(e', e)\pi_{X|E}(x|e) \\ & \quad + \sum_{x' \neq x} \tilde{\pi}(e, x')q_{X|E}(x', x|e). \end{aligned} \quad (12)$$

By direct substitution we see that the distribution $(\tilde{\pi}(e, x) : e \in \mathbf{E}, x \in \mathbf{X})$ satisfies the equations (12); hence it is the stationary distribution of the model.

Whenever the transition rates $q_E(e, e')$ of the environmental process $\{E(t)\}$ are small, the rates $\tilde{q}((e, x), (e', x'))$ of the perturbed model (11) are very close to the rates $q((e, x), (e', x'))$ of the original model (1). Hence the stationary distributions of the two models are expected to be also very close to each other and we conclude that the product-form distribution of the modified model (11) is indeed a good approximation for the stationary distribution of the original model (1). Thus, this product-form distribution is a legitimate approximation for queueing systems evolving in a slowly changing environment. More importantly, in the context of specific concrete models we can estimate the error of the approximation using the results obtained by van Dijk (1992). These results provide error bounds for the approximation of the stationary distribution of a given model by the stationary distribution of a perturbed model in terms of the differences of their transition rates, using a Markov reward approach.

3. AN APPLICATION TO JACKSON NETWORKS IN RANDOM ENVIRONMENT

As an illustration of the main result, we present its application in the study of Jackson networks in random environment. A Jackson network in random environment is a continuous-time Markov chain on $\mathbf{E} \times \mathbf{Z}_+^J$ with transition rates given by (1) and matrices $\mathbf{Q}_{X|E}(e)$, $e \in \mathbf{E}$ corresponding to Jackson networks, i.e.

$$q_{X|E}(\bar{x}, \bar{x}'|e) = \begin{cases} \lambda(e)p_{0j}(e) & \text{if } \bar{x}' = \bar{x} + \bar{e}_j \\ \mu_i(x_i|e)p_{ij}(e) & \text{if } \bar{x}' = \bar{x} - \bar{e}_i + \bar{e}_j \\ \mu_i(x_i|e)p_{i0}(e) & \text{if } \bar{x}' = \bar{x} - \bar{e}_i \\ 0 & \text{otherwise,} \end{cases} \quad (13)$$

where by $\bar{x} = (x_1, x_2, \dots, x_J)$ we denote a generic state of the network representing the queue lengths at the J stations and \bar{e}_j is the j -th unit vector with J components (with 1 in the j -th position and 0 elsewhere).

Therefore, in any time interval during which the environmental process $\{E(t)\}$ is in state e , the network operates as follows: Customers arrive at the network according to a Poisson process with rate $\lambda(e)$. An arriving customer goes to the j -th station with probability $p_{0j}(e)$ ($j = 1, 2, \dots, J$). The service at the i -th station of the network is offered at exponential rate $\mu_i(x_i|e)$ which depends on the number x_i of the present customers at that same station ($i = 1, 2, \dots, J$). Upon completing service at the i -th station, a customer is routed to station j with probability $p_{ij}(e)$ or leaves the network with probability $p_{i0}(e)$, ($i, j = 1, 2, \dots, J$). For every fixed $e \in \mathbf{E}$, the discrete-time Markov chain with transition probabilities $p_{ij}(e)$ ($i, j = 1, 2, \dots, J$) is supposed to be irreducible. This implies that the traffic equations

$$\alpha_j(e) = \lambda(e)p_{0j}(e) + \sum_{i=1}^J \alpha_i(e)p_{ij}(e), \quad j = 1, 2, \dots, J \quad (14)$$

have a unique positive solution $\bar{\alpha}(e) = (\alpha_1(e), \alpha_2(e), \dots, \alpha_J(e))$. Moreover, all the arrival and service processes are assumed independent. For a fixed e , the Markov chain representing a Jackson network with rates

given by (13) is positive recurrent if and only if

$$B_j^{-1}(e) = 1 + \sum_{x_j=1}^{\infty} \frac{\alpha_j(e)^{x_j}}{\mu_j(1|e)\mu_j(2|e)\dots\mu_j(x_j|e)} < \infty, \quad j = 1, 2, \dots, J. \quad (15)$$

The stationary distribution is then given by

$$\pi_{X|E}(\bar{x}|e) = \prod_{j=1}^J B_j(e) \frac{\alpha_j(e)^{x_j}}{\mu_j(1|e)\mu_j(2|e)\dots\mu_j(x_j|e)}. \quad (16)$$

Zhu (1994) proved from scratch a sufficient condition for product form. Using Theorem 2 we can easily show the necessity and the sufficiency of that condition for product form.

Corollary 4 *Let $\{E(t), \bar{X}(t)\}$ be a Jackson network in random environment with transition rates given by (1) and (13). For every $e \in \mathbf{E}$, let $\bar{\alpha}(e) = (\alpha_1(e), \alpha_2(e), \dots, \alpha_J(e))$ be the unique solution of the system of equations (14) and assume that the stability condition (15) holds. The following are equivalent:*

- (i) $\alpha_j(e)/\mu_j(x_j|e)$ is independent of e , for all $j = 1, 2, \dots, J$ and $x_j \geq 1$.
- (ii) The stationary distribution $\bar{\pi}$ is given by the product-form formula

$$\pi(e, \bar{x}) = \pi_E(e) \prod_{j=1}^J B_j(e) \frac{\alpha_j(e)^{x_j}}{\mu_j(1|e)\mu_j(2|e)\dots\mu_j(x_j|e)}, \quad e \in \mathbf{E}, \quad \bar{x} \in \mathbf{Z}_+^J, \quad (17)$$

where $(\pi_E(e) : e \in \mathbf{E})$ is the stationary distribution of a Markov chain with transition rates $(q_E(e, e'))$ and $B_j(e)$ are given by (15), i.e. all the equivalent conditions (i)-(vi) of Theorem 2 hold.

Proof. (i) \Rightarrow (ii) Suppose that (i) holds. Then by direct substitution we can show that the distribution given by (17) satisfies the balance equations (2). Indeed, using the fact that $(\pi_E(e) : e \in \mathbf{E})$ and $(\pi_{X|E}(\bar{x}|e) : x \in \mathbf{Z}_+^J)$ are the stationary distributions of \mathbf{Q}_E and $\mathbf{Q}_{X|E}(e)$ respectively, we have that

$$\begin{aligned} \pi_E(e) \pi_{X|E}(\bar{x}|e) & \cdot \left(\sum_{e' \neq e} q_E(e, e') + \sum_{\bar{x}' \neq \bar{x}} q_{X|E}(\bar{x}, \bar{x}'|e) \right) \\ & = \pi_{X|E}(\bar{x}|e) \sum_{e' \neq e} \pi_E(e') q_E(e', e) \end{aligned}$$

$$+ \pi_E(e) \sum_{\bar{x}' \neq \bar{x}} \pi_{X|E}(\bar{x}'|e) q_{X|E}(\bar{x}', \bar{x}|e). \quad (18)$$

But by condition (i) and (15) we obtain that $B_j^{-1}(e)$ is independent of e for all $j = 1, 2, \dots, J$. Hence by (16) we conclude that $\pi_{X|E}(\bar{x}|e)$ is independent of e for all \bar{x} . Then (18) assumes the form

$$\begin{aligned} \pi_E(e) \pi_{X|E}(\bar{x}|e) & \cdot \left(\sum_{e' \neq e} q_E(e, e') + \sum_{\bar{x}' \neq \bar{x}} q_{X|E}(\bar{x}, \bar{x}'|e) \right) \\ & = \sum_{e' \neq e} \pi_E(e') \pi_{X|E}(\bar{x}|e') q_E(e', e) \\ & \quad + \pi_E(e) \sum_{\bar{x}' \neq \bar{x}} \pi_{X|E}(\bar{x}'|e) q_{X|E}(\bar{x}', \bar{x}|e). \end{aligned}$$

i.e. the distribution $(\pi(e, x))$ given by (17) satisfies the balance equations (2).

(ii) \Rightarrow (i) By Theorem 2 (vi) \Rightarrow (ii) we have that for every \bar{x} the vector $(\pi(e, \bar{x}) : e \in \mathbf{E})$ satisfies the balance equations for the process $\{E(t)\}$. Hence $(\pi(e, \bar{x}) : e \in \mathbf{E})$ is a scalar multiple of $(\pi_E(e) : e \in \mathbf{E})$ and we conclude that $\pi(e, \bar{x}) = \phi(\bar{x}) \pi_E(e)$, $e \in \mathbf{E}$, $\bar{x} \in \mathbf{Z}_+^J$. Then for every $j = 1, 2, \dots, J$ and $x_j \geq 1$ we have

$$\frac{\alpha_j(e)}{\mu_j(x_j|e)} = \frac{\pi(e, \bar{x})}{\pi(e, \bar{x} - \bar{e}_j)} = \frac{\phi(\bar{x})}{\phi(\bar{x} - \bar{e}_j)},$$

i.e. $\alpha_j(e)/\mu_j(x_j|e)$ is independent of e .

REFERENCES

1. Economou, A. (2002) A relation between time-averages and event-averages for Markov chains in random environment.
2. Fakinos, D. and Economou, A. (1998) Overall station balance and decomposability for non-Markovian queueing networks. *Adv. Appl. Prob.* **30**, 870-887.
3. Falin, G. (1996) A heterogeneous blocking system in a random environment. *J. Appl. Prob.* **33**, 211-216.
4. Gaver, D. P., Jacobs, P. A. and Latouche, G. (1984) Finite birth-and-death models in randomly changing environments. *Adv. Appl. Prob.* **16**, 715-731.
5. Gelenbe, E. and Rosenberg, C. (1990) Queues with slowly varying arrival and service processes. *Management Science* **36**, 928-937.

6. Kelly, F. P. (1979) *Reversibility and Stochastic Networks*. Wiley.
7. Neuts, M. F. (1981) *Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach*. The Johns Hopkins University Press, Baltimore.
8. O'Cinneide, C. A. and Purdue, P. (1986) The M/M/ ∞ queue in random environment. *J. Appl. Prob.* **23**, 175-184.
9. Sztrik, J. (1987) On the heterogeneous M/G/n blocking system in a random environment. *J. Operat. Res. Soc.* **38**, 57-63.
10. van Dijk, N. M. (1992) An error bound theorem for approximate Markov chains. *Prob. Eng. Inf. Sci.* **6**, 413-424.
11. Zhu, Y. (1994) Markovian queueing networks in a random environment. *Oper. Res. Letters* **15**, 11-17.

BIOGRAPHY

Antonis Economou received the MA degree in Pure Mathematics (1994) from the University of California, Los Angeles and the MSc degree in Statistics and OR (1997) and the PhD degree in Mathematics (1998) from the University of Athens, Greece. During 1999-2001 he was visiting faculty member at the Applied Mathematics Department of the University of Crete. Since 2001 he has been a faculty member at the Mathematics Department of the University of Athens. His research interests include the performance evaluation and the control of queueing systems and computational and stochastic comparison problems for Markov chains.

SPECTRAL EFFICIENCY OF MQAM USING DIVERSITY TECHNIQUES

D. ALNSOUR, M. AL-AKAIDI

*School of Engineering and Technology
De Montfort University, Leicester
LE1 9BH, UK
Email: mma@dmu.ac.uk*

Abstract: The performance of Time Division Multiple Access (TDMA) is affected by various factors. Accurate coverage prediction, modulation and coverage control techniques can significantly improve the capacity. Spectrum efficiency and good coverage are the main objectives of radio system planning. The capacity of a cellular system is directly related to spectrum efficiency and is directly proportional to the cluster size, so that any decrease in the size indicates that co-channel cells are located much closer together. This allows more frequency channels to be reused taking into account minimum co-channel interference (CCI). The above factors have an impact on the capacity, it is therefore crucial to include them in the evaluation to minimize the assumptions made. This paper shows development of a Monte Carlo simulation model which accurately assesses the performance of cellular systems.

Keywords: Modulation, Efficiency, Diversity, Co-channel, Interference, Spectrum

1 INTRODUCTION

The radio spectrum is a finite resource and it is important that it is exploited efficiently by all users. Accordingly modulation scheme used for mobile environment should utilise the RF channel bandwidth and the transmitted power as efficiently as possible. This is due to the fact that the mobile radio channel is power and bandwidth limited, which implies the need to investigate different digital modulation, schemes under different propagation channels. A typical application of such results would be the choice of modulation technique for a digital mobile radio system in a specific environment.

This paper discusses the impact of the choice of a modulation scheme on the reuse distance. Since the value of the reuse distance has a decisive effect on the performance of the spectral Efficiency, a relation is set between the chosen modulation level and the spectral efficiency. Assuming the downlink case, the power at the mobile receiver from the desired base station (BS) and the co-channel BS's was calculated according to the propagation model used.

The CCI associated with a certain bit error rate (BER) for a particular modulation scheme will define the reuse distance, D . The value of D is the maximum value that complies with the constraint set by CCI ratio.

As detailed in [Parsons, 1992], selected propagation model for both desired and interfering signals are

applied. It is acceptable that frequencies are reused at distance D , therefore the area covered by the service of one set of these reused frequencies is roughly covered by, $\pi(D/2)^2$. The relation of the reuse distance D with radius R is given by the reuse distance factor, $R_u = D/R$ as shown in Figure 0.

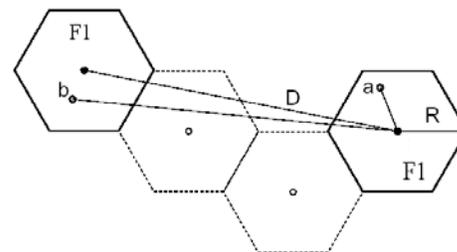


Figure 0: The relation of reuse distance D and radius R where a and b are mobile

Manhattan microcell, Cost231-Hata and Lee models were used to assess the capacity potential for high-level modulation schemes. In order to provide a comparative capacity evaluation, diversity vs. omni directional configurations where applied in different propagation models. These coverage control schemes are used to increase capacity by reducing the co-channel interference, thus the reuse distance D .

The spectrum efficiency evaluation technique described studies the effective area of each cell in a co-channel interference limited environment. By

specifying the BER performance, the corresponding carrier-to-interference ratio (CIR) can be obtained for a specific modulation scheme, thus the reuse distance is evaluated.

The following sections describe the capacity evaluation method and the simulation results obtained using different propagation models.

2 CHANNEL AND SYSTEM MODELS

Spectral efficiency is evaluated so that minimum assumptions are made. In this section, the propagation and interference models are explained.

2.1 Propagation models

Cell radius and path loss exponent are critical parameters, derived from the propagation model. These are assessed from the received signal in a specific environment.

The work done here describes three cells:

- 2 Macrocell, $R > 1\text{km}$, $S_{th} = 65\text{ dBm}$
- 3 Microcell, $300\text{m} > R > 1\text{ km}$, $S_{th} = 115\text{ dBm}$.
- 4 Microcell, $R < 300\text{m}$, $S_{th} = 115\text{ dBm}$.

Using different propagation models, the cell radius is determined by the predefined system threshold, S_{th} , below which the mobile receiver may not be able to detect the desired signal from noise [Holma and Toskala, 2000]. The path loss exponent 20 dB/dec can also be easily derived.

2.1.1 Lee Model

Lee's empirical propagation model is considered accurate enough and simple to use for macrocellular systems [Lee, 1990a]. Since this model is a semi statistical model, the results are also considered statistical. The model is used to predict the field strength of the received signal P_r which can be expressed as

$$P_r = P_o - \beta \log\left(\frac{d}{d_o}\right) - \eta \log\left(\frac{f_c}{900}\right) + \alpha_o$$

where f_c is the carrier frequency, P_o and η are the parameters found from empirical measurements at a 1.6 km point of interception, -64 dBm and -43.1 dBm respectively.

In this paper, these were taken for an urban area (Newark, USA), where β is the path loss exponent, d is distance in km from the transmitter, d_o is 1.6 km, and α_o is the correction factor for a different set of conditions.

2.1.2 Cost-231 Hata Model

This model can distinguish between three different environmental types. The model is expressed in terms of the carrier frequency f_c (in MHz), the base station antenna height h_b (between 30 and 200 meters), the mobile station antenna height h_m (between 1 and 10 meters), and the distance d (between 1 and 20 km) between transmitter and receiver. The urban path loss LU is given as $A + B \log_{10} d$, for urban areas with some correction factors for suburban areas (LSU) = $LU - 15.11$ and for open areas (LO) = $LU - 30.23$. The terms A and B are expressed as follows [6]:

$$A = 46.3 + 33.9 \log(f_c) - 13.2 \log(h_b) - a(h_m),$$

$$B = 44.9 + 6.55 \log(h_b),$$

where $a(h_m)$ depends on the city type; for small and medium cities:

$$a(h_m) = 1.1 \log_{10}(f_c - 0.7) * h_m - 1.56 \log_{10}(f_c - 0.8),$$

for large cities:

$$a(h_m) = 3.2(\log_{10}(11.75 * h_m))^2 - 7.97,$$

In our simulation, environment only large cities and urban areas is considered.

2.1.3 Manhattan Model

The line of sight path loss L_{LOS} is defined using this model for the microcell scenario, for a distance $d = 300$ [Holma and Toskala, 2000],

$$L_{LOS} = 82 + 40 \log_{10} \frac{d}{300}$$

From the previously described models, Table 1 shows the derived path loss exponent and radius.

Table 1: Path loss exponents and cell radius.

Propagation model	S_{th} (dBm)	Radius	Path loss exponent
Lee model	65	2.5 km	4.31
Cost231-hata	115	780 m	3.0647
manhattan	115	213 m	2.028

2.2 Interference Models

To simplify the analysis only first tier co-channel interference is taken into account. The desired user CIR is as follows,

$$CIR = \frac{S_d}{S_i} = \frac{S_d(r)}{\sum_{i=1}^{n_i} S_i(r_i)} \quad (1)$$

Where $S_d(r)$ is the desired power level from the desired mobile at a distance r from its BS, S_i is the total received interfering power level from the i^{th} interfering mobile at a distance r_i from the desired mobiles BS, and n_i is the number of active interferences, for the case of non-sectorized cellular systems $n_i=6$.

3 AREA SPECTRAL EFFICIENCY

The performance measure used in this paper is the area spectral efficiency, η_{area} , as defined by Alouini [Alouini and Goldsmith, 1999b] for FDMA/TDMA systems and is given by,

$$\eta_{area} = \frac{4}{\pi D^2} \int_{R_0}^R \log_2(1 + \gamma) p_r(r) dr \quad (2)$$

where D is the reuse distance at which the frequency are reused, R_0 corresponds to the closest distance the mobile can be from the BS antenna and R is the cell radius. The average area spectral efficiency $\bar{\eta}_{area}$ measured in [bits/s/Hz/km²] is defined as the sum of the maximum average data rates/Hz/Unit area for the system derived from Shannon capacity,

$$C=W \log(1+CIR)$$

where W is the total allocated bandwidth /cell [Lee, 1990b].

It should be noted that as spectral efficiency increases the constellation lattice becomes denser, hence, detection at the receiver becomes more difficult and BER may rise significantly. For this reason, greater power transmission is needed in order to maintain a specific quality of service. The higher transmitted power would raise the interference level in the system, which suggests larger reuse distance. Therefore by letting the CIR obtained from the modulation scheme

for a certain BER to determine the size of the cell, will lead to optimum results.

In cellular systems, modulation has a significant impact on the system capacity. The capacity can be obtained by estimating for each required CIR level which is determined by the BER requirement. In this paper all the simulation is performed for a BER performance of 0.001, the required co-channel interference ratio is obtained by using Figure 1 .

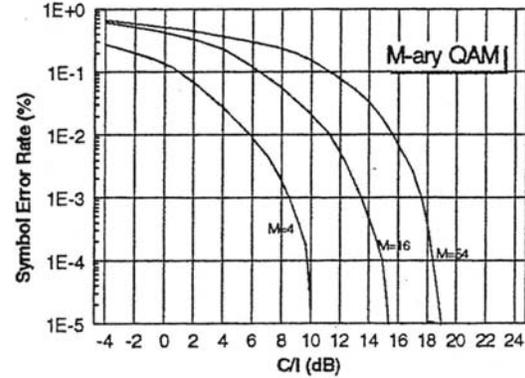


Figure 1: C/I required to meet BER requirement for different M.

The total bandwidth is determined from the following equation,

$$B_w = \frac{R_b(1 + \alpha)}{\beta}$$

where R_b is the bit rate, α is the excessive bandwidth often assumed 0.35 and β is the bandwidth efficiency measured in bits/sec/Hz.

With reference to the required signal to interference ratio, the six co-channel cells are moved towards the centre cell. The minimum reuse distance, D , corresponds to the point where the co-channel interference ratio is satisfied.

Using the evaluated cell radius, R , the reuse distance is obtained by estimating for each required co-channel interference level determined by the (BER) requirement.

The optimal reuse distance is based on the worst case configuration, where $r_i=D-R$. The simulated model also takes into account the effect of users' random location in their respective cells, the impact of propagation parameters, cell size, and cell sectorization is also considered. The spectral

efficiency is computed for the practical case where the user and the interferer are randomly allocated.

4 ADAPTIVE MQAM AND DIVERSITY

The radio channels in a wireless mobile communication system are affected by different types of fading (multipath, shadowing, etc), therefore, they will have negative effect on signals carried on these channels.

To compensate for these channel impairments imposed by fading, adaptive modulation scheme is used. In adaptive MQAM, information about the channel conditions at the receiver is fed back to the transmitter so that it will adjust its transmitted modulation level (constellation size) accordingly. This channel information is usually acquired by using a pilot signal or inserting a training sequence into the stream of MQAM data symbol to extract the channel induced attenuation and phase shift [Alouini and Goldsmith, 1999a].

In this work, the adaptive rate fixed-power MQAM system is combined with a well-known diversity combining techniques. In particular we use the maximal ratio combining (MRC) and selection combining (SC) of the received signal. The former requires the M signals to be weighted proportionately to their CIR and then summed coherently. Perfect knowledge of the branch amplitudes and phase is assumed. The disadvantage of MRC is that it requires knowledge of the branch parameters and independent processing of each branch. The PDF of the received CIR at the output of a perfect M-branch MRC is derived in in [Alouini and Goldsmith, 2000] to be:

$$P^{mrc}(\gamma) = \gamma^{M-1} e^{-\gamma/\bar{\gamma}} / (M-1)\bar{\gamma}^{-M} \quad (3)$$

In the SC technique only one of the M receivers having the highest baseband CIR is connected to the output. Unlike the MRC it does not require coherent reception.

The PDF of the received CIR at the output of M-branch is again derived in [Alouini and Goldsmith, 2000] and it is given by:

$$P^{sc}(\gamma) = \frac{M}{\gamma} (1 - e^{-\gamma/\bar{\gamma}})^{M-1} e^{-\gamma/\bar{\gamma}} \quad (4)$$

Assuming perfect coherent detection, thus the only source of error is noise and interference from the

channel BER is approximated by [Goldsmith and Chua, 1997]

$$BER(M, \gamma) \approx 0.2 \exp\left(-\frac{3\gamma}{2(M-1)}\right) \quad (5)$$

where γ is the CIR.

For given CIR and assuming ideal Nyquist pulses the spectral efficiency of a continuous rate MQAM can be approximated by inverting Eqn. 5 giving;

$$\eta = \log 2(M) = \log 2\left(1 + \frac{3\gamma}{2K}\right) \quad (6)$$

Where $K = -\ln(5BER)$

In practice the CIR is not fixed, it is rather fluctuating due to channel impairments. Therefore the area spectral efficiency is calculated by integrating the above equation over the distribution of CIR and substituting in Eqn. 2. In this case we integrate over MRC distribution function to yield the following:

$$\eta_{area}^{mrc} \approx \frac{4}{\pi D^2 \log_2(2)} \times P_M\left(\frac{-1}{\gamma_o}\right) \left(-E + \ln \bar{\gamma}_o + \frac{1}{\gamma_o}\right) + \sum_{k=1}^{M-1} \frac{P_k\left(\frac{-1}{\gamma_o}\right) - P_{M-k}\left(\frac{-1}{\gamma_o}\right)}{k} \quad (7)$$

Where $\bar{\gamma}_o = \frac{3}{2K} \bar{\gamma}$, and $\bar{\gamma}$ is the average received CIR.

In the case of SC diversity, an alternative diversity combining technique, we integrate for a selection combining distribution function to yield the following approximation [Alouini and Goldsmith, 2000]

$$\eta_{area}^{sc} \approx \frac{4M}{\pi D^2 \log_2(2)} \sum_{k=0}^{M-1} \frac{(-1)^k}{1+k} \binom{M-1}{k} \times e^{(1+k)/\bar{\gamma}} \left[E + \ln\left(\frac{1+k}{\gamma}\right) - \left(\frac{1+k}{\gamma}\right) \right] \quad (8)$$

Where the binomial coefficient is given as,

$$\binom{M-1}{k} = \frac{(M-1)!}{(M-k-1)!k!}$$

The results have shown a significant increase when using diversity combining technique; these will be shown in details in the following section.

5 SIMULATION SETUP AND RESULTS

In this section, the effects of the different propagation parameters, cell size and modulation level is computed on the spectral efficiency.

Monte Carlo method is applied to determine the rates for the user. The user position is randomly picked and the CIR is evaluated accordingly. After 30,000 repetitions, the average spectral efficiency is evaluated. In the case of MRC and SC diversity, number of branches assumed is 4.

Figure 3-5 depicts the effect of MRC and SC diversity on macrocells and microcells. For all cell sizes, the best performance can be seen using MRC diversity technique. It should also be noted that higher modulation level reduces the performance since they require higher CIR values. This will lead to larger sizes to mitigate the increased interference.

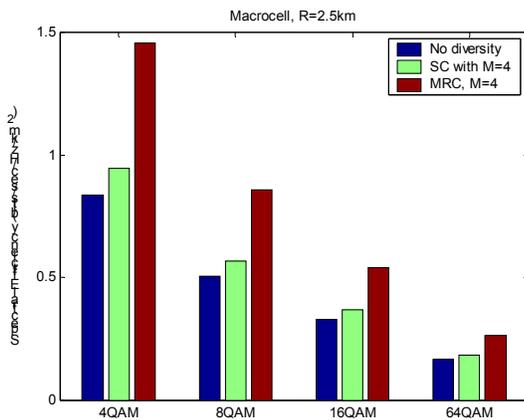


Figure 3: Modulation and diversity effects on system capacity in macrocell environment.

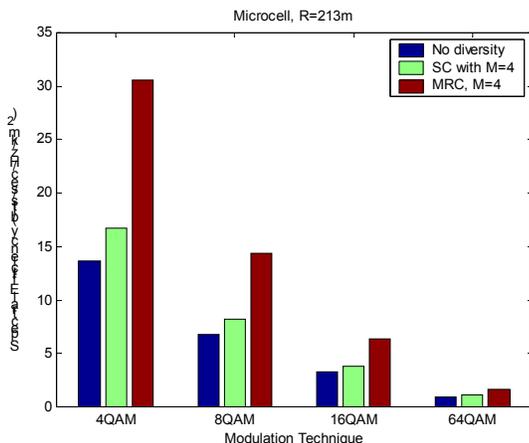


Figure 4: Modulation and diversity effects on system capacity in microcell environment.

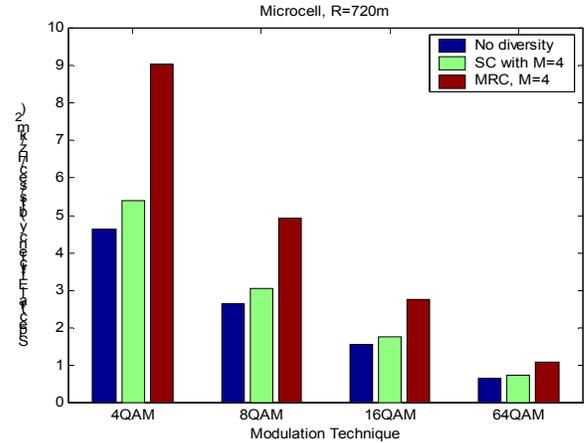


Figure 5: Modulation and diversity effects on system capacity in microcell environment.

6 CONCLUSION

Higher modulation and different antenna pattern using diversity were investigated using the developed capacity evaluation technique. In order, to improve system capacity, the factors, which are limiting the system performance, must be identified. It is also evident, that the best performance was obtained when a microcell is considered. Thus, the radius of the cell is inversely proportional to the spectral efficiency.

In a microcell, the performance of M-QAM degrades as the level increases. Despite better bandwidth efficiency at higher level modulation, more power is needed to maintain energy to noise, E_b/N_0 , ratio to achieve a fixed BER at the receiver. The increase in power will require larger cell sizes to mitigate the increased interference level, thus reducing the spectral efficiency of the system. Consequently, 4-QAM=QPSK is the best modulation scheme to achieve high capacity, nonetheless if the modulation level is controlled according to the CIR very high spectral efficiency can be expected. MRC significantly increased the performance of the system when using M=4.

In a macrocell, a higher level modulation scheme may be used to take advantage of higher transmitted power around the base site. A lower level modulation can be accommodated when the receiving power level is low. According to distance from the base station, different modulation schemes are adapted to site coverage, thus improving the system performance.

Finally, the improvement on the factors mentioned in this paper and the investigation of different

parameters, results in an overall capacity improvement.

7 REFERENCES

- 1) Parsons D. 1992, *The Mobile Radio Propagation Channel*, Pentech Press.
- 2) Al-Akaidi M., Alnsour D., Urwin P., Hammuda H. 2001, "Capacity Evaluation in Cellular Systems". In *IEE 3G2001*, March London, pp.175-179.
- 3) Holma H. and Toskala A. 2000, *WCDMA for UMTS*, Wiley.
- 4) Lee W. C. 1990, *Mobile Cellular Telecommunication System*, McGraw Hill.
- 5) Lee W. C. 1990, "Estimate of channel capacity in Rayleigh fading environment". In *IEEE transaction. Veh. Technology*, August, vol. VT-39, pp187-190.
- 6) Blaunstein N., *Radio Propagation in Cellular Networks*, Artech House, 1999.
- 7) Alouini M. and Goldsmith A 1999, "Capacity of Rayleigh fading channels under different adaptive transmission and diversity combining techniques". In *Trans. On Vehicular Technology*, July vol. 48, no.4.
- 8) Alouini M. and Goldsmith A 1999, "Area Spectral Efficiency of cellular mobile radio systems",. In *IEEE Trans. On Veh. Technology*, vol.48, no4.
- 9) Goldsmith P. V. 1997, "Capacity of fading channels with channel side information," *IEEE trans. Inform. Theory*, Vol. IT-43, pp. 187-190, November.
- 10) Glodsmith A. and Chua S. 1997, "Variable rate variable power M-QAM for fading channels", *IEEE trans. On communication*, October, Vol. COM_45, pp.1218-1230.
- 11) Jakes W. 1974, *Microwave mobile communications*, John Wiley & sons.
- 12) Alouini M. and Goldsmith A. 2000, "Adaptive Modulation over Nakagami Fading channels", *Kluwer Journal on Wireless Communications*, May, Vol. 13, No. 1-2, pp.119-143.

Professor Marwan Al-Akaidi has joined the department of Electronic Engineering at De Montfort University since 1991 as senior lecturer. He is promoted to principal lecturer in 1997, in November 2000 he became a professor of Communication & signal processing. His main research interest is in the field of Digital Signal Processing & Digital Communications including Speech Coding, Processing, Recognition, Wireless and Multimedia Mobile communication. His high quality of research own him a contract with Nokia – Finland to work on project on "Speech for Mobil & Wireless Communication" and another project with Panasonic to work on "Speech Recognition for New Generation TV & Internet", recent contract with ATDI to work towards Radio Network Planning.

He is the Editor & Chairman of over 20 national and international conferences including 3G Mobile Communication Technologies Joint Modular Languages Conferences. He is the guest editor for Simulation journal for it's special issue of Telecommunication in digital signal/image processing simulation journal. He was involve in the last 5 years in setting SCS chapter in various countries. In the last 4 years Chaired the Modeling & Simulation sponsored by SCS (Society for Computer Simulation).

He has achieved his Chartered Engineering status in 1990, and a fellowship of the Institute of Analyst and Programming in 1991. He is a senior member of the Institute of Electrical and Electronic Engineers and Fellow of the Institute of Electrical Engineering. He is a member of the European Council for SCS.

In September 1999, Professor Marwan Al-Akaidi appointed as a Chairman for the IEEE UKRI Signal Processing Society and March 2000 the IEEE UKRI Conferences chair and in December 2000 appointed to join the board of IEEE Industrial relation. He has won the award of the IEEE UKRI in Recognition of Outstanding Leadership as a chapter chair for Year 2001 & 2002.

In January 2002 he is appointed as a Head of the School of Engineering & Technology which have 3 divisions (Engineering, Technology & Design) at De Montfort University.

PERFORMANCE OF A CROSSBAR NETWORK USING MARKOV CHAINS

D. BENZAOUZ¹ A. FARAH²

1- *Laboratoire LMSS, FSI, Université de Boumerdes, 35000, Algeria*

2- *AUST, Faculty of Computer Science and Engineering, UAE*

Phone/fax (+213) 24 81 62 65 e-mail: dbenazzouz@umbb.dz

Abstract: A performance evaluation of a crossbar network is presented using discrete-time Markov chain (MC). Identical processors, totally synchronized with system clock and communicating through common memory modules. Simple MC models the behavior of each processor. We have developed state and output equations for discrete-time state space model. The transition probabilities of transition matrices are computed. The memory contention situation of the multiprocessor systems is considered. We show that the network is reliable for less than 100 processors. For network larger than 100 processors, a considerable degradation performance is observed which is due to contention.

Keywords: Crossbar - Performance analysis - Markov chains – Multiprocessor systems.

1- INTRODUCTION

Generally in multiprocessing systems, the processors can communicate and cooperate at different levels in solving a given problem, i. e., by sending messages or by sharing memory. Parallel systems are said to be tightly coupled if there is many processor interactions via shared memory. The speed of the machine is restricted by the memory bandwidth and hence by the interconnection network (IN) topology, and an IN with a dynamic topology is required. ALICE [Harrison and Reeve, 1987], designed to execute functional languages in parallel [Fiel and Harrison, 1988], and the NYU ultracomputer [Gottlieb et al, 1983] are examples of tightly coupled multiprocessor systems.

The performance studies of tightly coupled multiprocessor systems have generated a great interest [Marsan and Gerla, 1982]. The principle characteristic of a multiprocessor system is the ability of each processor to share a single main memory. The partitioning of main memory into several independent memory modules (MM), that can be in operation simultaneously, is known as memory interleaving. A memory system consisting of M memory modules (M-way interleaving) can be used to control severe performance degradation of the memory system. The interference occurs when two or more processors simultaneously attempt to access the same MM.

The mathematical models used to evaluate this class of multiprocessor systems are based on discrete-time Markov chains. A limit on the use of Markovian models of complex computer systems comes from the fact that their direct construction is

practically very difficult. Various approaches were examined [Skinner and Asher, 1969; Bhandarkar 1975; Ran, 1979].

The number of states increases very rapidly with system size. The explosive growth is due to the detailed information that the state must record the exact status of the queues at each server in the system. MC approach has been very successful technique for modelling, analysis and design of various kinds of systems [Florin et al. 1991; Benazzouz and Farah, 1998].

In this correspondence we develop a discrete time MC model for crossbar multiprocessor systems. It is assumed that the processors share M-way interleaved memory and can access any MM through the network. The entire system is synchronized with a system clock whose time period is referred to as the system cycle. The operation of the system can be assumed as under. At the beginning of each system cycle the processors are permitted to make selections of a MM at random. In case more than one request is made to same MM, the information is supplied to one of the processors selected at random, and the remaining processors permitted to make a retry at the next cycle. The behavior of the processors is considered to be independent but statistically identical. The entire process is thus stochastic in nature, and permits us to use MC to represent the state transition behavior of the processors. Section II describes the multiprocessor system interconnects. A detailed description of the crossbar network is presented in this section. The proposed model and the mathematical approach were developed in section III and IV respectively. The paper concludes with section V.

2- MULTIPROCESSOR SYSTEM INTERCONNECTS

Parallel processing demands the use of efficient system interconnect for fast communication among multiprocessors and shared memory, I/O, and peripheral devices. Hierarchical buses, crossbar switches, multistage and single stage networks are often used for this purpose. Switched networks provide dynamic interconnections between the processors and MM. Many classes of switched networks may be found in literature particularly the single stage interconnection network (SSIN) and multistage interconnection network (MIN) [El-Reweni and Lewis, 1997]. The crossbar switch network is a SSIN, nonblocking permutation network.

2.1- Crossbar Network

Crossbar networks provide the highest bandwidth and interconnection capability. A crossbar network can be visualized as a single stage switch network. Each crosspoint switch can provide a dedicated connection path between a pair. The switch can be set on or off dynamically upon program demand. The crossbar switch network configuration is illustrated in Fig.1. To build a shared memory multiprocessor, one can use a crossbar network between the processors and MM (Fig.1). This is essentially a memory access network. The Cmp multiprocessor [Wulf and Bell, 1972] has implemented a 16x16 crossbar network which connects 16 PDP 11 processors to 16 MM, each of which has a capacity of 1 million words of memory cells. The 16 MM can be accessed by at most 16 processors simultaneously. A crossbar network is cost effective only for small multiprocessors with a few processors accessing a few MM. A single stage crossbar network is not expandable one it is built. All processors can send memory requests independently and asynchronously. This poses the problem of multiple requests destined for the same MM at the same time. In such cases, only one of the requests is serviced at a time.

3- PROPOSED MODEL

One set of characteristics of a system is the states of the system. If we know all possible states of the system, then the behaviour of the system is completely described by its states. A system may have finite or infinite number of states. Here, we are concerned with only finite state systems. Suppose $X(t)$ describes the state of the system and has n values. That is, at a given time, $X_1(t), X_2(t), \dots, X_n(t)$ are the possible states of the system. $X_i(t)$ could be demand access to the MM or a process

with the private memory (PM) of the processor (a PM is an interne memory within the processor). The system will move from one state to another with some random fashion. That is, there is a probability attached to this. Let us suppose that $p(t)$ represents the probability distribution over $X(t)$ (note: $X(t)$ and $p(t)$ are vectors of size $n \times 1$) i.e. $p_1(t)$ is the probability of finding the system in state $X_1(t)$. In general, the predictive distribution for $X(t)$ is quite complicated with $p(t)$, being a function of all previous state variables $X(t-1), X(t-2)$ and so on. However, if $p(t)$ depends only upon the **preceding** state then the process is called Markov process.

A Markov process is a mathematical model that describes, in probabilistic terms, the dynamic behavior of certain type of system over time. The change of state occurs only at the end of the time period and nothing happens during the time period chosen. Thus, a Markov process is a stochastic process which has the property that the probability of a transition from a given state $p_i(t)$ to a future state $p_j(t+1)$ is dependent only on the present state and not on the manner in which the current state was reached.

The multiprocessor arrangement under consideration is shown in figure1. It consists of P processors and M memory modules interconnected through crossbar. Besides an M -way interleaved shared main memory, each processor has its own PM.

Contention problem arises when a message is attempted to be written in (or read from) a common MM by more than one processors. In crossbar multiprocessor systems (Fig.1) two types of possible interference can occur;

- When more than one processor attempts to access an idle MM at the same time.
- When a processor attempts to access a busy MM, (the processor is executing in its PM).

Due to this interference, a subset of processors might be blocked, thus giving degradation in the performance. The state space $X(t)$ of a processor in crossbar system can be a P valued random variable, taking only 3 values as a column vector;

$$X(t) = [X_1(t), X_2(t), X_3(t)] \quad (1)$$

$X_1(t)$ is the probability that the processor is in active state. It means that the processor is busy with its own private memory.

$X_2(t)$ is the probability that the processor is in accessing state.

$X_3(t)$ is the probability that the processor is queued at the required MM, due to nonavailability of MM.

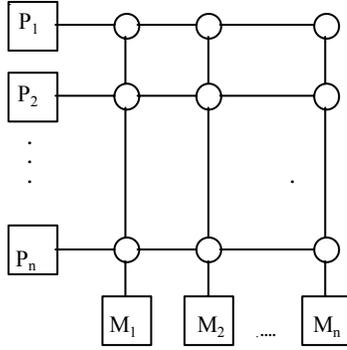


Fig. 1 Interprocessor-memory crossbar network built in C.mmp multiprocessor at Carnegie Mellon University (1972).

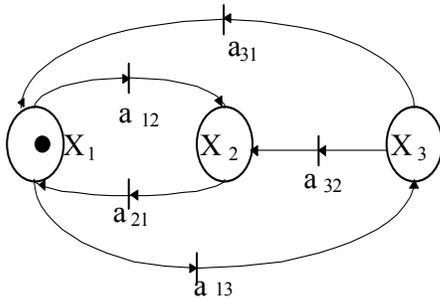


Fig. 2. Markov chain for a single processor.

For the sake of simplicity MC for the single processor is shown in Fig. 2 and can be coupled P times for P processors. The state transition behavior of each processor can be understood as follows.

The processor stays in active state for time duration equal to its processing time. It will enter the state $X_2(t)$ (accessing state) only if MM is idle and no other processor is a candidate for that module. The processor stays in this state for a time equal to the memory access time (processor-memory connection time is assumed to be zero). From accessing state it can go back to the active state after completing the memory access. Processor enter the queued state if MM has more than one simultaneous request from other processors and one of those requests is successful in getting access to the MM. In this state the processor has to wait in the queue until memory is available and then it goes back to the active state.

4- MATHEMATICAL APPROACH

A vector Markov process can be represented by the help of the discrete time state equation [Stark and Woods, 1986] provided that the input $r(t)$ is independent of previous state $X(t-1)$. The state and output equations are;

$$X(t+1) = AX(t) + Br(t) \quad (2)$$

where $X(t)$ and $X(t+1)$ are the probabilities that the system lies in specified state of the processor during current system cycle and the next cycle respectively.

The input probability vector $r(t)$ can be represented as a B-valued random variable and can be considered as;

$$r(t) = [r_1(t), r_2(t)] \quad (3)$$

where the probability $r_1(t)$ represents that the requesting MM is not available and $r_2(t)$ is the probability that the requesting MM is available.

The probability of the successful request (requests for which MM is idle) can be considered as an output $C(t)$ (which can be defined as P-valued random variable for the P processors) of the system.

$$C(t) = DX(t) + Er(t) \quad (4)$$

The matrices A , B , D , and E are of appropriate dimensions, and their components are the transition probabilities a_{ij} , b_{ij} , d_{ij} , and e_{ij} respectively. These components are defined as follows;

a_{ij} : the transition probability that the processor currently in state i goes to state j in the next system cycle.

b_{ij} : the transition probability that MM are available for transition to next state.

d_{ij} : the transition probability that the processor currently in any one of the states, active, accessing, queued, is successful to access requested MM in the next system cycle.

e_{ij} : the transition probability that the successful request i in the next system cycle is effected by the input j .

In order to calculate the transition probabilities following terms are introduced. The probability that a processor makes a request to access a particular MM at the beginning of a bus cycle is denoted by R . This is the probability of leaving states, active, and queued to access a particular MM. Therefore R is given by;

$$R = \frac{1}{M} (g_1 + g_3) \quad (5)$$

where g_i is the probability of leaving state i at the beginning of a system cycle. As the states of the processors are represented by irreducible ergodic MC, the g_i can be defined as the rate of leaving

state i . Thus g_i can be written as $g_i = \frac{K_i}{T_i}$

where T_i is the average time in any one of the states which is at least one system cycle and K_i is the limiting probability of being in state i . The probability that the processor finds a MM busy at the beginning of a system cycle and is also not

available in the next system cycle is denoted by BM and calculated as follows;

$$BM = \frac{P-1}{M}(K_2 - g_2) \quad (6)$$

That means one of the (P-1) processors is accessing that MM and is not going to release it in the next system cycle. Where K_2 is the probability that the processor is accessing MM and $(K_2 - g_2)$ is the probability that the processor is accessing and not going to release that MM in the next system cycle.

The term α is the probability that the memory request initiated by a processor finds the MM idle. The probability that a processor will not request a particular MM is (1-R), the probability that none of the P processors requests that MM is $(1-R)^P$, and therefore the probability that a particular MM is requested by at least one of the processors is $S=1-(1-R)^P$. The number of processors which request that MM at the beginning of the system cycle is PR. Then the value of α is given by;

$$\alpha = \frac{S}{PR} \quad (7)$$

The transition probability matrix of the model is denoted as T and given as follows,

$$T = \begin{bmatrix} -2(1+BM+\alpha) & (1-BM)-\alpha & (1-\alpha)-BM \\ 1 & -1 & 0 \\ \alpha & 1-BM & (BM-1)-\alpha \end{bmatrix}$$

Using Eqs. 5, 6, and 7, Eqs. 2 and 3 can be written as follows;

$$\begin{bmatrix} X_1(t+1) \\ X_2(t+1) \\ X_3(t+1) \end{bmatrix} = T \begin{bmatrix} X_1(t) \\ X_2(t) \\ X_3(t) \end{bmatrix} + \begin{bmatrix} (1-\alpha) & 0 \\ 0 & \alpha \\ (1-\alpha) & 0 \end{bmatrix} \begin{bmatrix} r_1(t) \\ r_2(t) \end{bmatrix} \quad (8)$$

$$C(t) = \begin{bmatrix} (1-BM)\alpha & 1 & (1-BM) \end{bmatrix} \begin{bmatrix} X_1(t) \\ X_2(t) \\ X_3(t) \end{bmatrix} + \begin{bmatrix} 1-\alpha & \alpha \end{bmatrix} \begin{bmatrix} r_1(t) \\ r_2(t) \end{bmatrix}$$

To solve these equations let's suppose a discrete time as said earlier, which means that we have a stationary homogeneous MC, where the probabilities $X_1(t)$, $X_2(t)$, and $X_3(t)$ are independent of time. Therefore, we can say that;

$$\begin{bmatrix} X_1(t+1) \\ X_2(t+1) \\ X_3(t+1) \end{bmatrix} = \begin{bmatrix} X_1(t) \\ X_2(t) \\ X_3(t) \end{bmatrix} \quad (9)$$

We can consider the fundamental relation to this linear system that consists of the sum of the state probabilities which is equal to 1. This relation is written as follows;

$$X_1(t) + X_2(t) + X_3(t) = 1 \quad (10)$$

Therefore, we can rewrite Eq.(8) by considering Eq.(9) and replacing one row by Eq.(10), thus we obtain the following expression,

$$\begin{bmatrix} 1 & 1 & 1 \\ -1 & 2 & 0 \\ -\alpha & BM-1 & 2-BM+\alpha \end{bmatrix} \begin{bmatrix} X_1(t) \\ X_2(t) \\ X_3(t) \end{bmatrix} = \begin{bmatrix} 1 \\ \alpha r_2(t) \\ (1-\alpha)r_1(t) \end{bmatrix} \quad (11)$$

We have developed an algorithm based on Gauss method to solve these equations. We calculate the probability states for one processor, then for P processors and try to see if these probabilities change as function of the number of processor. From fig. 3, we can conclude that these probability states $X_1(t)$, $X_2(t)$ and $X_3(t)$ stay almost constant. This is a very important result to proof the validity of the method since we can generalize the approach to P processors.

Two parameters were taken to evaluate the performance of a crossbar network. These parameters are the bandwidth (BW) and the probability of connection C(t).

The crossbar model will be analyzed under the same assumptions given by Agrawal [Bhuyan and Agrawal, 1983] which are:

- 1- The operation is synchronous; i.e., the messages begin and end simultaneously.
- 2- Each processor generates a random and independent request. The requests are uniformly distributed over all the memory modules.
- 3- At the beginning of a cycle, each processor generates a new request with a probability Pm. Thus is the average number of requests generated per cycle by each processor.
- 4- The requests which are not accepted are ignored. The requests issued at a cycle are independent of the requests issued in the previous cycle.

When the requests are random, it is possible for two or more processors to address the same memory module. Assumptions 1-4 are there to simplify the analysis.

The bandwidth (BW) and the probability of acceptance (Pa) of MxP crossbar network are presented by Agrawal and adapted to the model. BW is defined as the expected number of memory requests accepted per cycle. The BW and the Pa of the proposed model are given as follows;

$$BW = M - M(1 - C(t) / M)^P$$

where M is the number of MM and P is the number of processors.

Pa defines the probability that a request will be accepted;

$$Pa = BW / (C(t).P)$$

P_a is defined as the ratio of the expected BW to the expected number of requests generated by cycle.

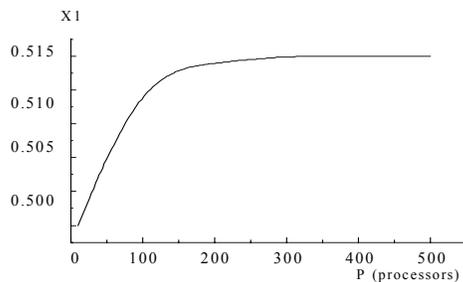
5. RESULTS

In this section, we present performance figures for a crossbar network in presence of contention. The simulation results are obtained from the proposed model where the number of processor is 500. Figures 3 shows the variation of the probability states of X_1 , X_2 and X_3 . For processors $P \leq 100$, we obtain an increase sharp of the state probabilities of X_1 , whereby a decrease sharp of the state probabilities of X_2 and X_3 is observed. We can say that, for low processors, most of the state probabilities show that the processors are in the active states. They are busy with there private memories. Almost, with the same state probabilities, the processors are either in the accessing or in the queuing states. For processors $P > 100$, the three state probabilities stay constant, $X_1=0.513$, $X_2=0.258$ and $X_3=0.229$. It is clearly shown in Fig.3d. We believe that there exists a degradation in performance and the system saturate in this range of processors. In the range of memories $M \leq 100$, a significant decrease of the probability of connection $C(t)$ and probability of acceptance P_a are shown in Fig.5 and 6 respectively. For $M > 100$, the memories are continuously busy and the $C(t)$ and P_a are small and stay almost constant. In this range of memories, the processors cannot access easily the memories. This causes a degradation performance which is due to the network size.

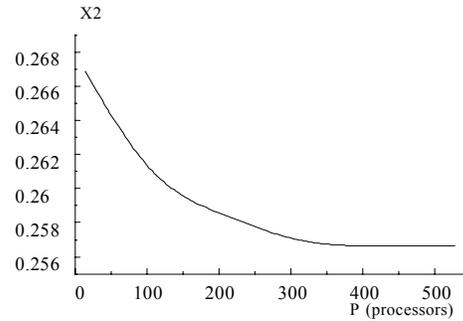
6. CONCLUSION

We have presented analytic model for blocking probability of crossbar network. The model is based on discrete-time MC under the assumption of random memory requests.

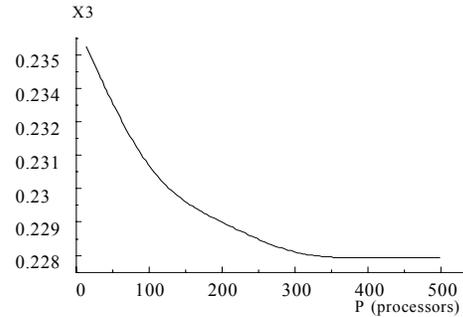
We believe that, the proposed analytic model of the crossbar can be used to adapt analytic models for the blocking probability of any arbitrary multistage interconnection network. The concepts developed here can later on be used to study the behavior of complex multiprocessor systems to resolve the memory contention problems under other considerations.



a) State probabilities of X_1

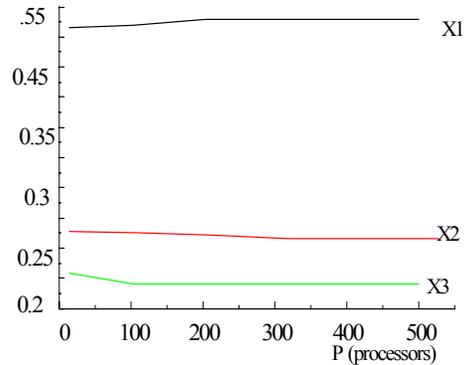


b) State probabilities of X_2



c) State probabilities of X_3

- State probabilities of X_1 , X_2 and X_3



d) State probabilities of X_1 , X_2 , X_3

Fig 3. Variation of the probabilities states with ($K_1=K_2=0.3, K_3=0.4, R_p=D=1, r_1=0.2, r_2=0.6$)

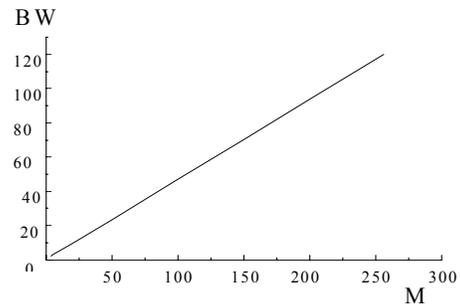


Fig. 4 Bandwidth of MxP networks

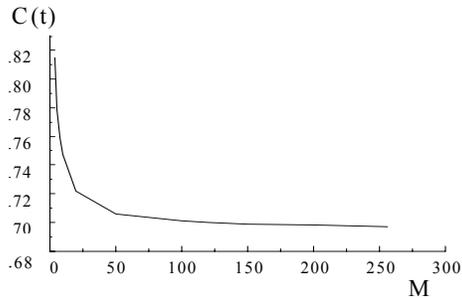


Fig. 5 Probability of connection of MxP networks

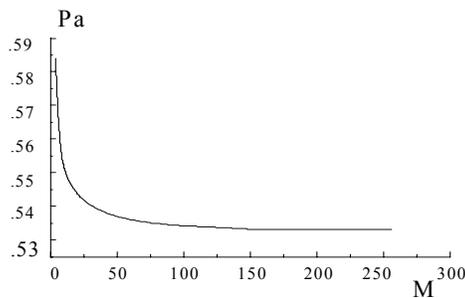


Fig. 6 Probability of acceptance of MxP networks

REFERENCES

- Harrison, P. G., and Reeve, M. J. 1987, "The parallel graph reduction machine, ALICE". *Proc. workshop on graph reduction*, Santa Fe, LNCS, N°279. Springer-verlag New York/Berlin.
- Fiel, A. J., and Harrison, P. G. 1988, "Functional programming", Addison-Wesley, New York.
- Gottlieb, A. et al. 1983, "The NYU ultracomputer designing an MIMD shared memory parallel computer", *IEEE Trans. Comput.* C32, 2, pp173-189.
- Marsan M.A. and Gerla M. 1982, "Markov models for multiple bus multiprocessor systems", *IEEE Trans. Comput.* Vol. C.31, pp239-248.
- Skinner C. E. and Asher J. R.. 1969, "Effects of storage contention on system performance", *IBM system J.* Vol. 8, pp319-333.
- Bhandarkar D. P. 1975, "Analysis of memory interference in multiprocessors", *IEEE Trans. Comput.*, Vol. C-34, pp897-908.
- Ran B. R. 1979, "Interleaved memory bandwidth in a model of a multiprocessor computer system", *IEEE Trans. Comput.*, Vol. C-28, N°9.
- Florin G., Fraize C. and Natkin S. 1991, "Stochastic Petri Net: Properties, Applications and

Tools", *Microelectron. reliability*, vol. 31, N°4, pp669-697.

Benazzouz D. and Farah A. 1998, "The use of Petri nets in the performance evaluation of shuffle-exchange network under uniform traffic distribution", *the Arabian Journal for Science and Engineering, AJSE*, vol. 23, n° 2B, pp. 253-263.

Benazzouz D. and Farah A. 1998, "Performance evaluation of baseline MIN", in *Proc. IMACS-IEEE on Computational Engineering in Systems Application*, Nabeul-Hammamet, Tunisia, pp.350-356.

Stark H., and Woods J.W. 1986, "probability random processes and estimation theory for engineers", New Jersey: *Prentice-Hall, Inc.*

Wulf W.A. and Bell C. G 1972, "C.mmp-A multi-miniprocessors", *Proc. Fall joint Compt. Conf.*, pp765-777.

El-Rewini H. & Lewis T. G. 1997, "Distributed and parallel computing", Ed. Manning, pp. 102-103.

Bhuyan L.N. and Agrawal P. 1983, "Design and performance of generalized interconnection networks", *Trans. on computers* vol. C-32, no12, pp.1081-1090.

BIOGRAPHY:



Dr. Djamel BENAZZOUZ graduated in 1982 from the National Institute of Electricity and Electronics (INELEC) Boumerdes, Algeria. He joined industry as maintenance engineer in the two major Algerian Companies: Sonatrach (Petroleum Industry) and Sonelgaz (Electric Utility Company). He returned to research and Education since 1986 at the National Institute of Mechanical Engineering which became in 1998 University of Boumerdes. He received his Magister degree in applied electronics in 1991 at INELEC and his Doctorate d'Etat in 1999 at Ecole Nationale Polytechnique, Algiers. He worked as associate Professor in 1998 at the university of Boumerdes. He has been heavily involved in the field of microprocessor-based systems. His research interests include architecture digital system design, verification and test of digital circuits, hardware and software and the identification systems using neural network and fuzzy logic.

MINSimulate – A MULTISTAGE INTERCONNECTION NETWORK SIMULATOR

DIETMAR TUTSCH and MARCUS BRENNER

*Technische Universität Berlin
Real-Time Systems and Robotics
D-10587 Berlin, Germany
{dietmart,mbrenner}@cs.tu-berlin.de*

Abstract: Multistage interconnection networks are frequently proposed as connections in multiprocessor systems or network switches. In this paper, a new tool for stochastic simulation of such networks is presented. Simple crossbars can be simulated as well as multistage interconnection networks that are arranged in multiple layers.

Keywords: network simulator, multicasting, multistage interconnection networks, crossbars, performance

1 Introduction

Multistage interconnection networks (MINs) with the banyan property are proposed to connect a large number of processors to establish a multiprocessor system [1]. They are also used as interconnection networks in Gigabit Ethernet [2] and ATM switches [3]. Such systems require high performance of the network. MINs were first introduced for circuit switching networks. To increase the performance of a MIN, buffered MINs were established as packet switching networks. For instance, Dias and Jump [4] inserted a buffer at each input of the switching elements (SE). Patel [5] defined delta networks. Delta networks are a subset of banyan networks (MINs with just one path between a given input and output). It is additionally required that packets can use the same routing tag to reach a certain network output independently of the input at which they enter the network.

Many variations of delta networks were introduced. Most of them result in MINs that lose the unique path property (and therefore the delta property) in order to reduce blocking. Clos [6] presented a MIN consisting of three stages and non-quadratic SEs. Turnaround MINs [7] are established by bidirectional links between the SEs. Network inputs and SE inputs operate also as outputs. Dilated banyan networks [8] arise by multiplying the links between the SEs: the link bandwidth is enhanced. Replicated banyan networks [8] originate from multiplying the whole banyan network. Multilayer multistage interconnection networks (MLMINs) are introduced to apply especially to multicast traffic. Those networks are established similarly to replicated MINs but a new replication starts at every network stage [9]. The network results in a growing number of layers from sources to destinations. Many other kinds of MINs are known. A detailed description can be found for instance in [3].

In this paper, a simulation tool for performance evaluation of MINs is presented. The tool is designed to investigate MINs with the delta property or with multiple layers. Various design parameters can be examined concerning network performance in terms of throughput and

delay. Network traffic and resource scheduling is modeled by stochastic simulation.

The paper is organized as follows. The architecture of MINs is described in Section 2. Section 3 applies to the simulator features and shows how the simulator operates. Some examples of results are given in Section 4. Section 5 summarizes and gives conclusions.

2 Architecture of MIN

Various architectures of multistage interconnection networks exist. This section presents those architectures that can be modeled by the new simulator (called *MINSimulate*).

2.1 MIN with Banyan Property

Multistage interconnection networks with the banyan property are networks where a unique path from an input to an output exists. Such MINs of size $N \times N$ consist of $c \times c$ switching elements with $n = \log_c N$ stages. An 8×8 MIN consisting of 2×2 SEs is represented by Figure 1.

To achieve synchronously operating switches, the network is internally clocked. In each stage k ($0 \leq k \leq n-1$) of non-shared buffer networks, there is a FIFO buffer of size $m_{max}(k)$ in front of each switch input. The packets are routed by store and forward routing or cut-through switching from a stage to its succeeding one by backpressure mechanism.

Networks consisting of shared buffers are established by replacing the c FIFO input buffers of size $m_{max}(k)$ of a $c \times c$ switch with one common buffer of size $c \cdot m_{max}(k)$ [10]. This shared buffer is organized as follows: Each switch input reserves sufficient buffer space to store at least one packet in order to avoid the isolation of inputs (see below). The remaining buffer space of $c \cdot m_{max}(k) - c$ packets is available to all inputs. Each input forms a FIFO input queue of packets. If an input receives a new packet from the previous stage that has to be stored, the input al-

locates buffer space of the commonly used buffer part if available. If there is no further buffer available the packet is blocked at the previous stage.

An input with a queue of more than one packet deallocates buffer space if it sends a packet to the next network stage. This space is returned to the pool of the commonly available buffer space.

Guaranteeing at least one buffer space to each input avoids that an input without any buffer cannot participate in the switch routing process because it is not able to receive a packet that has to be forwarded. For instance, let us assume that one of the inputs (hot spot input) receives much more packets than the other ones. This input would allocate up to all of the buffers. Packets of the previous stage that are directed to the other inputs would be blocked at the previous stage even if their final destination is different from the first packet queued at the hot spot input. Only the hot spot input would contribute to the switch traffic and all other inputs would remain idle.

Multicasting is performed by copying the packets within the $c \times c$ switches. In ATM context, this scheme is called cell replication while routing (CRWR). Figure 1 shows such a scenario for an 8×8 MIN consisting of 2×2 SEs. A packet is received by Input 3 and destined to Out-

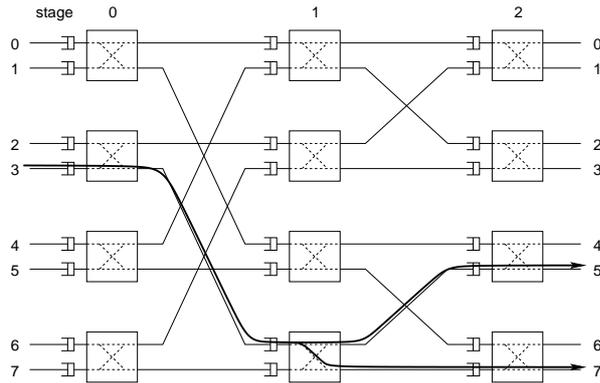


Figure 1: Multicast while routing

put 5 and Output 7. The packet enters the network and is not copied until it reaches the middle stage. Then, two copies of the packet proceed on their way through the remaining stages.

Packet replication before routing in the above example would copy the packet and send it twice into the network. Therefore, packet replication while routing reduces the amount of packets in the first stages.

Comparing the packet density in the stages in case of replication while routing shows that the greater the stage number, the higher is the amount of packets. In other words: there are much more packets in the last stages due to replication than in the first stages. The only exception is if the traffic pattern results in such a destination distribution that packet replication has to take place at the first stage. Then, the amount of packets is equal in all stages. But such a distribution is very unlikely, in general.

To set up multistage interconnection networks that are

appropriate for multicasting, the previously mentioned different traffic densities of the stages must be considered. MLMINs, which are described later in this section, belong to this kind of networks. Their roots are in replicated MINs.

2.2 Replicated MIN

Replicated MINs enlarge regular multistage interconnection networks by replicating them L times. The resulting MINs are arranged in L layers. Corresponding input ports are connected as well as corresponding output ports. Figure 2 shows the architecture of an 8×8 replicated MIN consisting of two layers in a three-dimensional view. Such a concept was introduced by Kruskal and Snir [8]. Packets are received by the inputs of the network and distributed to the layers. Layers may be chosen at random, by round

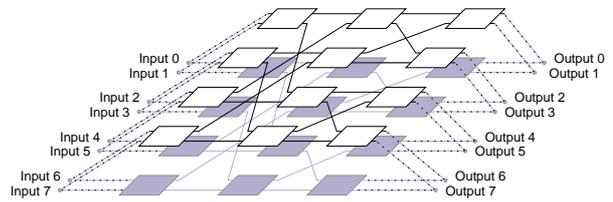


Figure 2: Replicated multistage interconnection network ($L = 2$, 3D view)

robin, dependent on layer loads, or any other scheduling algorithm. The distribution is performed by a $1:L$ demultiplexer.

At each network output, an $L:1$ multiplexer collects the packets from the corresponding layer outputs and forwards them to the network output. Two different output schemes are distinguished: single acceptance (SA) and multiple acceptance (MA). Single acceptance means that just one packet is accepted by the network output per clock cycle. If there are packets in more than one corresponding layer output, one of them is chosen. All others are blocked at the last stage of their layer. The multiplexer decides according to its scheduling algorithm which packet to choose.

Multiple acceptance means that more than one packet may be accepted by the network output per clock cycle. Either all packets are accepted or just an upper limit R . If an upper limit is given, R packets are chosen to be forwarded to the network output and all others are blocked at the last stage of their layer. As a result, single acceptance is a special case of multiple acceptance with $R = 1$.

In contrast to regular multistage interconnection networks, replicated MINs may cause out of order packet sequences. Sending packets belonging to the same connection to the same layer avoids destruction of packet order.

2.3 Multilayer MIN

Multilayer multistage interconnection networks (MLMINs) consider the multicast traffic character-

istics. As mentioned above, the amount of packets increases from stage to stage due to packet replication. Thus, more switching power is needed in the last stages compared to the first stages of a network.

To supply the network with the required switching power, MLMIN structure increases the number of layers in each stage. The factor with which the number of layers is increased is called growth factor G_F ($G_F \in \mathbb{N} \setminus \{0\}$). Figure 3 shows an 8×8 MLMIN (3 stages) with growth factor $G_F = 2$ in lateral view. That means the number of

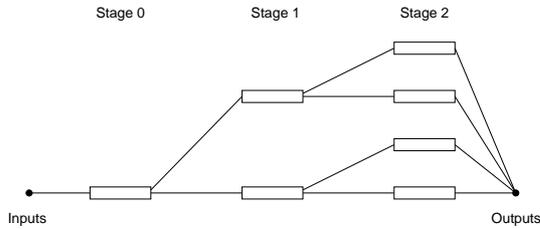


Figure 3: Multilayer multistage interconnection network ($G_F = 2$)

layers is doubled each stage and each switching element has twice as much outputs as inputs. Consider for instance that 2×2 SEs are used. Such an architecture ensures that even in case of two broadcast packets at the inputs all packets can be sent to the outputs (if there is buffer space available at the succeeding stage). On the other hand, unnecessary layer replications in the first stages are avoided.

Choosing $G_F = c$ ensures that no internal blocking occurs in an SE, even if all SE inputs broadcast their packets to all SE outputs. Nevertheless, blocking may still occur at the network output depending on R .

A drawback of MLMIN architecture arises from the exponentially growing number of layers for each further stage. The more network inputs are established, the more stages and the more layers result. To limit the number of layers and therefore the amount of hardware, two options are available: starting the replication in a more rear stage and/or stopping further layer replication if a given number of layers is reached.

The first option is demonstrated in Figure 4 in lateral view. The example presents an 8×8 MLMIN in which replication does not start before Stage 2 (last stage) with $G_F = 2$. A 3D view is given in Figure 5. The stage num-

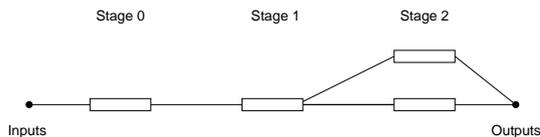


Figure 4: MLMIN in which replication starts at Stage 2 (lateral view)

ber in which replication starts is defined by G_S ($G_S \in \mathbb{N}$). Figures 4 and 5 introduce a MLMIN with $G_S = 2$. Of course, moving the start of layer replications some stages

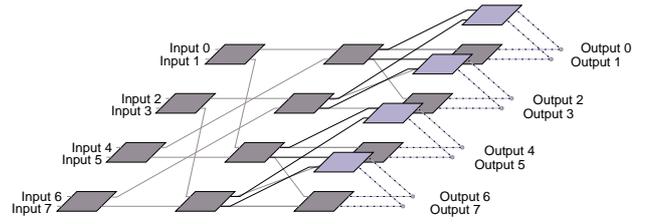


Figure 5: MLMIN in which replication starts at Stage 2 (3D view)

to the rear not just reduces the number of layers. It also reduces the network performance due to less SEs and therefore less paths through the network.

Stopping further layer replication if a given number G_L of layers is reached also reduces the network complexity ($G_L \in \mathbb{N} \setminus \{0\}$). It prevents exponential growth beyond reasonable limits in case of large networks. Figure 6 shows such an MLMIN with limited number of layers in lateral view. 3D view is presented in Figure 7. The number of

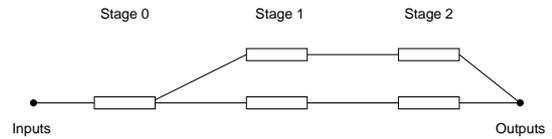


Figure 6: MLMIN with limited number of layers (lateral view)

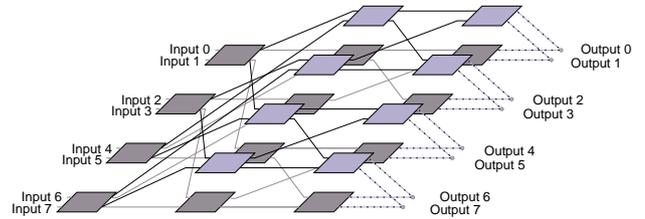


Figure 7: MLMIN with limited number of layers (3D view)

layers of this 8×8 MLMIN is limited to an upper number of $G_L = 2$. Layers are replicated with a growth factor of $G_F = 2$. As in the previous option, the reduced amount of SEs decreases network performance as well.

Both presented options can be combined to reduce network complexity further. Such a network is determined by parameters G_S (start of replication), G_F (growth factor), and G_L (layer limit). For instance, Figure 7 shows an MLMIN with $G_S = 1$, $G_F = 2$, and $G_L = 2$.

Regular MINs and replicated MINs can be considered as special cases of MLMINs. Regular MINs are equivalent to MLMINs with $G_F = 1$. In this case, G_S and G_L have no effect. Replicated MINs are equivalent to MLMINs with $G_S = 0$, $G_F = L$, and $G_L = L$.

3 Simulator *MINSimulate*

The new simulator presented in this paper is called *MIN-Simulate*. It is designed to model MINs with the banyan property, replicated MINs, and MLMINs, as well as simple crossbar switches.

3.1 Features

Stochastic simulation is performed by C++ code. According to the network parameters given by the user, the network is first established. It is represented as a directed graph starting at the sources (network inputs) and ending at the destinations (network outputs). The simulator is packet based. Packets are generated at the sources. Each packet is provided with a tag determining its destination. Due to multicasting this tag is modeled by a vector of N binary elements, each representing a network output. The elements of the desired outputs are set to “true”. If the packet arrives at a $c \times c$ switch, the tag is divided into c subtags of equal size. Each subtag belongs to one switch output, the first (lower indices) subtag to the first output, etc. If a subtag contains at least one “true” value a copy of the packet is sent to the corresponding output containing the subtag as the new tag.

To keep the amount of allocated memory as small as possible, just a representation of the packets, referred to as containers, is routed along the network paths. These containers are replaced by the actual packets at the network outputs allowing evaluations. Figure 8 gives a short sketch of the simulation model. So called Contain-

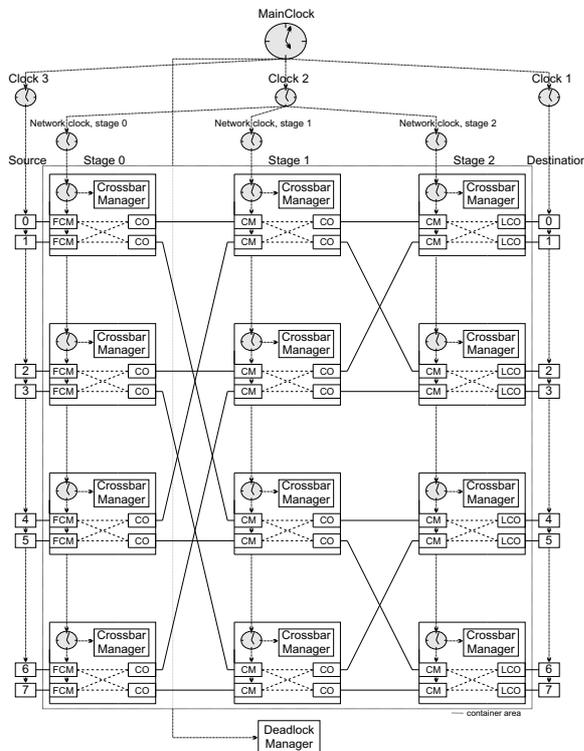


Figure 8: Sketch of the simulation model

erMultiplexers (CM) receive the containers and store them in the queues. At the first network stage, First-Container-Multiplexers (FCM) additionally perform the replacement of the packets by containers. So called Container-Outputs (CO) send the containers to the next network stage. At the last stage, Last-Container-Outputs (LCO) additionally replace the containers by the corresponding packets. Each operation of a switch is controlled by its Crossbar Manager. The clocks perform the sequencing of the parallel actions due to computer simulation.

Confidence level and relative error of simulation results is observed by the toolkit *Akaroa*. The simulation is stopped when those termination criteria are met. *Akaroa* is developed at the University of Canterbury, New Zealand [11].

3.2 Graphical User Interface

The network to be evaluated is determined by the user via a graphical user interface (GUI). Figure 9 shows the main tab to settle simulation parameters. A short sketch

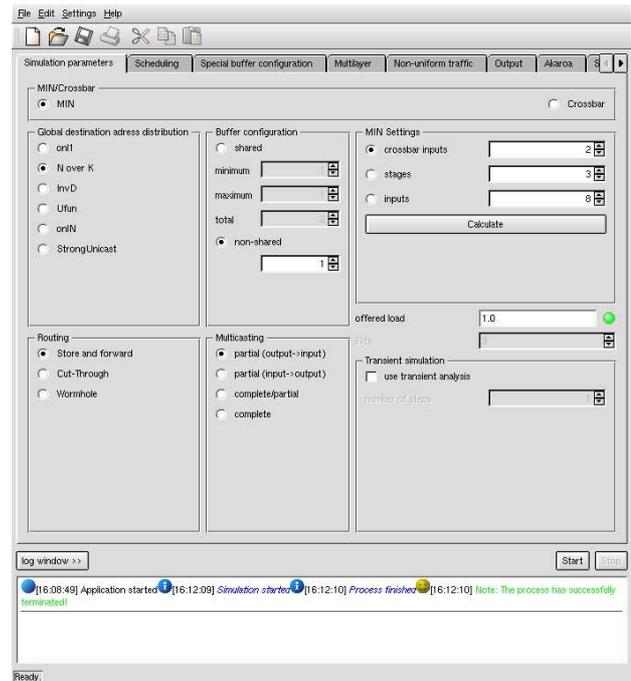


Figure 9: Main tab of *MINSimulate*

of the parameters and their available settings is given below. They are described in more detail in [12].

First, it can be chosen whether to simulate single crossbars or MINs. If crossbars are chosen the number of inputs can be defined. If MINs are chosen two of the three parameters according to equation $n = \log_c N$ must be set by the user. The third one is determined by the GUI. Input buffers can be chosen as shared ones (with a minimum size and maximum size for each crossbar input, as well as the overall buffer size of a crossbar) or as

non-shared ones (with the size per crossbar input). Tab Special Buffer Configuration allows individual buffer settings for each stage.

The global address destination distribution of packets entering the network can also be varied. The most important patterns are `on11` (only unicast; all targets are with equal probability a packet's destination), `N over K` (multicast; all target combinations are with equal probability a packet's destination), `Ufun` (multicast with many unicasts and many broadcasts), and `on1N` (only broadcast). If single sources are desired to produce deviating address distributions, `tab Non-uniform Traffic` helps.

When choosing the routing algorithm the following packet switching schemes are available: store and forward routing, cut-through switching, or wormhole routing. Multicast in case of wormhole routing usually suffers from deadlocks. The wormhole routing algorithm of *MIN-Simulate* avoids deadlocks by grouping appropriate parts of the network [13]. Wormhole routing requires dividing the packets into flits. The number of flits per packet is also a parameter for the simulation.

The kind of multicast can be set to complete multicast, partial multicast, or a two phase version of both. A further parameter represents the offered load to each network input. The last parameter in the main tab determines whether to observe measures transiently instead of observing the steady state. In case of transient simulation, the number of clock cycles to simulate can be fixed.

Tab `Multilayer` allows to configure MLMINs as presented in Section 2.3. Choosing constant number of layers refers to replicated MINs.

Instead of simulating a particular network configuration, a parameter can be varied to deal with parameter dependent results. In `tab Simulation Series`, the parameter to vary is chosen. A start value, end value, and step size determines the variation. If desired, step size can be changed once in the parameter interval.

Performance measures are chosen via `tab Output`. Most important ones are throughput, delay times, and queue lengths. A histogram of delay times within an interval is also available. Deadlines can be added to packets and packets that exceed their deadline are then removed. In such scenario packet loss results as a measure.

Akaroa parameters to determine confidence level and relative error of results are set in `tab Akaroa`.

4 Example

To present an example of the results obtainable using *MINSimulate* the following evaluation will be performed: Given the task to design a multistage interconnection network of size $N = 64$, how will the network's performance be affected by the choice of the switching element size c ?

Simulations are run for MINs composed of 2×2 and 4×4 SEs. This example's simulations were performed using an accuracy of 0.02 and a confidence level of 98%.

The size of the buffer in front of each SE (non-shared

buffering) is set to $m_{\max}(k) = 2$ for all stages k . Routing is performed according to a store and forward scheme.

The MINs to be evaluated are being offered multicast traffic governed by multicast traffic pattern *N over K*, that is, all possible combinations of target ports have an equal probability of being a packet's destination. Packets are offered to the network at a constant (time-independent) rate.

Figure 10 shows the output throughput for both network configurations. There is little to no difference in

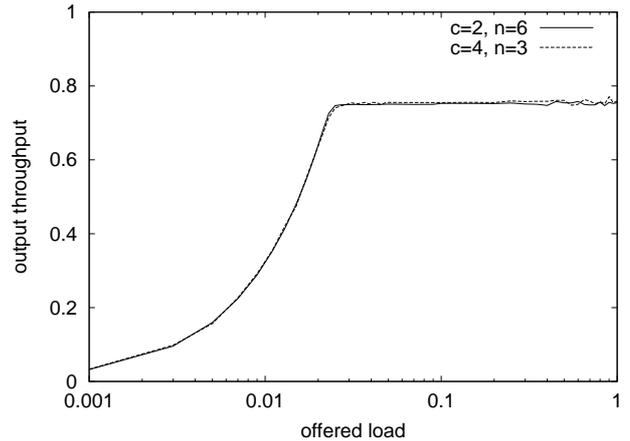


Figure 10: Throughput at the network's outputs

throughput performance between MINs based on 2×2 and 4×4 SEs. When the offered load rises above approx. 0.02 (average number of offered packets per input and clock cycle) the network begins to saturate: Due to the multiplication of packets inside the SEs, queues build up and by the aforementioned backpressure mechanism the actual input throughput is limited to about 0.02 as well. I. e., in saturation there are less packets accepted by the MIN than are being generated and ready to be sent.

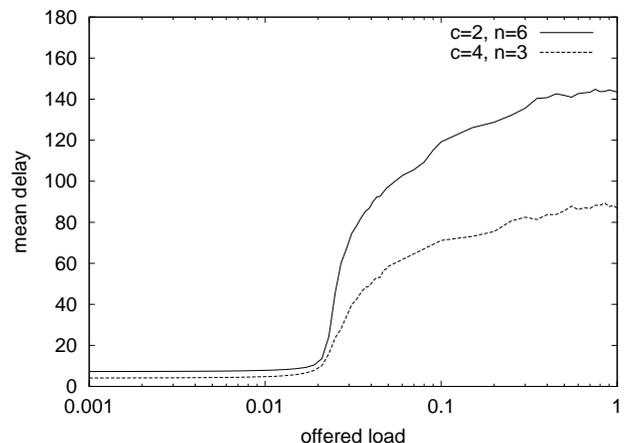


Figure 11: Mean packet delay times

When looking at the average time it takes a packet to reach its destination, differences between the two network configurations become apparent (Figure 11). In case of store and forward routing the minimum amount of time required to traverse the MIN is equal to the number of stages in the network (3 and 6, respectively). Of course, this requires that the packet is not being delayed even a single time. With rising offered load the number of conflicts between packets increases and buffers in front of the switching elements fill up. Thus, for an offered load greater than approx. 0.02 the mean delay time rises considerably. In case of an unlimited buffer space $m_{\max}(k)$ the delay times would grow beyond limit, similar to an unstable queuing system.

The most apparent difference between the two MIN configurations is the lower mean delay of the 4×4 -based one in saturation (offered load > 0.02): 80–90 opposed to 120–140 clock cycles for the 2×2 -configuration. Because of the lower number of stages there occur less conflicts between packets, resulting in an overall lower delay. In addition, the slope of the curve in the area of 0.02 to 0.07 is less steep for the MIN consisting of only 3 stages.

To conclude this example, if achieving low packet delay times was an issue in designing a MIN, one would opt for 4×4 switching elements (if the additional cost would be acceptable), although throughput would not benefit.

5 Conclusion

This paper presents the tool *MINSimulate* for stochastic simulation of multistage interconnection networks (MINs). Due to the tool's GUI support the wide variety of simulation parameters is easily accessible to the user.

MINSimulate is able to simulate simple (i. e. fully meshed) crossbars, multistage interconnection networks (MINs) with the delta property, and MINs that are arranged in multiple layers. Within each network architecture, simulations can be performed using a wide variety of input parameters such as offered load, multicast traffic pattern, routing scheme or (internal) buffer configuration. The simulations yield performance measures such as throughput, mean delay, delay time distributions or mean buffer queue lengths in individual network stages. During simulation, all data is evaluated according to pre-set levels of confidence and accuracy using the statistical library *Akaroa*[11].

Transient network behavior can also be evaluated with *MINSimulate*, this is useful for studies of the fine grain time-dependent performance, especially when observing traffic that changes with time.

The presented tool allows for evaluating various network configurations under different traffic conditions. Therefore, one can easily establish knowledge whether a particular network design suits a task at hand.

Currently, *MINSimulate* is extended in the way that also non-delta networks can be simulated. As a first step, Clos networks and turnaround MINs are incorporated.

6 Acknowledgements

We would like to thank Daniel Benecke, Matthias Hendler, Rainer Holl-Biniasz, and Arvid Walter for implementing and maintaining *MINSimulate*. Without their work and ideas, this project would never have been as successful.

References

- [1] Gheith A. Abandah and Edward S. Davidson. Modeling the communication performance of the IBM SP2. In *Proceedings of the 10th International Parallel Processing Symposium (IPPS'96); Hawaii*. IEEE Computer Society Press, 1996.
- [2] Toshio Soumiya, Koji Nakamichi, Satoshi Kakuma, Takashi Hatano, and Akira Hakata. The large capacity ATM backbone switch 'FETEX-150 ESP'. *Computer Networks*, 31(6):603–615, 1999.
- [3] Ra'ed Y. Awdeh and H. T. Mouftah. Survey of ATM switch architectures. *Computer Networks and ISDN Systems*, 27:1567–1613, 1995.
- [4] Daniel M. Dias and J. Robert Jump. Analysis and simulation of buffered delta networks. *IEEE Transactions on Computers*, C-30(4):273–282, April 1981.
- [5] Janak H. Patel. Performance of processor–memory interconnections for multiprocessors. *IEEE Transactions on Computers*, C-30(10):771–780, October 1981.
- [6] C. Clos. A study of nonblocking switching network. *Bell System Technology Journal*, 32:406–424, March 1953.
- [7] Hong Xu, Yadong Gui, and Lionel M. Ni. Optimal software multicast in wormhole-routed multistage networks. *IEEE Transactions on Parallel and Distributed Systems*, 8(6):597–606, June 1997.
- [8] Clyde P. Kruskal and Marc Snir. The performance of multistage interconnection networks for multiprocessors. *IEEE Transactions on Computers*, C-32(12):1091–1098, 1983.
- [9] Dietmar Tutsch and Günter Hommel. Multilayer multistage interconnection networks. In *Proceedings of 2003 Design, Analysis, and Simulation of Distributed Systems (DASD 2003); Orlando*, pages 155–162. SCS, April 2003.
- [10] Dietmar Tutsch, Matthias Hendler, and Günter Hommel. Multicast performance of multistage interconnection networks with shared buffering. In *Proceedings of the IEEE International Conference on Networking (ICN 2001); Colmar*, pages 478–487. IEEE, July 2001.
- [11] Krzysztof Pawlikowski, Victor W. C. Yau, and Don McNickle. Distributed stochastic discrete-event simulation in parallel time streams. In *Proceedings of the 1994 Winter Simulation Conference; Lake Buena Vista*, pages 723–730, December 1994.
- [12] <http://uscream.cs.tu-berlin.de/minsimulate/>.
- [13] V. Varavithya and P. Mohapatra. Asynchronous tree-based multicasting in wormhole-switched multistage interconnection networks. *IEEE Transactions on Parallel and Distributed Systems*, pages 1159–1178, November 1999.

ECONOMIC RELIABILITY FORECASTING IN AN UNCERTAIN WORLD

ED STOKER[†] and JOANNE BECHTA DUGAN[†]

[†] *University of Virginia*

Electrical and Computer Engineering Department

P.O. Box 400473 Charlottesville VA 22904-4136

estoker@cox.net and jbd@virginia.edu

Abstract Substantial work has applied stochastic techniques to network reliability models. These techniques can estimate *risk*, *variance* and *uncertainty* values. Unfortunately, these models do not address the issues of revenues, return on investment, or the time-value of money. To address these issues, we have developed an Economic Reliability Analysis [ERA] framework at the University of Virginia that fuses reliability engineering methods with economic analysis. We combine the ERA framework with stochastic techniques to evaluate a simple network and a proposed network upgrade. We simulate key availability and financial elements of both networks and apply the ERA framework to these elements. These results are compared with full path enumeration results of the same networks. This analysis provides a richer, more complete method to apply stochastic network techniques to operational network upgrades.

Index Terms Stochastic network models, Reliability and maintenance models, Stochastic simulation.

I. INTRODUCTION

Consider the following scenario. A company has a network that provides the basis for its revenue. The company must choose whether to maintain the status quo or modify the network to gain a new revenue stream. The company only has resources to choose one of these projects. The question is, *which project should be implemented*. The general problem, simply stated is: "How do you profitably operate, maintain and evolve a dependable operational network?" This general problem can be addressed by a set of smaller, more directed questions. These questions are:

- 1) What is the economic effect of developing and implementing a network change?
- 2) What is the economic improvement associated with improving network reliability?
- 3) When do the costs of improving network availability exceed its expected benefits?
- 4) How reliable must a new network be before it becomes operational?

These questions can be difficult to answer. Most organizations have several different types of network components in a network, each with different associated reliability and repair cost data. In addition, different *user-oriented* measurements for availability and their economic impact must be understood. As such, these two costs associated with a network failure (*network component repair cost* and, *lost revenue associated with a network failure*) must be considered when modeling the economic impact of network repairs.

This article aims to extend the network reliability model techniques with an Economic Reliability Analysis (ERA) framework developed at the University of Virginia [7] and apply it to an operational network system.

The motivation for writing this paper is threefold. First, we want to address the dilemma of picking a project that will affect the reliability and economics of an *operational* network system. We also want to extend the framework to provide some estimates of *risk* and *confidence* that can be provided from the inclusion of stochastic modeling techniques. Finally, we want to illustrate the power and usefulness that simulation techniques can provide to practical business and engineering decisions.

II. RELATED WORK

Several research groups have investigated the relationship between reliability and economic value. Current literature indicates this relationship has taken several directions.

Research at British Telecom, [8], [1], [9] focused on modeling repair costs of their own telecommunications network. This directed research aimed at predicting expected costs without providing any structural insight into controlling these costs. Their system was a very large, distributed network, where the principal issue was the rapid detection, identification and restoral of telephone service outages. Network design or new service offerings were not considered.

Economic models have been proposed to deal with the production and distribution of electrical power in which the reliability of the power grid, electrical production and distribution costs along with macroeconomic models are considered [10], [11], [12]. Yoon and Ilić treat electrical power as a commodity and propose a new business model for this industry. Their research aims to improve delivery of electrical power to consumers with greater economic efficiency.

Mitchell and Gelles [4] describe a framework for risk-value models. Research in Markov reward models [3], Petri net models [2], advanced Monte-Carlo simulation procedures [5],

and rare event simulation [6] provide insights into the use of stochastic techniques to estimate network reliability.

Current approaches do not adequately describe the monetary benefits (i.e. *revenues*) associated with *operational* networks or the *time-value of money*. Inclusion of these concepts can produce a better understanding of the economic worth of a reliable *operational* network.

In [7], Stoker and Dugan define an Economic Reliability Analysis methodology to evaluate the economic worth of a reliable network. The general strategy behind the *ERA* framework is to collect and use *availability* and *financial* data about a system from within an organization rather than build "yet another reliability/financial model." The *ERA* framework provides a means to determine how *changes* in component reliability, service pricing, and component/task dependencies influence system *availability*, *return on investments*, and *design*.

Step Function

1. Determine the duration, size and scope of the analysis.
2. Build network reliability models for all design choices.
3. Map network reliability information into **component** and **task** failure data.
4. Calculate revenue vectors for all design choices.
5. Calculate lost revenue vectors for all design choices.
6. Calculate recurring cost vectors for all design choices.
7. Calculate other cost vectors for all design choices.
8. Calculate capital cost vectors for all design choices.
9. Calculate [ERV] for all design choices.
10. Analyze and interpret results of evaluation.

TABLE I
ERA FRAMEWORK ALGORITHM

Table I provides a summary of the operational set of processes performed by the *ERA* framework. These processes will be illustrated in the following example.

III. STOCHASTIC RELIABILITY EXAMPLE

Economic and reliability processes will be simulated using *stochastic* methods, sampled and evaluated with the *ERA* framework. The results of this simulation will be compared to results using *deterministic* methods. See [7] for a complete description of the Economic Framework, network solutions and exact results. These simple simulations will allow us to easily compare the impact of *uncertainty* on expected system *reliability* and *economic worth*. In addition, these simulations will provide an expected range of parametric values for both networks. Finally, comparisons that account for normal variances between networks can be made.

Figure 1 shows a current network (**Network A**) and a proposed network (**Network B**). The proposed change will be to add a node and move two links (A_3 and A_6) to connect between Nodes s and t . **Network B** is more *reliable* than **Network A**,

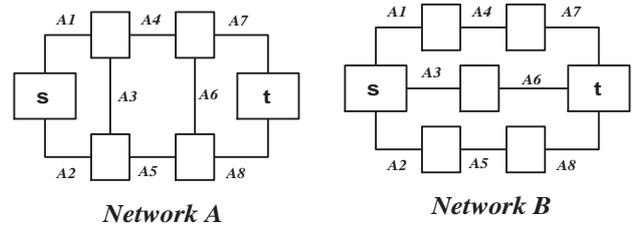


Fig. 1. Example Networks

assuming that only edges fail and that edge reliability metrics are identical in both Networks.

IV. QUESTIONS TO ANSWER

Stochastic network reliability models can provide *confidence intervals* on reliability parameters by simulating when components fail and repair rates. These models also allow one to examine the impact of *uncertainty* on network availability. Both of these elements can significantly alter network design choices. Below are a set of questions that can be answered using a stochastic network model.

- 1) What is the range of expected availability for a given network?
- 2) What is the range of expected *Economic Reliability Values* for a given network?
- 3) What is the impact of *uncertainty* on expected network availability?
- 4) What is the impact of *uncertainty* on network *Economic Reliability Value*?
- 5) How does component reliability *variances* affect network availability?
- 6) How does component reliability *variances* affect network *Economic Reliability Value*?

A. Model Assumptions

- 1) $A_{Net_A} = \$1500$ and $A_{Net_B} = \$1550$.
- 2) Network failure cost is \$100 per failure. Network failure duration is 2 hours.
- 3) Edge MTTR is 2 hours, edge availability is .99, average edge repair cost is \$10/repair and the average edge repair rate is \$0/hour for all edge repairs and applies to both networks.
- 4) $F_{Net_A} = \$0$ and $F_{Net_B} = \$10$.
- 5) $G_{Net_B} = \$1000$ for the initial time period else, $G_{Net_B} = \$0$. $G_{Net_A} = \$0$.
- 6) Discount rate process (*DR*) is 1% per month for the duration of the analysis.
- 7) *Investment period is 24 months*. This is used to limit the size of the economic vectors.
- 8) *All revenues and expenses are estimated on a monthly basis*.
- 9) *Only edges fail*. Nodes do not fail.

- 10) All edges fail identically and independently in all time periods.
- 11) The analysis only deals with the two-terminal ($s - t$) network availability.
- 12) Performance failures and costs are ignored in this analysis. This limits the size and complexity of the analysis.

B. Uncertainty

We will now add *uncertainty* to stochastic model assumptions (1,2 and 3) by incorporating *stochastic* rather than *deterministic* revenue, cost, and component failure functions. We will also assume that all network failures in both networks are detected and solved. Revenue *uncertainty* is usually treated as receiving *less* than expected (or contracted) payments for services. Typically, accountants will assign a 'reserve' for bad credit extended to customers. It is important to model revenue stochastically rather than as a weighted average to account for the *time value* of the revenue vector. The same reasoning applies to modeling network and component costs.

- 1) $A_{Net_A} = \$1500$ occurs with a probability of 0.95; $A_{Net_A} = \$1300$ occurs with a probability of 0.05 of the time when one or more customers do not pay. This amounts to a \$200 loss of revenue in the month that it occurs. $A_{Net_B} = \$1550$ occurs with a probability of 0.95; $A_{Net_B} = \$1300$ occurs with a probability of 0.05 of the time when one or more customers do not pay. This amounts to a \$250 loss of revenue in the month that it occurs.
- 2) The network failure cost function for Net_A and Net_B exhibits a *uniform pdf* between \$75 per failure and \$125 per failure with an average of \$100 per failure. The network *MTBF* function has a *uniform pdf* between 5667 hours per failure and 7667 hours per failure with an average of 6667 hours per failure.
- 3) The component failure cost function for Net_A exhibits a *uniform pdf* between \$5 per failure and \$15 per failure with an average of \$10 per failure. The component *MTBF* function has a *uniform pdf* between 168 hours per failure and 228 hours per failure with an average of 198 hours per failure.

V. QUESTIONS ANSWERED

A simulation was run on both networks using the assumptions described above. The simulation consisted of 10,000 runs for each month for both networks. Minimum, mean, median, and maximum along with the 5th and 95th quantile values for *revenues*, *lost revenues* and *component repair costs* were captured. The *ERVs* for both networks were also solved using *full path enumeration* to get a *deterministic* set of values. This analysis is used to answer the questions raised earlier.

1. What is the range of expected availability for a given network?

Monthly availability for Network_A ranges from a minimum of 0.9962 in month₃ to a maximum of 0.9983 in month₄ with an average of 0.9970 over the 24 month duration. These values compare with the exact monthly availability value for Network_A of 0.9997. The monthly availability for Network_B ranges from a minimum of 0.9996 in months_{0,11,21} to a maximum of 1.0 in months_{15,17,22} with an average of 0.9998 over the 24 month duration. These values compare with the exact monthly availability value for Network_B of 0.99998.

2. What is the range of expected Economic Reliability Values for a given network?

Table II compares the exact and stochastic *ERVs* for both networks over a 24 month period. The first observation to note is that, for the duration of the analysis, the stochastic *Network A* model always has a greater *ERV* than stochastic *Network B* model. This is a different result than is obtained from solving an *analytic* model. The *analytic* choice over a 24 month period is *Network B*.

Time	Analytic <i>Net_A</i>	Analytic <i>Net_B</i>	Stochastic <i>Net_A</i>	Stochastic <i>Net_B</i>
0	1196.91	247.36	1184.58	221.13
1	2381.97	1482.37	2357.69	1430.95
2	3555.29	2705.15	3518.68	2627.69
3	4717.00	3915.82	4668.45	3812.46
4	5867.20	5114.51	5806.76	4986.87
5	7006.02	6301.33	6932.82	6149.30
6	8133.56	7476.40	8049.11	7300.59
7	9249.94	8639.83	9153.63	8440.15
8	10355.26	9791.75	10247.23	9569.16
9	11449.64	10932.26	11330.45	10686.28
10	12533.19	12061.48	12402.32	11791.63
11	13606.01	13179.52	13463.29	12886.36
12	14668.20	14286.48	14514.58	13970.91
13	15719.88	15382.49	15555.11	15043.87
14	16761.14	16467.65	16585.21	16106.32
15	17792.10	17542.06	17605.11	17158.98
16	18812.85	18605.84	18615.53	18200.40
17	19823.49	19659.08	19615.43	19232.11
18	20824.13	20701.89	20606.20	20254.02
19	21814.86	21734.38	21587.05	21264.86
20	22795.77	22756.65	22557.30	22265.48
21	23766.98	23768.79	23518.55	23256.84
22	24728.57	24770.92	24470.17	24238.40
23	25680.64	25763.12	25411.69	25210.39

TABLE II
ANALYTIC / STOCHASTIC ERV TABLE

The discrepancies between the *stochastic* and *analytic* models occur in the *revenues*. They are lower in the *stochastic* models and most notably in the network costs which are higher in the *stochastic* models. These differences disappear as one narrows the range of the costs and the *MTBF* values to the *average*. The net result shows the difference in the *analytic ERVs* decreases more quickly than the *stochastic ERVs*.

3. What is the impact of uncertainty on expected network availability?

Allowing *uncertainty* into availability calculations (in the form of a probabilistic function for component MTBF), produced a slightly lower average network availability over the two year forecast for both networks (see Table III). The proportional error difference for the expected network availability of these networks is 0.27% for $Network_A$ and 0.019% for $Network_B$.

Network	Min. Avail.	Avg. Avail.	Exact Avail.	Max. Avail.
Net_A	0.9962	0.9970	0.9997	0.9983
Net_B	.9996	.9998	.99998	1.0

TABLE III
AVAILABILITY COMPARISON TABLE

4. What is the impact of uncertainty on network Economic Reliability Value?

Uncertainty has a slightly greater impact on the economic elements that form the *ERV* than it has on network availability. This is due in part to the asymmetric nature of *uncertainty* leading to *lower revenue* and *higher failure costs*. This impact can be seen in Table IV.

Month	Min. Annuity	5 th Q Annuity	Mean Annuity	Median Annuity	95 th Q Annuity	Max. Annuity
0	952.06	1111.67	1184.58	1190.69	1244.94	1301.74
1	930.11	1113.39	1184.84	1190.60	1245.50	1319.61
2	942.99	1113.84	1184.32	1190.07	1244.95	1305.40
3	949.17	1116.69	1184.60	1190.40	1245.31	1298.22
4	950.34	1113.21	1184.53	1190.67	1244.99	1299.44
5	962.09	1109.98	1183.50	1189.80	1244.26	1296.89
6	960.24	1119.12	1184.97	1190.05	1245.41	1299.84
7	966.44	1110.77	1184.19	1190.15	1246.20	1312.84
8	948.55	1107.64	1184.21	1190.02	1244.21	1301.29
9	946.44	1109.43	1184.70	1190.24	1244.93	1300.53
10	921.13	1108.02	1184.02	1190.19	1245.17	1296.77
11	943.77	1112.36	1183.69	1190.00	1245.15	1312.01
12	960.12	1111.44	1184.62	1191.04	1244.83	1323.03
13	944.08	1111.83	1184.22	1189.78	1244.07	1301.74
14	949.72	1106.15	1184.07	1190.03	1244.66	1308.01
15	949.22	1111.66	1184.07	1190.28	1245.47	1302.65
16	893.83	1114.04	1184.80	1190.36	1245.06	1300.81
17	949.10	1108.18	1184.18	1189.74	1245.72	1313.72
18	946.72	1110.85	1185.11	1191.04	1244.79	1302.16
19	929.60	1112.14	1184.97	1191.39	1245.68	1309.00
20	925.56	1103.78	1183.88	1190.40	1245.17	1310.42
21	940.28	1114.16	1184.64	1190.44	1244.48	1300.13
22	953.94	1114.66	1184.49	1189.93	1243.06	1299.00
23	948.04	1106.08	1183.64	1189.77	1243.49	1302.75
Average	944.31	1111.30	1184.37	1190.30	1244.90	1304.92

TABLE IV
NETWORK_A MONTHLY ANNUITY METRICS TABLE

The exact monthly *annuity* for $Network_A$ is 1193.96. The percentage error for the median stochastic annuity and the exact annuity is 0.31%, which compares with the relative network availability error of 0.27%. The percentage error of the expected stochastic *ERV* from the expected closed form *ERV* is 0.32% for $Network_A$ and 0.89% for $Network_B$. Figure 2 plots the relative *ERV* error for both networks over time. The comparatively large relative error in $Network_B$ in the first month is caused by the small size of $Network_B$ *ERV* in the first month.

VI. SUMMARY AND CONCLUSIONS

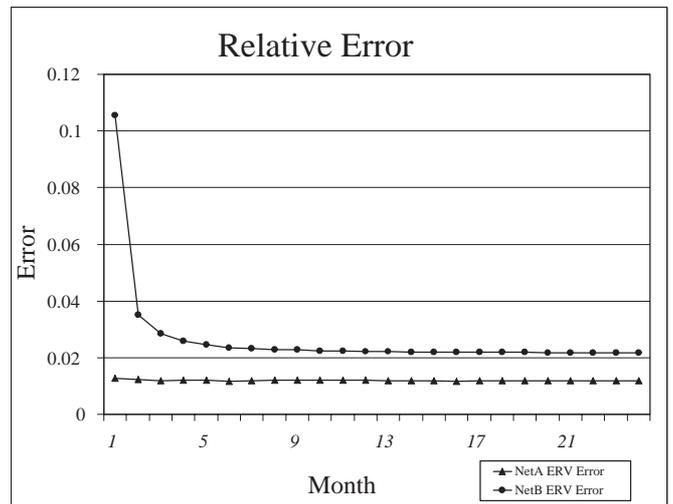


Fig. 2. Network_A & Network_B relative error over time

Stochastic reliability techniques have long been used to estimate network availability. Even simple stochastic models of networks can provide reasonable estimates of both economic efficiency and network availability with greater behavioral realism than comparable analytical methods. We have applied these techniques to estimate the expected economic impact of network and component availability and validated the results against a closed form solution.

The initial results are encouraging. We have taken our *Economic Reliability Analysis* framework and incorporated stochastic methods into it with satisfactory results. Further research and application is planned. Current plans are to apply the methods described in this paper to an operational network.

REFERENCES

- [1] P. Bell, R. Walling, and J. Peacock, "Costing the access network - an overview," *British Telecom Technology Journal*, vol. 14, no. 2, pp. 128 – 132, April 1996.
- [2] Y. Dutuit, E. Châtelet, J. P. Signoret, and P. Thomas, "Dependability modelling and evaluation by using stochastic petri nets: application to two test cases," *Reliability Engineering and System Safety*, vol. 55, pp. 117 – 124, 1997.
- [3] K. Goseva-Popstojanova and K. Trivedi. (2001, November 9) Stochastic modeling formalisms for dependability, performance and performativity. [Online]. Available: <http://link.springer.de/link/service/series/0558/bibs/1769/17690403.htm>
- [4] D. W. Mitchell and G. M. Gelles, "Risk-value models: Restrictions and applications," *European Journal of Operational Research*, vol. 145, pp. 109 – 120, 2003.
- [5] H. J. Pradwarter and G. I. Schuëller, "On advanced monte carlo simulation procedures in stochastic structure dynamics," *International Journal of Non-Linear Mechanics*, vol. 32, no. 4, pp. 735 – 744, 1997.
- [6] P. Shahabuddin, "Rare event simulation in stochastic models," in *Proceedings of the 27th conference on Winter simulation*. New York, N.Y., United States: ACM Press, 1995, pp. 178 – 185. [Online]. Available: <http://doi.acm.org/10.1145/224401.224460>
- [7] E. J. Stoker and J. B. Dugan, "A framework for economic reliability analysis," University of Virginia, Charlottesville, VA, United States, Technical Report TR-ES2003, February 2003.
- [8] M. R. Thomas, P. Bell, C. A. Gould, and J. Mellis, "Fault rate analysis, modelling and estimation," *British Telecom Technology Journal*, vol. 14, no. 2, pp. 133 – 139, April 1996.
- [9] J. Tindle, S. J. Brewis, and H. M. Ryan, "Advanced simulation and optimisation of the telecommunications network," *British Telecom Technology Journal*, vol. 14, no. 2, pp. 140 – 146, April 1996.

- [10] Y. T. Yoon and M. D. Ilić, "Independent transmission company (itc) and markets for transmission," in *Power Engineering Society Summer Meeting*, vol. 1. Los Alamitos, CA, United States: IEEE Press, 2001, pp. 229 – 234.
- [11] —, "Price-cap regulation for transmission: objectives and tariffs," in *Power Engineering Society Summer Meeting*, vol. 2. Los Alamitos, CA, United States: IEEE Press, 2001, pp. 1052 – 1057.
- [12] —, "A possible notion of short-term value-based reliability," in *Power Engineering Society Winter Meeting*, vol. 2. Los Alamitos, CA, United States: IEEE Press, 2002, pp. 772 – 778.

Ed Stoker (born 1951) received his B.A., M.A., and M.B.A in 1975, 1980 and 1981 respectively from the University of Pittsburgh, Pittsburgh PA, USA and has worked in computer network engineering since that time. He is currently a PhD. candidate in Computer Engineering at the University of Virginia.

Joanne Bechta Dugan (F'00) received the B.A. degree (1980) in mathematics and computer science from La Salle University, Philadelphia, PA, and the M.S. and Ph.D. degrees in 1982 and 1984, respectively, in electrical engineering from Duke University, Durham, NC.

She is Professor of Electrical and Computer Engineering with the University of Virginia. She has performed and directed research on the development and application of techniques for the analysis of computer systems that are designed to tolerate hardware and software faults. Her research interests include hardware and software reliability engineering, fault tolerant computing and mathematical modeling using dynamic fault trees, Markov models, Petri nets and simulation.

Dr. Dugan was an Associate Editor of the IEEE TRANSACTIONS ON RELIABILITY for 10 years, and is Associate Editor of the IEEE TRANSACTIONS ON SOFTWARE ENGINEERING. She served on the USA National Research Council Committee on Application of Digital Instrumentation and Control Systems to Nuclear Power Plant Operations and Safety.

APPENDIX

Symbol	Definition
BPR	Business Process Re-engineering
DCF	Discounted Cash Flow
ERA	Economic Reliability Analysis
ERV	Economic Reliability Value
NPV	Net Present Value
A	Revenue process as a function of network design and finance
\bar{A}	Revenue vector produced by A
B	Lost revenue process as a function of network failure
\vec{B}	Lost revenue vector produced by B
$\vec{T}F_n$	Task _n failure vector
$\vec{T}C_n$	Task _n repair cost per failure vector
$\vec{P}F_n$	Process _n failure vector
$\vec{P}C_n$	Process _n repair cost per failure vector
C	Lost revenue process as a function of QoS failure
\vec{C}	Lost revenue vector produced by C
LR	Lost revenue process: $B + C$
\vec{LR}	Lost revenue vector: $\vec{B} + \vec{C}$
D	Component repair cost process as a function of network failures
\vec{D}	Component repair cost produced by D
E	Component repair cost process as a function of QoS failures
\vec{E}	Component repair cost vector produced by E
F	Other recurring cost process based on normal network operations
\vec{F}	Other recurring cost vector produced by F
OC	Other recurring cost process unrelated to normal network operations
\vec{OC}	Other recurring cost vector produced by OC
RC	Recurring Cost process: $D + E + F + OC$
\vec{RC}	Recurring cost vector: $\vec{D} + \vec{E} + \vec{F} + \vec{OC}$
G	Capital cost process as a function of network design and finance
\vec{G}	Capital cost vector produced by G
H	Annuity process as a function of reliability
\vec{H}	Annuity vector produced by H
DR	Discount rate process
\vec{DR}	Discount rate vector produced by DR
EC	ERV Contribution process
\vec{EC}	ERV Contribution vector produced by EC

TABLE V
NOTATION

A MONTE CARLO DISPERSION ANALYSIS OF A ROCKET FLIGHT SIMULATION SOFTWARE

F. SAGHAFI, M. KHALILIDELSHAD

*Department of Aerospace Engineering
Sharif University of Technology
E-mail: saghafi@sharif.edu
Tel/Fax: 0098216022731*

Abstract: A Monte Carlo dispersion analysis has been completed on a medium range solid propellant rocket simulation software. This analysis has been carried out to find the optimum values of the rocket fincant angle and spin motor torque for the best impact point error, and the probability of flight-to-target success. The simulation is developed based on rotating earth equations of motion and has many components. This paper describes the methods used to accomplish the Monte Carlo analysis and gives an overview of the processes used in the implementation of the dispersions. Selected results from 70000 Monte Carlo runs are presented with suggestions for the values of the desired parametres.

Keywords: Flight Simulation, Stochastic, Monte Carlo, Dispersion Analysis, Rocket

INTRODUCTION:

If all characteristics of a rocket, together with atmospheric conditions are exactly equal to a set of predicted values, the rocket will fly on a known trajectory and hits a target point. This trajectory is called nominal trajectory. In practice, there are always some differences between the real and predicted values. These are mainly due to manufacturing, measurement and atmospheric modeling errors. These differences make the rocket not to fly exactly on its nominal trajectory, and to hit a target. Therefore, there are always some errors between the positions of a desired and a real impact point. Estimation of these errors is very important from the operational point of view. Also, investigation of the error sources and their effects can help a rocket designer to optimize design parameters for the lowest impact point error.

In this study, a Monte Carlo dispersion analysis has been completed on flight simulation software of a rocket to investigate its impact point error. The rocket is a solid propellant medium range type with 320-km range, 9-m length, 0.5-m diameter and a maximum weight of 3500 kg. Four cross type stabilizer fins and a spin motor provide the rocket static stability. The rocket six-degree-of-freedom simulation was used to repeatedly fly a near nominal trajectory. This simulation software has been developed based on the rotating earth equations of motion. It has many components such as, aerodynamics, mass properties, equations of motion, atmospheric model, wind model at different altitudes and a gravity model. No intervention was required to simulate a complete trajectory because there is no

control on a rocket after it is launched. This allows multiple runs to be directly compared.

The Monte Carlo method of dispersion analysis uses a given system model (in this case, the rocket flight mathematical model) and introduces statistical uncertainties on as many of the individual parameters as practical. In this work, a set of forty-one parameters is selected in different categories including Aerodynamics, Propulsion, Atmosphere and Wind, Mass and Inertia, Dimension and Launching. A uniform distribution of uncertainties around the nominal values of each parameter is considered. The range of magnitudes of uncertainties are defined based on a set of known observations and a first step individual error analysis. Uniformly distributed random values in the defined ranges were selected and applied to the simulation parameter. Each Monte Carlo simulation run had different random variation of the dispersions.

The number of Monte Carlo runs containing uncertainty combinations that result in failure to complete a normal flight were identified. Thus, the probability of flight-to-target success was established. Although, establishing the probability of flight-to-target success was one of the primary goals of this analysis, other objectives such as system validation also were accomplished. Completing, the Monte Carlo analysis also allowed for the identification of weaknesses in rocket design and margins in specific rocket parameter.

The objective of this report is to demonstrate how Monte Carlo simulation analysis can be used to identify and analyze the trajectory problems for a rocket and provide some preliminary results. Results

are presented for 70000 Monte Carlo runs done in this analysis in the form of dispersion plots of different parameters including maximum angular speed, flight time, maximum speed, maximum angle of attack, range error, directional error and radial error.

DISPERSION MODELS:

The dispersions used in the Monte Carlo simulation has been applied to the rocket dynamics and external environment models. The models modified in the rocket simulation to include dispersion capabilities were the aerodynamics, mass properties, propulsion,

atmospheric and launching models. Table 1 shows a list of forty-one uncertainty parameters that have been used in this work. It is tried to consider all the important parameters except the aerodynamic coefficients. A similar study has already been carried out to investigate the effect of the aerodynamic coefficients uncertainties on the rocket impact point error when all other parameters were in their nominal values [Sarikhani and Roshanaiyan, 2002]. Therefore, in the present work, for the sake of computational time reduction, efforts focused to find out the errors associated with the other parameter uncertainties.

Table 1: Uncertainty parameters and ranges

	Parameter definition	Uncertainty range	Unit
1	Launching pitch angle	[-0.3 0.3]	deg
2	Launching yaw angle	[-0.5 0.5]	deg
3	Fuel burnning time	[-1.0 1.0]	sec
4	Fuel mass	[-1.0 1.0]	%
5	Rocket gross weight	[-0.75 0.75]	%
6	Angular thrust vector deviation in xz plane	[-0.3 0.3]	deg
7	Angular thrust vector deviation in xy plane	[-0.3 0.3]	deg
8	Linear thrust vector deviation in body x direction	[-20 20]	mm
9	Linear thrust vector deviation in body y direction	[-10 10]	mm
10	Linear thrust vector deviation in body z direction	[-10 10]	mm
11	Thrust	[-1.0 1.0]	%
12	Rocket lenght	[-50 50]	mm
13	Rocket diameter	[-2 2]	mm
14	Moment of inertia Ixx, (with fuel)	[-2 2]	%
15	Moment of inertia Ixx, (without fuel)	[-2 2]	%
16	Moment of inertia Iyy, (with fuel)	[-2 2]	%
17	Moment of inertia Iyy, (without fuel)	[-2 2]	%
18	Center of mass position, (with fuel)	[-20 20]	mm
19	Center of mass position, (without fuel)	[-20 20]	mm
20	Air density	[-5 5]	%
21	Wind speed, zero altitude	[-2 2]	m/s
22	Wind direction, zero altitude	[-2 2]	deg
23	Wind speed, 1km altitude	[-5 5]	m/s
24	Wind direction, 1km altitude	[-5 5]	deg
25	Wind speed, 2km altitude	[-5 5]	m/s
26	Wind direction, 2km altitude	[-5 5]	deg
27	Wind speed, 5km altitude	[-10 10]	m/s
28	Wind direction, 5km altitude	[-5 5]	deg
29	Wind speed, 10km altitude	[-10 10]	m/s
30	Wind direction, 10km altitude	[-5 5]	deg
31	Jet force damping coefficient	[-0.1 0.1]	ton.m/s
32	Jet moment damping coefficient	[-0.2 0.2]	ton.m ² /s
33	Launcher coefficient of friction	[-15 15]	%
34	Spin motor starting time	[-5 5]	%
35	Spin motor operation time	[2 2]	%
36	Spin motor torque	[-0.0 0.0]	%
37	Izz & Iyy difference	[-0.2 0.2]	%
38	Product moment of inertia, Ixy	[-5 5]	% of Ixx
39	Product moment of inertia, Ixz	[-5 5]	% of Ixx
40	Product moment of inertia, Iyz	[-5 5]	% of Ixx
41	Fincant angle	[-0.0 0.0]	deg

The limits of uncertainties presented in Table 1 were based on a set of previously known observations and measurements completed with a first step individual error analysis. In this individual error analysis, a range of values in the defined limits, Table 1, was given to each parameter and simulation was run several times. Using simulation results, it was possible to plot the impact point distance error vs different parameters variation. Some of these plots

are shown in Figs. 1-6. The plots were then used to find a good estimation for each parameter uncertainty, so that the impact point error was in a normal practically observed range. The analysis discussed in this report tested over the entire uniform distribution mainly to ensure that most of the worst possible cases are considered, and to identify the probability of flight-to-target success.

Fig 1: The effect of thrust misalignment (ϵ_1) on dispersion

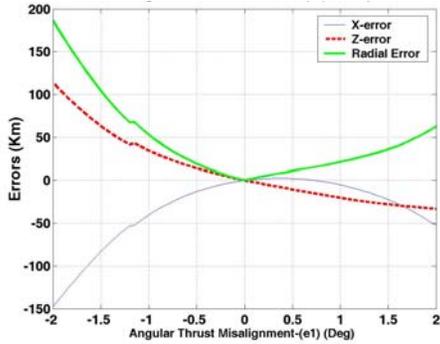


Fig 2: The effect of thrust error on dispersion

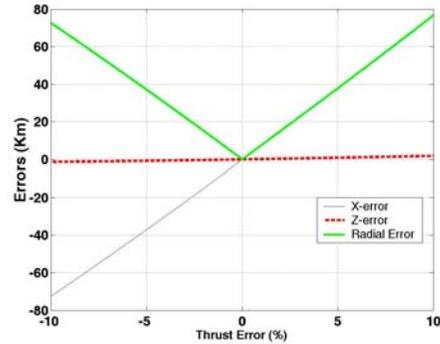


Fig 3: The effect of air density error on dispersion

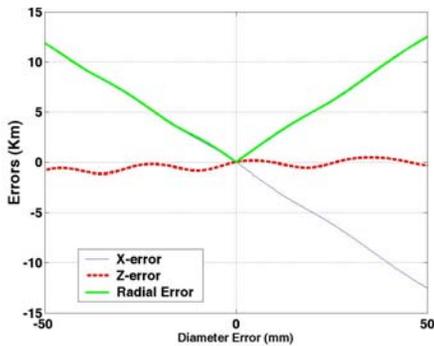


Fig 4: The effect of rocket diameter error on dispersion

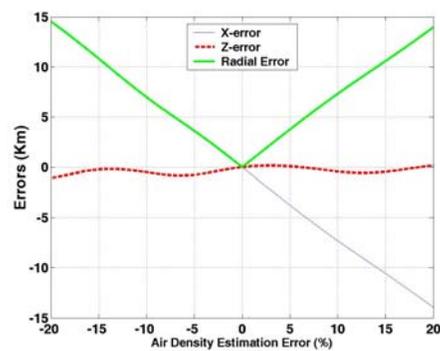


Fig 5: The effect of elevation angle error on dispersion

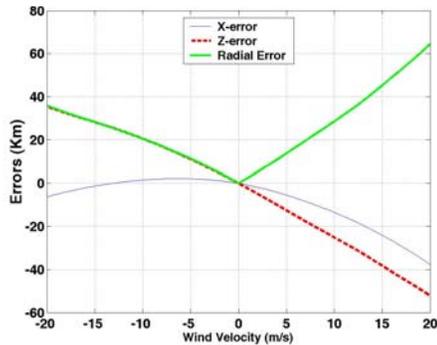
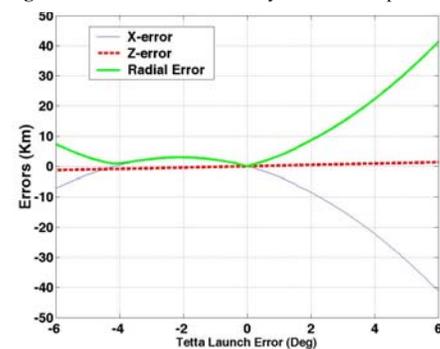


Fig 6: The effect of wind velocity at $h=0$ on dispersion



METHODS OF APPROACH:

Monte Carlo analysis estimates the statistics of random variables by analyzing the statistics of many

trials. One important question associated with Monte Carlo analysis is determining the number of trials needed before the statistics of a variable can be estimated with reasonable accuracy. In this work, the

number of Monte Carlo trials were determined based on the work previously done by [Williams, 2001], and was selected to be one thousand trials. When the desired number of runs was determined, files were generated containing all relevant dispersions. Dispersion values were randomly selected from a uniform distribution, and then stored in individual input files.

This collection of files was sequentially run from a main script, which directed the storage of relevant data. Additional scripts were written to process the data for analysis. Because each simulation run lasted approximately two minutes and was recording large amounts of data, storing the relevant data for flight without storing the entire data file generated by the simulation became necessary. For this reason, scripts were developed that took “snapshots” of the data. This snapshot process was performed on the entire data file after a run was completed. These scripts extracted the data at the beginning of each flight phase and directed the storage into separate and much smaller files. In this way, most of the data were discarded, and the process of completing many runs could be automated without exceeding memory limitations.

One of the most important reasons for the present work to be carried out was to find out the effect of the rocket fincant angle and spin motor torque on the minimum impact point error, and select the best

combination of them. These are two terms which can be set relatively accurate during the rocket manufacturing time. Therefore, these two parameters were selected as control parameters in running the simulation software. A set of known discrete values for fincant angle containing $-0.5, -0.2, 0.0, 0.3, 0.5, 0.7$ and 1.0 deg. was selected. For each of the selected values, dispersion analysis has been carried out over a range of spin motor torque from 0 to 9000 N.M with a step of 1000 N.M. For each pair of the fincant angle and spin motor torque, 1000 simulation were run, totally 70000 times, in which all dispersions were randomly varied over a uniform distribution. For each simulation run, the maximum value of the rocket angle of attack, sideslip angle, linear and angular velocities and accelerations, dynamic pressure and attitude angles during flight were obtained and stored together with the flight time, range, directional and radial impact point errors and the random values selected automatically for the other thirty-nine parameters in Table 1.

SIMULATION RESULTS:

This section presents results for the 70000 Monte Carlo runs completed in this analysis. Some typical distribution plots of various variables are shown in Figs. 7-10. The plotted data in these figures are generated with 0.5 deg. fincant angle and maximum spin motor torque. As shown, a reasonable normal distribution of different variables is observed.

Fig 7: The freq. distrib. of flight time (Max. Spin, Fin=0.5 deg)

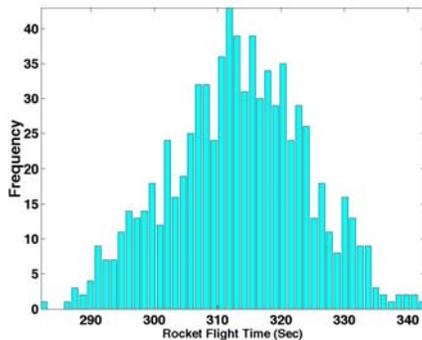


Fig 8: The freq. distrib. of “Max Wx” (Max. Spin, Fin=0.5deg.)

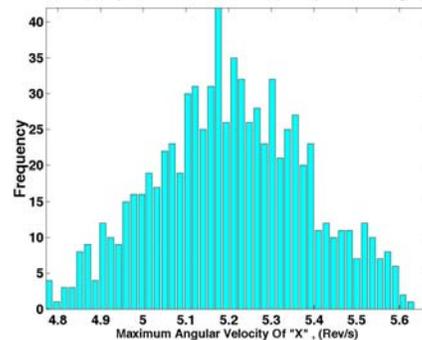


Fig 9: The freq. distrib. Of Max. A.O.A (Max Spin, Fin=0.5 deg.)

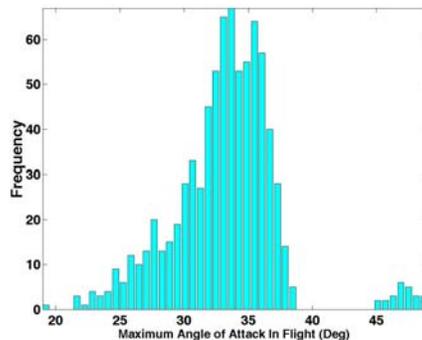
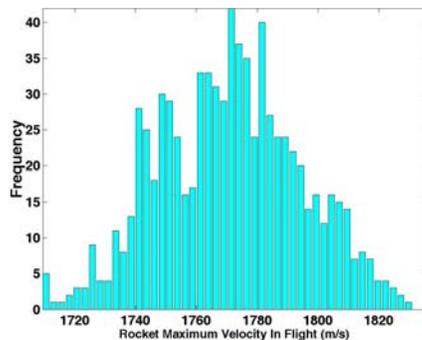


Fig 10: The freq. dist. of rocket Max. Vel. (M. Spin, Fin=0.5 deg.)

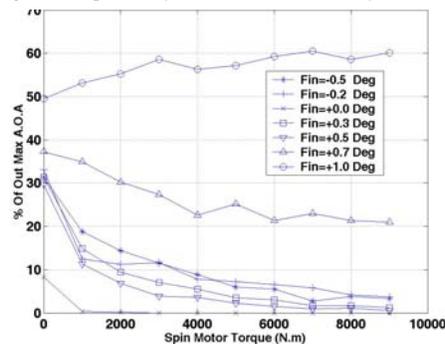


It should be pointed out that the aerodynamic model of the simulation software is valid over a certain range of the angle of attack and sideslip, therefore, any simulation run in which the maximum angle of attack or sideslip angle has exceeded the model validity range, can not be invoked. The generated data for these cases are only meaningless numerically calculated values and does not demonstrate the real flight specifications of the rocket. For a correct analysis, these kind of data must be removed from the total set of the output data. Similarly, there is a structural limitation on the maximum tolerable acceleration of the rocket. Again, those simulation runs showing unacceptable maximum values of longitudinal and lateral rocket accelerations are not useful and should be removed. In this analysis, acceptable maximum longitudinal and lateral accelerations are +/-20g and +/-10g,

respectively [Saghafi and Khalilidelshad, 2003]. Also, the validity range of the aerodynamic model is up to 50 deg. angle of attack or sideslip.

The percentage of out of limit maximum angle of attack runs to the total runs for various combinations of fincant angle and spin motor torque are shown in Fig. 11. In general, the percentage of out of limit angle of attack is increased with increasing fincant angle and decreasing spin motor torque. It should be noted that flight in high angle of attack and load factor (high acceleration) as big as the given limiting values is practically impossible for an uncontrolled rocket. Therefore, flight-to-target in these cases are considered to be unsuccessful. Thus, the probability of flight-to-target success can be estimated by dividing the number of the out of limit simulation runs to the total number of runs.

Fig 11: The percentage of out of limit max. angle of attack



Having removed the unacceptable simulation data, the statistical characteristics such as mean values and standard deviations were calculated and used for examining the results. The variations of the mean value and standard deviation of the rocket directional impact point errors in different fincant angles versus spin motor torque are shown in Figs.

12-13. These results and the other similar plots for the rocket range and radial impact point errors, not presented here, show the great effect of the spin motor torque on the rocket impact point error reduction. Therefore, an obvious conclusion is that the maximum spin motor torque is the best value for all flight conditions.

Fig 12: The mean value of X-error in different launches

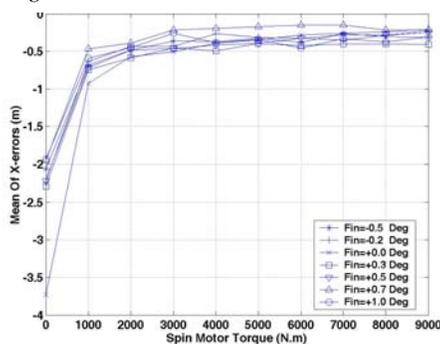
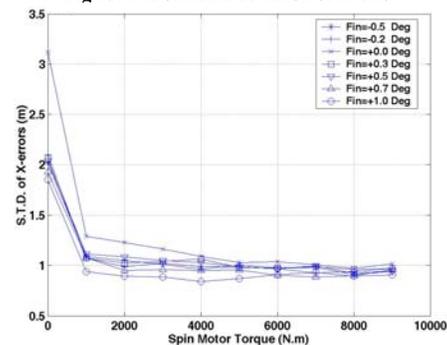


Fig 13: The standard deviation of X-error



Having set the torque, the fincant angle should be selected. A fincant angle corresponding to a minimum mean value and standard deviation of the impact point errors in maximum spin motor torque, would be the optimum value. To find this optimum value, dispersion plots for the range, directional and radial impact point errors in different fincant angles and maximum spin motor torque were used. Typical dispersion plots of these kinds for 0.3 and 1.0 deg. fincant angles are shown in Figs. 14-17. Using these plots, the cumulative probability of impact point errors to be in predefined limits, could be determined. The cumulative probabilities of the range, directional and radial impact point errors for different fincant angles and limitations are shown in

Figs. 18-20. As shown, the cumulative probability does not change noticeably with the fincant angles. In fact, there is no fincant angle which have a considerable effect on the cumulative probability of errors. Therefore, from the error point of view any fincant angle can be selected for the rocket. However, other considerations such as the severe effect of negative fincant angles on the rocket lateral acceleration, or fincant angles bigger than one, on impact point error, have limited the selectable values in the range of 0.0 to 1.0. Considering the possibility of manufacturing errors and to be far enough from the limits, a value of 0.5 deg. for fincant angle is proposed.

Fig 14: The probability distribution of X-error

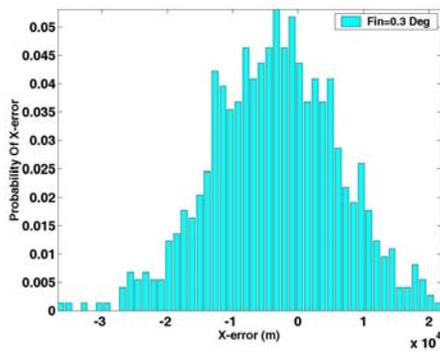


Fig 15: The probability distribution of X-error

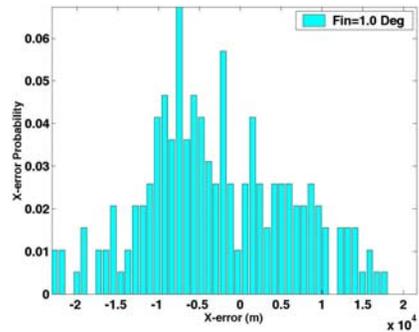


Fig 16: The probability distribution of radial error

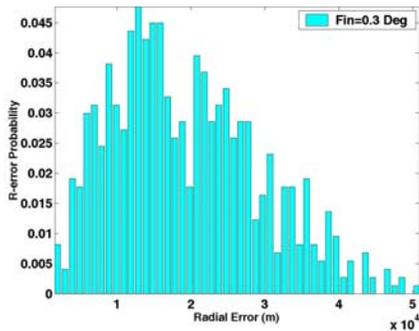


Fig 17: The probability distribution of radial error

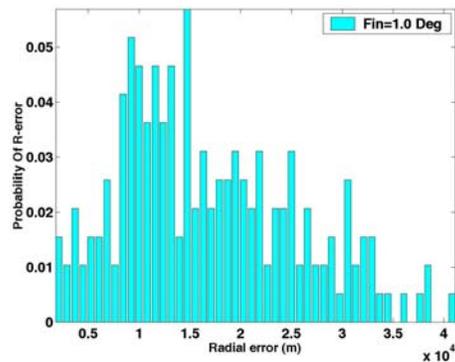


Fig 18: The cumulative probability of different ranges

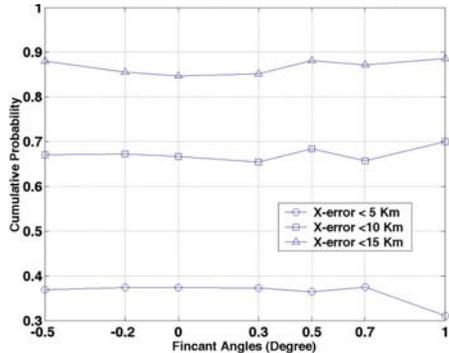


Fig 19: The cumulative probability of different Z-errors

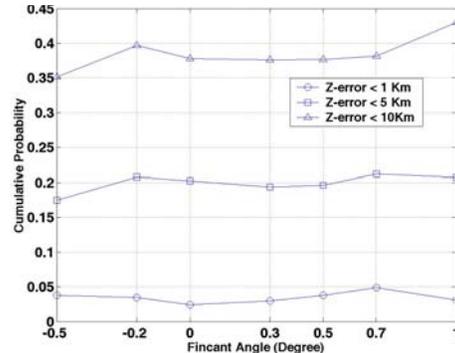
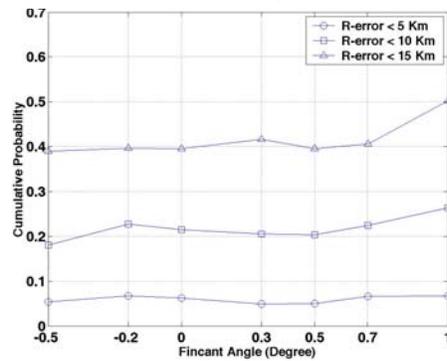


Fig 20: The cumulative probability of different radial errors



SUMMARY:

A Monte Carlo dispersion analysis of a medium range solid propellant simulation software was undertaken to show the usefulness of this type of analysis in the identification of design weaknesses in margins of specific parameters. Also, it is used to find out the optimum values of the rocket fincant angles and spin motor torque for the lowest impact point error, and the probability of flight-to-target success.

Results were presented for the selected conditions in the form of dispersion and statistical plots. This Monte Carlo analysis showed that the spin motor torque has a great effect on the rocket stability and its impact point error. Instead, the fincant angle has no noticeable effect on these parameters in high values of spin motor torque. Finally, regarding other considerations, a fincant angle of 0.5 deg. together with a maximum value for spin motor torque are proposed.

REFERENCES:

Sarikhani A. and Roshanaiyan J. 2002, "Combined Analytical and Numerical Methods for Estimation of the Effect of Modified Aerodynamic Coefficients on the Impact Point Errors". In *Proc. 10th Annual Conference of the Iranian Society of Mechanical Engineering, ISME2002*, Tehran. (In Farsi)

Peggy S. W. 2001, "A Monte Carlo Dispersion Analysis of the X-33 Simulation Software", *AIAA-2001-4067*.

Saghafi F. and Khalilidelshad M. 2003, "Dispersion Analysis of a Solid Propellant Rocket Using Statistical Simulation", In *Proc. 4th Annual Conference of the Iranian Society of Aerospace Engineering, AERO2003*, Tehran. (In Farsi)

BIBLIOGRAPHY:

Newman M.E. and Barkema G.T. 1999, "Monte Carlo Methods in Statistical Physics", *Oxford University Press*.

Monahan G.E. 2000, "Introduction to Monte Carlo Simulation", *University of Illinois, Cambridge University Press*.

Broddner S. 1970, "Effect of High Spin on the Internal Ballistics of a Solid Propellant Rocket Motor", *Astronautica ACTA, Vol. 15, No. 4*.

Calise J. and Shirbing A.E. 2001, "An Aerodynamic Control for Direct Spinning Projectiles", *The American Institute of Aeronautics and Astronautics Inc.*

Lucy M.H. "Spin Acceleration Effects on Some Full-Scale Rocket Motors", *Journal of Spacecraft, Vol. 5, No. 2, February 1968*.

BIOGRAPHY:



Dr. Saghafi has received his B.Sc. (1987) and M.Sc. (1990) in Mechanical Engineering from Shiraz University in Iran, and his PhD (1996) in Aerospace Engineering from Cranfield University in England. Currently, he is a professor of Flight Dynamics

in the Department of Aerospace Engineering of Sharif University of Technology. His research interest is the modeling and simulation of aerospace vehicle dynamics.

INTERACTIONS BETWEEN TRANSMISSION POWER AND TCP THROUGHPUT FAIRNESS IN WIRELESS CDMA NETWORKS¹

LAURA GALLUCCIO, ALESSANDRO LEONARDI, GIACOMO MORABITO

*Dipartimento di Ingegneria Informatica e delle Telecomunicazioni
University of Catania*

V.le A. Doria, 6 - Catania 95125 (Italy)

Email: {laura.galluccio, alessandro.leonardi, giacomo.morabito}@diit.unict.it

Abstract: TCP protocols have scarce performance when transmitting over wireless channels. In fact, wireless medium is characterized by multipath fading and high time variability which lead to a great amount of losses in the transmitted packets. In such environments many new problems arise, the most serious of which are the decrease of throughput performance and the unfairness due to the great difference in the Round Trip Times (RTTs) of the connections sharing the same wireless link. In particular, the unfairness is related to the fact that the lower is the RTT of a connection, the higher is its throughput.

In literature many solutions aimed at improving TCP fairness through modifications of the *congestion avoidance* algorithm have been proposed. However, all these solutions are not suitable for wireless networks because they still interpret transmission losses as congestion clue while it is not. In this paper it is demonstrated that the unfairness problem can be solved through appropriate power management. In particular, by letting connections with longer RTT transmit with stronger power, the proposed approach tries to increase the fairness. The performance evaluation is carried out using the NS2 Simulator and interesting simulation results are provided. In particular, it can be observed that, if compared to standard wireless CDMA solutions for transmission over wireless networks, the proposed approach allows to guarantee a good level of throughput and a higher fairness.

keywords: Analytical and Numerical Simulation, Information Networks, Communication Systems.

1 INTRODUCTION

The use of a TCP protocol over wireless networks introduces problems mainly related to the medium characteristics. In fact, it is known that the TCP protocol, thought to work on traditional wired networks, assumes that all losses and possible delays are due to network congestion. When dealing with wireless links, however, packet losses are mainly due to transmission errors. As a consequence, when a timeout elapses, the TCP protocol decreases immediately its transmission rate thus trying to solve congestion. The sudden reduction in the transmission rate, causes a serious decrease in throughput performance as we can see in [Zorzi et al, 2002]. Another problem is the unfairness in the bandwidth allocation when many connections with different RTT share the same congested network. This behavior is due to the congestion avoidance algorithm commonly used by the TCP protocol. During the *Congestion Avoidance*, the congestion window is increased by one segment when no congestion is detected, and decreased to half its value when congestion

is observed. This mechanism is known as *Additive Increase and Multiplicative Decrease* (AIMD). It is clear that, since the decreasing rate in TCP is about one half for all connections, and the increasing rate is about one segment per RTT, the increase in the transmission rate is not uniform but varies strictly depending on the RTT of each connection [Lakshman and Madhow, 1997]. If TCP works over a wireless network, AIMD is more frequent because of the common link failures which, as said above, are misinterpreted by TCP and considered as congestion clue. If connections with different RTT share the same link, unfairness will be encountered because connections with lower RTT will monopolize resources before slower connections get some bandwidth. In literature, many solutions which aim at improving TCP fairness in various environments have been proposed so far [Henderson et al, 1998; Pilosof et al, 2003]. However, they are mainly based on modifications of the congestion avoidance algorithm and, for this reason, they are suitable only for wired networks. The target of this paper is, instead, to demonstrate that an appropriate power management can give a good improvement in TCP fairness without

¹This paper was partially supported by CEC under contract ANWIRE-IST 2001-38835

requiring any modification to the congestion avoidance algorithm. Based on this approach, the paper will be organized as follows. In Section Analytical Framework, the analytical framework required for evaluating the transmission power levels needed to obtain the desired fairness is introduced. In Section Performance Evaluation the performance of the considered system is evaluated through simulation. Finally, some concluding remarks are drawn in Section Conclusions.

2 ANALYTICAL FRAMEWORK

The target of this section is deriving an analytical relationship linking the loss probability to the power received by a Base Station and related to a given connection. Our study refers to a system composed of M wireless connections in a W-CDMA scenario. Connections employ a TCP protocol and are characterized by different RTT s. Let us say C_i the received power at the Base Station for the i -th connection and let us neglect the thermal noise, because of its lower value with respect to the interference due to the other $M - 1$ mobile users [Wu, 1999; Grandhi et al, 1994; Lee and Lin 1996]. Let us consider the $\frac{E_b}{I_0}$ ratio, which is defined as the ratio between the average energy for an information bit, (E_b), and the power spectral density of interference (I_0). The considered ratio, for the i -th user, evaluated at the Base Station, can be written as follows:

$$\left(\frac{E_b}{I_0}\right)_i = \frac{C_i/R_i}{\left(\sum_{j=1, j \neq i}^M C_j\right)/W} = \frac{W}{R_i} \cdot \frac{C_i}{\sum_{j=1, j \neq i}^M C_j} \quad (1)$$

where R_i is the considered user bit rate, while W represents the chip rate chosen equal to that of a FDD-type W-CDMA interface, which is, for UMTS, equal to 3.84 Mcps [Ojampera and Prasad, 1998]. The ratio W/R_i in Eq. (1) represents the so called Process Gain (P_G) of the W-CDMA system. Looking at Eq. (1) it can be observed that, in a W-CDMA multiple access scenario, if the transmission power and thus the received power of a connection increases, an higher $\frac{E_b}{I_0}$ ratio for the same connection is expected. This means that other connections employing the same link will suffer of higher levels of interference. We maintain, in particular, that a right choice of the transmission power distribution for all connections can give an improvement in the overall fairness. To this aim, let us consider the relationship between the Bit Error Rate (BER) and the $\frac{E_b}{I_0}$ ratio for the i -th connection. Let us suppose to refer to a spread spectrum system employing a BPSK modulation that is the most common type of modulation employed in CDMA systems. In addition, in order to improve the link reliability observed by TCP, forward error correction (FEC) should be introduced. FEC is based on adding a certain amount of redundancy to the data being transmitted.

This redundancy will be used at the destination to correct possible transmission errors which could be encountered. If we consider an information packet of K bits, FEC adds some redundant data, so that the size of the new coded data packet becomes N bits, where $N > K$. According to [Proakis, 1995], the BER for the i -th connection can be written as:

$$BER_i = 12 \cdot Q \left(\sqrt{2 \left(\frac{E_b}{I_0}\right)_i \cdot R \cdot d_H} \right) \quad (2)$$

In the previous equation, R represents the *code rate* defined as $R = \frac{K}{N}$, d_H is the *Hamming distance* representing the number of bits by which two valid codewords differ from each other, while $Q(x) = \frac{1}{2} \cdot \text{erfc}\left(\frac{x}{\sqrt{2}}\right)$, where $\text{erfc}(x)$ is the *Complementary Error Function*. Since it is known that the relationship between the packet loss probability P_{Loss_i} , and BER_i can be written as:

$$P_{Loss_i}^2 = 1 - (1 - BER_i)^{PS} \quad (3)$$

being PS the packet size (in bit), Eq. (3) can be inverted to derive BER_i and thus find a relationship between $\left(\frac{E_b}{I_0}\right)_i$ and P_{Loss_i} .

3 PERFORMANCE EVALUATION

3.1 Simulation Model

In order to solve the unfairness problem and guarantee an equal throughput for all the connections employing the same link, we made a set of simulations to derive the P_{Loss} values which satisfy the condition of giving the same throughput.

In our simulation, we refer to a topology like the one shown in Fig. 1.

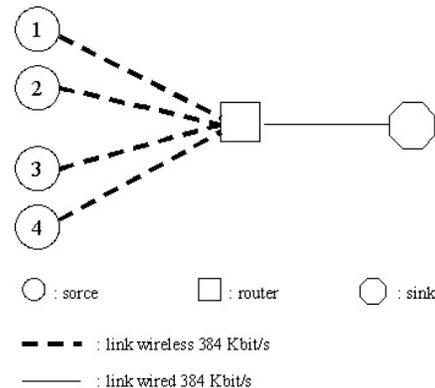


Fig. 1. Network Topology

Four mobile terminals, using a CDMA multiple access interface, communicate with a Base Station through a shared wireless link. This Station is equipped

²The formula applies if we assume that bit errors are independently distributed

with a router with an output link connecting to a sink by using a wired link. Connections are grouped in two sets, consisting, each of them, of two connections with the same RTT. The Process Gain value was assumed to be the same for the four connections, i.e. $P_G=10$, while three different scenarios for the RTT and two set of connections were considered:

- Case I: two of the four connections have the same fixed RTT equal to 100 ms while the other two connections have the same RTT, but variable in the range [100, 400] ms.
- Case II: two of the four connections have the same fixed RTT equal to 200 ms while the other two connections have the same RTT, but variable in the range [100, 400] ms.
- Case III: two of the four connections have the same fixed RTT equal to 300 ms while the other two connections have the same RTT, but variable in the range [100, 400] ms.

Simulations aimed at evaluating TCP performance were done by using the Berkeley Network Simulator [Fall and Vardhan, 1999], version 2. For the characterization of TCP, the New Reno version was used [Floyd and Henderson, 1999], because its implementation of fast recovery and fast retransmit allows to achieve very good performance in wireless networks, if compared to other TCP versions. The TCP connections support a File Transfer Protocol (FTP) application and, in order to model a large file transfer, are represented as always having a packet to send.

In order to simulate a situation of resource sharing, we put a buffer in the router with a FIFO (First In First Out) queue scheme. The maximum queue size was chosen to be 50 packets. In our simulations, data packet size is fixed at 128 byte. The wired link represents the bottleneck of the network and the available bit rate is 384 kbit/s. The packet loss probability P_{Loss} over the wireless links, was assumed to be variable in the range $[10^{-5}, 10^{-2}]$. Simulation parameters are summarized in Table I.

Numerous simulations were run in order to find the values of the P_{Loss} which guarantee the desired fair throughput. The derived values of the packet loss probability were substituted in Eq. (3) so that, combining Eq. (2) and Eq. (1), the searched values for the received power of each connection can be obtained.

3.2 Simulation Results

The performance of the proposed algorithm was evaluated in terms of the average throughput for each of the involved TCP connections as well as of a fairness metric. Our approach was compared with the Standard power controlled technique used in typical cellular and CDMA systems. This Standard scheme aims at limiting the *near-far effect* through an appropriate equalization of the transmission power of all the involved connections. The *near-far effect* is the condition in which, the

more a given sender is close to the destination, the higher will be, at this destination, the perceived level of its signal with respect to the other connections. If the Standard scheme is used, the ratio of the received power for the involved connections is one and the $\frac{E_b}{I_0}$ ratio for a particular connection, evaluated at the Base Station, is:

$$\left(\frac{E_b}{I_0}\right)_i = \frac{W}{R_i} \cdot \frac{1}{M-1} \quad (4)$$

If $P_G=10$ and $M=4$, from Eq. (2) and Eq. (3), a $P_{Loss}=8.3 \cdot 10^{-3}$ can be obtained.

As regards the fairness metric, in this paper we refer to a fairness parameter which has demonstrated to be the most complete among those who have been proposed so far in literature. This metric is called the *Jain fairness index* [Jain et al, 1984]. Considering n flows, with flow i receiving a fraction x_i of the given link bandwidth, the fairness index, using the Jain metric, is defined as:

$$f(x_1, x_2, \dots, x_n) = \frac{(\sum_{i=1}^n x_i)^2}{n \sum_{i=1}^n x_i^2} \quad (5)$$

As can be seen in Eq. (5), the fairness index ranges in the interval $[\frac{1}{n}, 1]$.

Figs. 2-7 show the average throughput and the fairness index values for the three considered and above described scenarios. In particular, these figures aim at comparing the performance which can be achieved by using the Standard and Modified power management approaches.

Fig. 2 shows the throughput for the two considered set of connections employing a Modified approach, with respect to the throughput obtained with a Standard equalized approach. In this figure, the couple of connections having the same RTT (i. e. 100 ms) and thus the same throughput, are plot together. As it can be seen, the Modified approach guarantees that the two throughput curves remain very close to each other in spite of the RTT increase. This is the evidence that the fairness we expected can be achieved. On the other hand, a Standard solution causes an increasing difference in the throughput of the two set of connections as long as the RTT increases. In addition, it can be observed that, within 200 ms, the total average throughput, which can be obtained by summing the two throughput contributions for each of the proposed approaches, is the same. Fig. 3 shows the fairness index for the two Modified and Standard approaches. It is evident that, as expected by observing Fig. 2, the fairness which the proposed Modified solution allows to achieve, is very close to 1. This is the evidence that, our power management solution gives, not only an average throughput which is very close to the one which can be obtained in Standard CDMA systems, but also the desirable fairness which, on the other hand, Standard CDMA systems do not preserve, as evident

Parameter	Value
Number of connections	4
TCP version	New Reno
Packet size	128 bytes
Buffer type	FIFO
Buffer size	50 pkts
Wireless link bandwidth	384 Kb/s
P_{Loss}	$10^{-5} \div 10^{-2}$
Wired link bandwidth	384 Kb/s
Simulation time	200 sec

TABLE I
SIMULATION PARAMETERS

in Fig. 3. When the fixed RTT of two of the four connections is 200 ms or 300 ms, Figs. 4-7 can be derived. In these last cases, the situation is additionally improved. In fact, not only the throughput curves derived for the two power management approaches are very close, but also the fairness index obtained with the Modified technique greatly outperforms the one derived for Standard CDMA solutions.

By simulations, it has been observed that, employing a fair throughput strategy actually increases the performance of the system leading to a balance in terms of throughput and reduction of interference for connections sharing the same resources. Finally, it is worth noting that, the Modified approach outperforms, in terms of fairness, the Standard CDMA approach for all the considered scenarios.

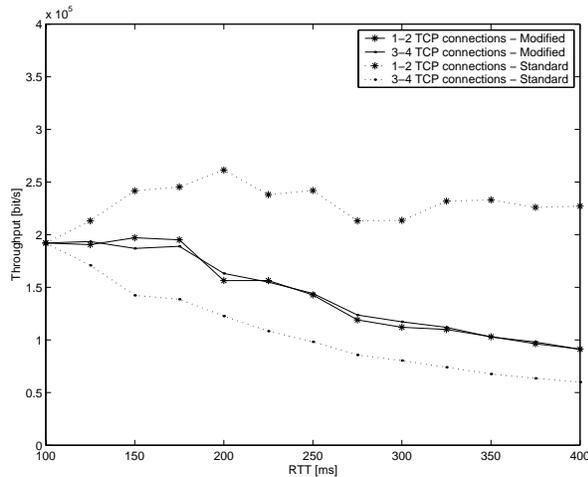


Fig. 2. Throughput comparison for Modified and Standard approach in Case I.

4 CONCLUSIONS

In this paper we have developed a new approach for power management in order to resolve the unfairness problem which arises when many connections with different values of the RTT share the same resources. In particular, by simulations, it has been possible to calculate the required values of transmission power for

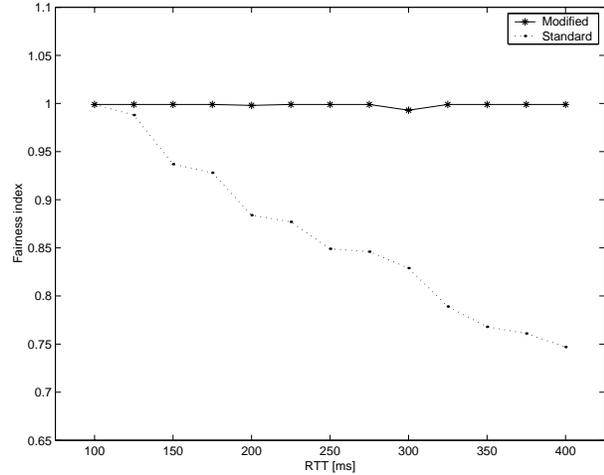


Fig. 3. Fairness index comparison for Modified and Standard approach in Case I.

the involved connections which guarantee the fairness in the exploitation of the available resources. The performance of the proposed power management strategy has been evaluated through simulation and it has been demonstrated that, with respect to a Standard CDMA approach, the introduced approach allows to achieve the desirable fairness as well as a good throughput.

REFERENCES

- Zorzi M. Rossi M. Mazzini G. 2002, "Throughput and energy performance of TCP on a wideband CDMA air interface", *Journal of Wireless Communications and Mobile Computing (WCMC)*, vol. 2, no. 1, pp. 71-84.
- Lakshman T. V. and Madhow U. 1997, "The performance of TCP/IP for networks with high bandwidth-delay products and random loss", *IEEE/ACM Transactions on Networking*, vol. 5 no. 3, pp. 336-350.
- Henderson T. R. Sahouria E. McCanne S. and Katz R. H. 1998, "On improving the fairness of TCP congestion avoidance", *In Proc. of IEEE GLOBECOM '98*.
- Pilosof S. Ramjee R. Raz D. Shavitt Y. and Sinha P. 2003, "Understanding TCP fairness over wireless LAN", *In Proc. of IEEE INFOCOM 2003*.
- Wu Q. 1999, "Performance of optimum transmitter

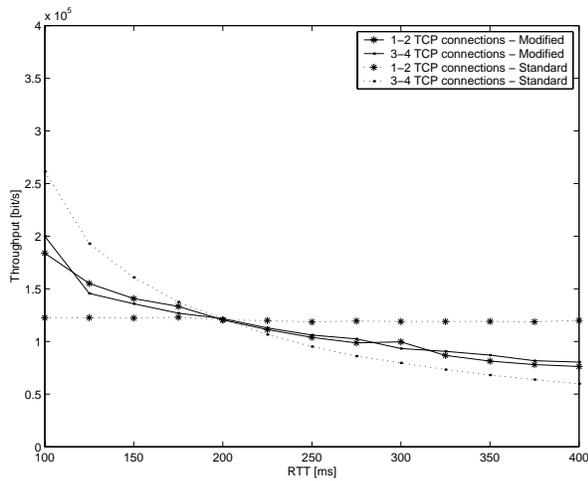


Fig. 4. Throughput comparison for Modified and Standard approach in Case II.

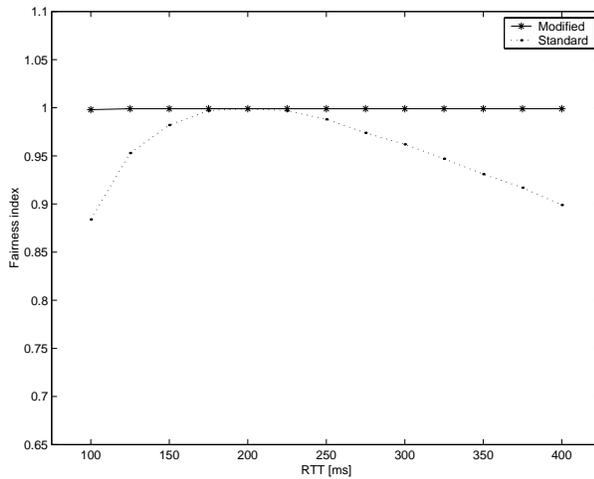


Fig. 5. Fairness index comparison for Modified and Standard approach in Case II.

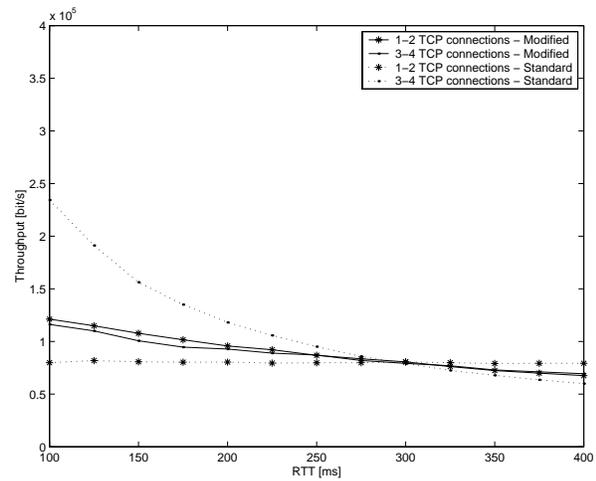


Fig. 6. Throughput comparison for Modified and Standard approach in Case III.

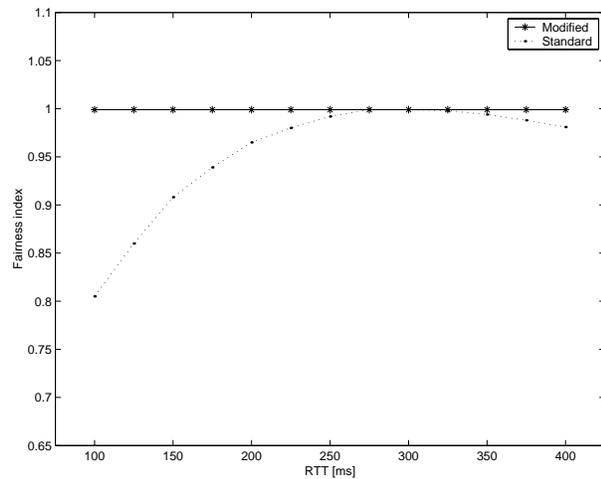


Fig. 7. Fairness index comparison for Modified and Standard approach in Case III.

power control in CDMA cellular mobile systems”, *IEEE Transactions on vehicular technology*, vol. 48, no. 2, pp. 571-575.

Grandhi S. A. Vijayan R. and Goodman D. J. 1994, ”Distributed power control in cellular radio systems”, *IEEE Transactions on Communications*, vol. 42, no. 2-3-4, pp.226-228.

Lee T. H. and Lin J. C. 1996, ”A fully distributed power control algorithm for cellular mobile systems”, *JSAC*, vol. 14, no. 4, pp.692-697.

Ojampera T. and Prasad R. 1998, ”An overview of air interface multiple access for IMT-2000/UMTS”, *IEEE Communication Magazine*, vol. 36, no. 9, pp. 82-95.

Proakis J. G. 1995, ”Digital communication”, Mc Graw-Hill, 3rd Edition.

Fall K. and Vardhan K. 1999, ”NS note and documentation”, the source code available at <http://www.mash.cs.berkeley.edu/ns/>.

Floyd S. and Henderson T. 1999, ”The NewReno

modification to TCP’s Fast Recovery Algorithm”, Tech. Rep., RFC 2582.

Jain R. Chiu D. and Hawe W. 1984, ”A quantitative measure of fairness and discrimination for resource allocation in shared computer systems”, DEC Research Report TR-301.



Laura Galluccio received her laurea degree in Electrical Engineering from University of Catania, Catania, Italy in 2001. She is Ph.D student with the Dipartimento di Ingegneria Informatica e delle Telecomunicazioni, University of Catania and also she currently is a CNIT Researcher in the VICOM project. In 2002 she won the International Prize Marisa Bellisario. Her research interests include Satellite and Wireless Networks, Ad Hoc Networks and

Network Performance Analysis.

TRANSIENT ANALYSIS OF A MARKOV MODULATED FLUID QUEUE WITH LINEAR SERVICE RATE

L. RABEHASAINA, B. SERICOLA

IRISA-INRIA

Campus universitaire de Beaulieu

35042 Rennes cedex, France

Email : {landy, sericola}@irisa.fr,

Abstract: We consider an infinite capacity fluid queue governed by a continuous time Markov chain and with linear service rate. The transient behavior of this fluid flow model is described by a linear differential equation. We study the transient distribution of the fluid level in the queue and we derive a partial differential equation satisfied by the cumulative distribution function of the fluid level. Using this partial differential equation, we obtain a simple expression of the moments of this transient distribution, as well as its Laplace transform.

Keywords: Stochastic Models, Fluid Queue, Markov Models, Queueing Systems.

1 INTRODUCTION

We consider in this paper an infinite capacity fluid queue of which level at time t is denoted by $Q(t)$. Fluid arrives in this queue according to a non decreasing process $A(t)$ and leaves the buffer at a rate $\tau(X(t), Q(t))$, where $X(t)$ is a continuous time Markov chain and τ is a non negative function. This is a generalization of standard fluid queues driven by a superposition of On-Off sources (see [Anick et al, 1982]), since here the service rate depends on the queue level, in addition to the state of the underlying Markov chain.

The fluid level in the queue $Q(t)$ then satisfies the following differential equation reflected at 0

$$dQ(t) = dA(t) - \tau(X(t), Q(t))dt + dL_t$$

where L_t is a non decreasing process, (called the *regulator*), interfering only when $Q(t) = 0$ and preventing it from being negative.

We consider in this paper the *linear* model which corresponds to the case where $\tau(X(t), Q(t))$ is linear in $Q(t)$, that is $\tau(X(t), Q(t)) = \mu(X(t))Q(t)$, where μ is a positive function which depends only on the Markov chain X . This model has been studied in [Asmussen and Kella, 1996], [Kella and Stadje, 2002] and [Kella and Whitt, 1999], where $A(t)$ is a Lévy process or a Markov modulated Lévy process. These papers mainly focus on the limiting distribution of

$Q(t)$. The authors identify a functional equation satisfied by the Laplace-Stieltjes transform of the limiting distribution, which can be used to evaluate the two first moments of the buffer level distribution.

We consider in this paper the case where $dA(t) = \lambda(X(t))dt$, where λ is a non negative function of the Markov chain. The evolution of $Q(t)$ is then described by the following equation.

$$dQ(t) = \lambda(X(t))dt - \mu(X(t))Q(t)dt, \quad (1)$$

the term L_t having disappeared because 0 is an impenetrable barrier for $Q(t)$ (see [Asmussen and Kella, 1996]. Note that $(X(t), Q(t))$ is then a Markov process. In the literature, the differential equations governing this process are generally obtained from backward analysis. In this paper we use a forward argument to obtain them, this yields an easier study of the moments of $Q(t)$.

Throughout this paper, X denotes a stationary ergodic Markov chain evolving on a finite state space $S = \{1, \dots, N\}$. We denote by $A = (a_{i,j})_{(i,j) \in S \times S}$ its infinitesimal generator and by $\pi = (\pi_1, \dots, \pi_N)$ its stationary distribution. We also suppose that X is a two sided process, i.e. indexed by \mathbb{R} .

The paper is organized as follows. In Section 2, we give an explicit expression of the fluid level $Q(t)$ and we describe the jumps of its distribution. In Section 3, we derive a partial differential equation satisfied by the cumulative distribution function of $Q(t)$

and we obtain in Section 4 an expression of the moments of $Q(t)$, as well as its Laplace transform.

2 PRELIMINARIES

Let us denote by $Q^y(t)$ the fluid level in the queue at time t with the initial condition $Q^y(0) = y$. For $t \geq 0$, $Q^y(t)$ satisfies the equation (1) and it can be easily checked that $Q^y(t)$ is given by

$$Q^y(t) = y \exp\left(-\int_0^t \mu(X(s)) ds\right) + \int_0^t \exp\left(-\int_s^t \mu(X(v)) dv\right) \lambda(X(s)) ds.$$

We also have the following relation between $Q^y(t)$ and $Q^y(t')$ for $t \geq t' \geq 0$,

$$Q^y(t) = Q^y(t') \exp\left(-\int_{t'}^t \mu(X(s)) ds\right) + \int_{t'}^t \exp\left(-\int_s^t \mu(X(v)) dv\right) \lambda(X(s)) ds.$$

Using the stationarity of X , we have that $Q^y(t)$ and $\tilde{Q}^y(t)$ have the same distribution, where $\tilde{Q}^y(t)$ is given by

$$\tilde{Q}^y(t) = y \exp\left(-\int_{-t}^0 \mu(X(s)) ds\right) + \int_{-t}^0 \exp\left(-\int_s^0 \mu(X(v)) dv\right) \lambda(X(s)) ds. \quad (2)$$

Let us denote by X^* the reversed process of X defined by $X^*(s) = X(-s)$. It is standard that X^* is a continuous time Markov chain with infinitesimal generator $A^* = \Pi^{-1} A^T \Pi$, where T denotes the transpose operator and Π is the diagonal matrix containing the vector π , which is also the stationary distribution of X^* . The variable changes $s := -s$ and $v := -v$ in Relation (2) leads to the following expression for $\tilde{Q}^y(t)$

$$\tilde{Q}^y(t) = y \exp\left(-\int_0^t \mu(X^*(s)) ds\right) + \int_0^t \exp\left(-\int_0^s \mu(X^*(v)) dv\right) \lambda(X^*(s)) ds. \quad (3)$$

Because the initial buffer level y is fixed throughout this paper, and for readability purpose, we simply use the notation $Q(t)$ instead of $Q^y(t)$.

It is easy to check that for a fixed $t > 0$, the distribution of $Q(t)$ has jumps which correspond to the fact that the Markov chain X^* stays during the whole interval $[0, t]$ in a subset of states having the same values for $\lambda(i)$ and $\mu(i)$. More precisely, let

m be the number of distinct pairs $(\lambda(i), \mu(i))$ for $i \in S$. If we denote these m different pairs by $(u(1), v(1)), \dots, (u(m), v(m))$, we obtain the partition B_1, \dots, B_m of the state space S by defining B_l as

$$B_l = \{i \in S \mid (\lambda(i), \mu(i)) = (u(l), v(l))\}.$$

For $l = 1, \dots, m$ and $t > 0$, we denote by $s_l(t)$ the quantities

$$s_l(t) = ye^{-v(l)t} + \frac{u(l)(1 - e^{-v(l)t})}{v(l)}.$$

We then have from Relation (3)

$$Q(t) = s_l(t) \iff X^*(s) \in B_l, \forall s \in [0, t]$$

It follows that

$$\Pr\{Q(t) = s_l(t)\} = \pi_{B_l} e^{A_{B_l B_l} t} \mathbb{1},$$

where $A_{B_l B_l}$ is the sub-infinitesimal generator of dimension $|B_l|$ obtained from A by considering only the internal transitions of the subset B_l and π_{B_l} is the subvector of dimension $|B_l|$ obtained from vector π by considering the stationary probabilities of the subset B_l . The vector $\mathbb{1}$ is the column vector with all its entries equal 1, its dimension being given by the context.

3 DISTRIBUTION OF THE FLUID LEVEL IN THE QUEUE

We denote by $F_i(t, x)$ the cumulative distribution function of $Q(t)$ given that $X^*(0) = i$, that is, $F_i(t, x) = \Pr\{Q(t) \leq x \mid X^*(0) = i\}$. We denote by $F(t, x)$ the column vector $(F_i(t, x))_{i \in S}$, by D the diagonal matrix containing the $\mu(i)$'s and by Λ the diagonal matrix containing the $\lambda(i)$'s.

The distribution $F(t, x)$ of $Q(t)$ verifies the following differential equation.

Theorem 3.1 *For every (t, x) such that $ye^{-\mu(i)t} + \lambda(i)(1 - e^{-\mu(i)t})/\mu(i) \neq x$ for all $i \in S$ we have*

$$\partial_t F(t, x) = A^* F(t, x) + (Dx - \Lambda) \partial_x F(t, x). \quad (4)$$

Proof. Let us denote by P^* the transition probability matrix of the uniformized discrete-time Markov chain associated with X^* . We then have $P^* = I + A^*/\nu$, where I is the identity matrix and ν is the uniformization rate satisfying $\nu \geq \max\{-a_{i,i}^*, i \in S\}$. Let T_1 be the first instant of jump of X^* . T_1 is linked to the uniformized Markov chain via the equality

$$d \Pr\{X^*(T_1) = j, T_1 = u \mid X^*(0) = i\} = p_{i,j}^* \nu e^{-\nu u} du.$$

Defining $G_{i,j}(t, u, x) = \Pr\{Q(t) \leq x \mid X^*(T_1) = j, T_1 = u, X^*(0) = i\}$, we obtain

$$\begin{aligned} F_i(t, x) &= \sum_{j \in S} \int_0^\infty G_{i,j}(t, u, x) \\ &\quad d\Pr\{X^*(T_1) = j, T_1 = u \mid X^*(0) = i\} \\ &= \nu \sum_{j \in S} p_{i,j}^* \int_0^\infty G_{i,j}(t, u, x) e^{-\nu u} du. \end{aligned} \quad (5)$$

Let us define for all $t \geq 0$

$$I_1(i, t, x) = \nu \sum_{j \in S} p_{i,j}^* \int_0^t G_{i,j}(t, u, x) e^{-\nu u} du$$

$$\text{and } I_2(i, t, x) = \nu \sum_{j \in S} p_{i,j}^* \int_t^\infty G_{i,j}(t, u, x) e^{-\nu u} du.$$

We first consider $I_1(i, t, x)$, where we integrate $G_{i,j}(t, u, x) e^{-\nu u}$ for $u \in [0, t]$. Then, when u lies in that interval and $T_1 = u$ and $X^*(0) = i$, we have

$$\begin{aligned} Q(t) &= y \exp\left(-\int_0^t \mu(X^*(s)) ds\right) \\ &\quad + \int_0^t \exp\left(-\int_0^s \mu(X^*(v)) dv\right) \lambda(X^*(s)) ds \\ &= y \exp\left(-\int_0^u \mu(X^*(s)) ds - \int_u^t \mu(X^*(s)) ds\right) \\ &\quad + \int_0^u \exp\left(-\int_0^s \mu(X^*(v)) dv\right) \lambda(X^*(s)) ds \\ &\quad + \int_u^t \exp\left(-\int_0^s \mu(X^*(v)) dv\right) \lambda(X^*(s)) ds \\ &= ye^{-\mu(i)u} \exp\left(-\int_u^t \mu(X^*(s)) ds\right) \\ &\quad + \frac{\lambda(i)(1 - e^{-\mu(i)u})}{\mu(i)} \\ &\quad + \int_u^t \exp\left(-\int_0^s \mu(X^*(v)) dv\right) \lambda(X^*(s)) ds \\ &= ye^{-\mu(i)u} \exp\left(-\int_u^t \mu(X^*(s)) ds\right) \\ &\quad + \frac{\lambda(i)(1 - e^{-\mu(i)u})}{\mu(i)} \\ &\quad + e^{-\mu(i)u} \int_u^t \exp\left(-\int_u^s \mu(X^*(v)) dv\right) \\ &\quad \quad \lambda(X^*(s)) ds \\ &= \frac{\lambda(i)(1 - e^{-\mu(i)u})}{\mu(i)} + e^{-\mu(i)u} Q_u(t), \end{aligned}$$

where

$$\begin{aligned} Q_u(t) &= y \exp\left(-\int_u^t \mu(X^*(s)) ds\right) \\ &\quad + \int_u^t \exp\left(-\int_u^s \mu(X^*(v)) dv\right) \lambda(X^*(s)) ds. \end{aligned}$$

Hence for $u \in [0, t]$, we have

$$\begin{aligned} G_{i,j}(t, u, x) &= \Pr\{Q(t) \leq x \mid X^*(T_1) = j, T_1 = u, X^*(0) = i\} \\ &= \Pr\left\{\frac{\lambda(i)(1 - e^{-\mu(i)u})}{\mu(i)} + e^{-\mu(i)u} Q_u(t) \leq x \mid X^*(T_1) = j, T_1 = u, X^*(0) = i\right\}. \end{aligned}$$

Now from the Markov property and the homogeneity of X^* we get that the distribution of $Q_u(t)$ given $X^*(u)$ is the same as the distribution of $Q(t - u)$ given $X^*(0)$, thus

$$\begin{aligned} G_{i,j}(t, u, x) &= \Pr\left\{\frac{\lambda(i)(1 - e^{-\mu(i)u})}{\mu(i)} + e^{-\mu(i)u} Q(t - u) \leq x \mid X^*(0) = j\right\} \\ &= \Pr\{Q(t - u) \leq e^{\mu(i)u} \left(x - \frac{\lambda(i)(1 - e^{-\mu(i)u})}{\mu(i)}\right) \mid X^*(0) = j\} \\ &= F_j(t - u, xe^{\mu(i)u} + \lambda(i)(1 - e^{\mu(i)u})/\mu(i)). \end{aligned} \quad (6)$$

Let us now consider $I_2(i, t, x)$. For $u \geq t$, the expression of $G_{i,j}(t, u, x)$ is given by

$$\begin{aligned} G_{i,j}(t, u, x) &= \Pr\{ye^{-\mu(i)t} + \lambda(i)(1 - e^{-\mu(i)t})/\mu(i) \leq x \mid X^*(T_1) = j, T_1 = u, X^*(0) = i\} \\ &= \mathbf{1}_{\{ye^{-\mu(i)t} + \lambda(i)(1 - e^{-\mu(i)t})/\mu(i) \leq x\}}. \end{aligned} \quad (7)$$

We denote this indicator function by $\eta(i, t, x)$. $\eta(i, t, x)$ is differentiable in t and x for all (t, x) in the domain $\{(t, x) \mid ye^{-\mu(i)t} + \lambda(i)(1 - e^{-\mu(i)t})/\mu(i) \neq x\}$. Since $\eta(i, t, x)$ is a constant equal to 0 or 1 in this domain, its derivatives in t and x are both equal to 0.

Hence from (6) and (7) we have

$$\begin{aligned} I_1(i, t, x) &= \nu \sum_{j \in S} p_{i,j}^* \int_0^t F_j(t - u, xe^{\mu(i)u}) \\ &\quad + \lambda(i)(1 - e^{\mu(i)u})/\mu(i) e^{-\nu u} du \\ &= \nu e^{-\nu t} \sum_{j \in S} p_{i,j}^* \int_0^t F_j(u, xe^{\mu(i)(t-u)}) \\ &\quad + \lambda(i)(1 - e^{\mu(i)(t-u)})/\mu(i) e^{\nu u} du, \\ I_2(i, t, x) &= \sum_{j \in S} p_{i,j}^* \eta(i, t, x) e^{-\nu t} \\ &= \eta(i, t, x) e^{-\nu t} \end{aligned}$$

where we made the change variable $u := t - u$ in the expression of $I_1(i, t, x)$.

Let us now derive (5) with respect to t . We have $\partial_t F_i(t, x) = \partial_t I_1(i, t, x) + \partial_t I_2(i, t, x)$, and direct calculation yields

$$\begin{aligned} \partial_t I_1(i, t, x) &= -\nu I_1(i, t, x) + \nu \sum_{j \in S} p_{i,j}^* F_j(t, x) \\ &+ \nu e^{-\nu t} (\mu(i)x - \lambda(i)) \sum_{j \in S} p_{i,j}^* \int_0^t \partial_x F_j(u, \\ &xe^{\mu(i)(t-u)} + \lambda(i)(1 - e^{\mu(i)(t-u)})/\mu(i)) e^{\nu u} du \\ &= -\nu I_1(i, t, x) + \nu \sum_{j \in S} p_{i,j}^* F_j(t, x) \\ &+ (\mu(i)x - \lambda(i)) \partial_x \left[\nu e^{-\nu t} \sum_{j \in S} p_{i,j}^* \int_0^t F_j(u, \right. \\ &xe^{\mu(i)(t-u)} + \lambda(i)(1 - e^{\mu(i)(t-u)})/\mu(i)) e^{\nu u} du \left. \right] \\ &= -\nu I_1(i, t, x) + \nu \sum_{j \in S} p_{i,j}^* F_j(t, x) \\ &+ (\mu(i)x - \lambda(i)) \partial_x I_1(i, t, x). \end{aligned}$$

and

$$\begin{aligned} \partial_t I_2(i, t, x) &= -\nu \eta(i, t, x) e^{-\nu t} \\ &= -\nu I_2(i, t, x). \end{aligned}$$

By adding these terms, we get

$$\begin{aligned} \partial_t F_i(t, x) &= -\nu F_i(t, x) \\ &+ \nu \sum_{j \in S} p_{i,j}^* F_j(t, x) + (\mu(i)x - \lambda(i)) \partial_x I_1(i, t, x). \end{aligned}$$

But since $\partial_x I_2(i, t, x) = 0$, we obtain

$$\begin{aligned} \partial_t F_i(t, x) &= -\nu F_i(t, x) \\ &+ \nu \sum_{j \in S} p_{i,j}^* F_j(t, x) + (\mu(i)x - \lambda(i)) \partial_x F_i(t, x), \end{aligned}$$

and the results follow by using the relation $P^* = I + A^*/\nu$. \blacksquare

3.1 Moments evaluation

We consider in this section the moments of the transient buffer level $Q(t)$. Let us note that, since the jumps of the cumulative distribution function of $Q(t)$ are known, the equation (4) has a unique solution provided that the initial conditions are fixed. However, we succeed in finding an expression of the moments of $Q(t)$ and an expression of its Laplace transform, without solving this equation.

We first recall the following well-known result

Lemma 3.2 *Let H be the cumulative distribution function of a non negative random variable. For every $r \geq 1$, if the r -th order moment exists, we have*

$$\int_0^\infty x^r dH(x) = r \int_0^\infty x^{r-1} (1 - H(x)) dx.$$

Proof. See for instance [Feller, 1957]. \blacksquare

Let us denote by $v_i(t, k)$ the k th moment of $Q(t)$ given that the initial state of the Markov chain X^* is i , that is

$$v_i(t, k) = E(Q(t)^k \mid X^*(0) = i).$$

We denote by $V(t, k)$ the column vector containing the $v_i(t, k)$. By definition, we have $V(t, 0) = \mathbb{1}$. In the following corollary of Theorem 3.1, we give an expression for all the moments of the buffer level $Q(t)$.

Corollary 3.3 *For every $k \geq 1$, we have the following recursion for the process $\{V(t, k), t \geq 0\}$*

$$\begin{aligned} V(t, k) &= e^{(A^* - kD)t} y^k \mathbb{1} \\ &+ e^{(A^* - kD)t} \int_0^t e^{-(A^* - kD)s} k \Lambda V(s, k-1) ds \end{aligned} \quad (8)$$

Proof. Since $A^* \mathbb{1} = 0$, relation (4) can be written as

$$\begin{aligned} -\partial_t (\mathbb{1} - F(t, x)) &= -A^* (\mathbb{1} - F(t, x)) \\ &+ (Dx - \Lambda) \partial_x F(t, x). \end{aligned}$$

Multiplying both sides by x^{k-1} , for $k \geq 1$, and after integration, we get

$$\begin{aligned} -\partial_t \int_0^\infty x^{k-1} (\mathbb{1} - F(t, x)) dx & \\ &= -A^* \int_0^\infty x^{k-1} (\mathbb{1} - F(t, x)) dx \\ &+ D \int_0^\infty x^k \partial_x F(t, x) dx \\ &- \Lambda \int_0^\infty x^{k-1} \partial_x F(t, x) dx. \end{aligned}$$

Using Lemma 3.2, we easily get

$$V'(t, k) = (A^* - kD)V(t, k) + k\Lambda V(t, k-1).$$

It is easily checked that the solution to this equation is (8), which completes the proof. \blacksquare

The k th moment of $Q(t)$ is then easily given by $E(Q(t)^k) = \pi V(t, k)$. Note that in the case $k = 1$ simple computation yields the following expression for $E(Q(t))$:

$$E(Q(t)) = \gamma \pi e^{(A^* - D)t} \mathbb{1} + \pi (A^* - D)^{-1} [e^{(A^* - D)t} - I] \Lambda \mathbb{1}.$$

It is easy to verify that in the case $\mu(i) = \mu$ (i.e. $D = \mu I$) we have

$$E(Q(t)) = ye^{-\mu t} - \frac{1}{\mu}[e^{-\mu t} - 1]\pi\Lambda\mathbb{1}.$$

We now easily deduce the Laplace transform $\phi(t, \theta) = E(\exp(\theta Q(t)))$ of $Q(t)$ for all t :

Corollary 3.4 For every $\theta \in \mathbb{R}$, we have

$$\phi(t, \theta) = \pi \sum_{k=0}^{\infty} \frac{V(t, k)\theta^k}{k!}.$$

Proof. First note that it is easy to see that $Q(t)$ is upper bounded by the deterministic value $c(t, y) = y + t \sup_{i \in S} \lambda(i)$. Besides we have $\partial_{\theta}^k \phi(t, 0) = E(Q(t)^k) = \pi V(t, k)$. Hence we have for all $N \geq 0$

$$\begin{aligned} & \left| \phi(t, \theta) - \sum_{k=0}^N \frac{\partial_{\theta}^k \phi(t, 0)\theta^k}{k!} \right| \\ &= \left| E \left(\exp(\theta Q(t)) - \sum_{k=0}^N \frac{Q(t)^k \theta^k}{k!} \right) \right| \\ &= \left| E \left(\sum_{k=N+1}^{\infty} \frac{Q(t)^k \theta^k}{k!} \right) \right| \\ &\leq \sum_{k=N+1}^{\infty} c(y, t)^k \frac{\theta^k}{k!} \\ &\rightarrow 0 \quad \text{as } N \rightarrow \infty. \end{aligned}$$

This completes the proof. ■

REFERENCES

Anick D., Mitra D. and Sondhi M. M. 1982, “Stochastic theory of a data-handling system with multiple sources” *The Bell System Tech. Journal*, Vol. 61. Pp1871-1894.

Asmussen S. and Kella O. 1996 “Rate modulation in dams and ruin problem” *Journal of Applied Probability*, Vol. 33(2). Pp523-535.

Feller W. 1957 *An introduction to probability theory and its applications. Vol. 1.* Wiley series in probability and mathematical statistics.

Kella O. and Stajje W. 2002 “Markov modulated linear fluid networks with Markov additive input” *Journal of Applied Probability*, Vol. 39(2). Pp413-420.

Kella O. and Whitt W. 1999 “Linear Stochastic Fluid Networks” *Journal of Applied Probability*, Vol. 36(1). Pp244-260.

BIOGRAPHY



LANDY RABEHASAINA is a PhD student in Applied Mathematics at the University of Rennes I. His current work deals with stochastic processes applied to fluid queues and stochastic networks.



BRUNO SERICOLA received the Ph.D. degree in computer science from the University of Rennes I in 1988. He has been with INRIA (Institut National de Recherche en Informatique et Automatique, a public research French laboratory) since 1989. His main research activity is in computer and communication systems performance evaluation, dependability and performability analysis of fault-tolerant architectures and applied stochastic processes.

STATIONARY ANALYSIS OF TANDEM FLUID QUEUES FED BY HOMOGENEOUS ON-OFF SOURCES

N. BARBOT and B. SERICOLA

IRISA-INRIA

Campus universitaire de Beaulieu

35042 Rennes cedex, France

Email : {Nelly.Barbot, Bruno.Sericola}@irisa.fr,

Abstract: We consider a fluid system composed of multiple buffers in series. The first buffer receives fluid from a finite superposition of independent identical on-off sources. The active and silent periods of sources are exponentially distributed. The i th buffer releases fluid in the $(i + 1)$ th buffer. Assuming that the input rate of one source is greater than the service rate of the first buffer, the output process of each buffer can be modeled by an on-off source with the active period distributed as the busy period of an M/M/1 queue. For $i \geq 2$, the stationary content distribution of the i th buffer is obtained by the use of generating functions which are explicitly inverted.

Keywords: Tandem fluid queues, output process, generating functions.

1 INTRODUCTION

We consider tandem fluid queues fed by a finite number of identical on-off sources. It is assumed that silent and active periods of the sources are independent and exponentially distributed. Tandem fluid queues are composed of consecutive infinite capacity buffers. The stationary behavior of the first buffer is explicitly derived in [Anick et al., 1982], using spectral decomposition arguments. As far as the other buffers are concerned, the output processes need to be characterized. In [Aalto, 1998] and [Boxma and Dumas, 1998], the authors consider a fluid queue driven by a superposition of on-off sources, with exponentially distributed silent periods and generally distributed active periods. Assuming that the input rate of one source is greater than the constant service rate of the buffer, they prove that the output process behaves as an on-off source with exponentially distributed silent periods and active periods distributed like the busy periods of a M/G/1 queue. In this paper, we consider the stationary behavior of each buffer level in the tandem fluid queues, apart from the first one. Using results of [Aalto, 1998] and [Boxma and Dumas, 1998], the output processes look like on-off sources with active periods distributed as busy periods of an M/M/1

queue. This tandem of fluid queues has been studied in [Aalto, 1998], where the output processes have been considered as alternating renewal processes. The authors obtained the stationary fluid level distribution of each buffer in terms of a Bessel function integral. Here, we derive a new analytic expression of these distributions. By using the method developed in [Leguesdron et al, 1991] and [Barbot and Sericola, 2002], we write the solutions in terms of a matrix exponential and then via generating functions that are explicitly inverted. Nevertheless, as shown in the next section, we deal here with a more general setting than the one of [Barbot and Sericola, 2002].

2 MODEL FORMULATION

We consider M infinite capacity fluid queues in series. The first one is fed by the superposition of N independent identical on-off sources with exponentially distributed on-off periods with parameters μ and λ respectively. During the on period, a source emits fluid at a constant rate c_0 . The first buffer empties in the second one at the rate c_1 . For $i \geq 2$, the input of the i th buffer is the output from the buffer $i - 1$ and its service rate is denoted by c_i . It is assumed that $Nc_0 > c_1 > \dots > c_M > 0$ in order

to avoid the trivial case where one or more buffers remain empty. Moreover, we make the restrictive assumption $c_0 \geq c_1$ which permits the output process of the first buffer to be simply derived.

Definition 1 *An on-off source is called an MM1(β, a, b, r) source if the off periods are exponentially distributed with rate β and the on periods are distributed as the busy periods of an M/M/1 queue with arrival rate a and service rate b . During the on periods, the source emits fluid at rate r .*

The infinitesimal generator associated with such a source is denoted by A . Its non-zero entries are

$$A_{0,0} = -\beta, A_{0,1} = \beta, A_{j,j-1} = b,$$

$$A_{j,j} = -(a+b) \text{ and } A_{j,j+1} = a \text{ for } j \geq 1. \quad (1)$$

Note that in [Barbot and Sericola, 2002], we considered a single fluid queue fed by a classical M/M/1 queue, which is, our definition, a fluid queue fed by an MM1(a, a, b, r) source. Here we have to deal with MM1(β, a, b, r) sources, where $\beta \neq a$, which generalizes the results of [Barbot and Sericola, 2002].

The following lemmas are proved in [Aalto, 1998] and [Boxma and Dumas, 1998].

Lemma 2 *In the stationary regime, the output process of the first buffer is equivalent to an MM1($N\lambda, \lambda_1, \mu_1, c_1$) source where $\lambda_1 = \lambda(N - c_1/c_0)$ and $\mu_1 = \mu c_1/c_0$.*

Lemma 3 *In the stationary regime, the output process of a buffer with service rate c and fed by an MM1(β, a, b, r) source is equivalent to an MM1(β, a', b', c) source where $a' = ac/r + \beta(1 - c/r)$ and $b' = bc/r$.*

Using Lemmas 2 and 3, the output process of the i th buffer, for $1 \leq i \leq M$, is equivalent to an MM1($N\lambda, \lambda_i, \mu_i, c_i$) source where $\lambda_i = \lambda(N - c_i/c_0)$ and $\mu_i = \mu c_i/c_0$. Therefore, the traffic intensity in the i th buffer is given by $\rho_i = c_0 N \lambda / (c_i(\lambda + \mu))$ and the stability condition of the tandem fluid queues is $\rho_M < 1$.

3 A BUFFER FED BY AN MM1(β, a, b, r) SOURCE

We consider a single fluid buffer fed by an MM1(β, a, b, r) source. The service rate of the buffer is denoted by c , $c < r$. We derive an expression of the stationary buffer content distribution in terms of a series whose coefficients correspond to the successive powers of a *key matrix* G . The

generating function of G is expressed as a function of the known generating function of a *key matrix* T and is explicitly inverted.

The continuous time birth and death process associated with the MM1(β, a, b, r) source is denoted by $\{X_t, t \geq 0\}$ and its infinitesimal generator A is described by (1). We assume that $a \leq \beta$.

The drifts of that fluid queue represent the difference between the input and the service rates. Let d_j be the drift when X_t is in the state j . We thus have $d_0 = -c$ and $d_j = r - c$, for every $j \geq 1$. The diagonal matrix containing these drifts is denoted by D . Since we are concerned by the stationary behavior of that fluid queue, we suppose that $a < b$ and that the stability condition is satisfied. Since the mean duration of on periods is $1/(b - a)$, we have

$$\rho_0 = \frac{r\beta}{c(b - a + \beta)} < 1.$$

The stationary state of the Markov chain $\{X_t, t \geq 0\}$ and the stationary amount of fluid in the buffer are denoted X and Q respectively.

Let $F_j(x) = \Pr\{X = j, Q \leq x\}$. It is easy to see that for $j \geq 1$, we have $F_j(0) = 0$ and it has been shown in [Sericola and Tuffin, 1999] that $F_0(0) = 1 - \rho_0$. It is well-known, see e.g. [Mittra, 1988], that the functions F_j satisfy, for $x > 0$, the following system of differential equations $F'(x) = F(x)AD^{-1}$ where $F(x)$ denotes the infinite row vector containing the $F_j(x)$ and $F'(x)$ the derivative of $F(x)$ with respect to x . Its solution is given by $F(x) = F(0)e^{AD^{-1}x}$. Using a method similar to the uniformization technique, we introduce the *key matrix* G defined by $G = I + AD^{-1}/\theta$, where $\theta = (a + b)/(r - c)$ and I is the identity matrix. We then have, for every $j \geq 0$,

$$F_j(x) = (1 - \rho_0) \sum_{n=0}^{\infty} e^{-\theta x} \frac{(\theta x)^n}{n!} G_{0,j}^n, \quad (2)$$

where $G_{0,j}^n$ denotes the $(0, j)$ entry of matrix G^n . In what follows, we focus on the calculation of $G_{0,j}^n$ using generating functions.

3.1 Generating Functions

Let us consider the complex matrices M indexed on $\mathbb{N} \times \mathbb{N}$. We define

$$\nu(M) = \sup_{i \in \mathbb{N}} \sum_{j=0}^{\infty} |M_{ij}|$$

and denote by \mathcal{M} the set of infinite complex matrices M such that $\nu(M)$ is finite. ν is a norm on

\mathcal{M} and (\mathcal{M}, ν) is a Banach algebra. With each $M \in \mathcal{M}$, we associate the complex function Φ_M , called potential kernel of M or generating function, defined by

$$\Phi_M(z) = \sum_{k=0}^{\infty} M^k z^k$$

for every z such that $|z| < 1/\nu(M)$. Note that for $M \in \mathcal{M}$ and z such that $|z| < 1/\nu(M)$, we have $\Phi_M(z) \in \mathcal{M}$ since $\nu(\Phi_M(z)) \leq 1/(1 - |z|\nu(M)) < +\infty$.

The following lemma is a classical straightforward result, so we give it without proof.

Lemma 4 *For every matrix H , $H\Phi_M$ is the only solution to the matrix equation*

$$X(z) = H + zX(z)M$$

for every z such that $|z| < 1/\nu(M)$.

We shall also need the following result, due to [Leguesdron et al., 1991], which will be used along with Lemma 4.

Lemma 5 *For every M and N in \mathcal{M} , we have $\Phi_{M+N}(z) = \Phi_M(z) + z\Phi_{M+N}(z)N\Phi_M(z)$ for every z such that $|z| < \min\{1/\nu(M), 1/\nu(M+N)\}$.*

Let us now introduce some notations. We define the infinite matrices V , W , R and S as

$$V_{i,j} = I_{i+1,j}, \quad W_{i,j} = I_{i,j+1}, \quad R_{i,j} = I_{i,0}I_{0,j}$$

and $S_{i,j} = I_{i,0}I_{1,j}$ for i and $j \in \mathbb{N}$. We studied in [Barbot and Sericola, 2002] the *key matrix* T associated to a fluid buffer fed by an M/M/1 queue with arrival rate a and service rate b . The input and service rates of the buffer are respectively r and c . Therefore, the non-zero entries of T are given by

$$T_{0,0} = q + pr/c, \quad T_{0,1} = p, \quad T_{1,0} = q - qr/c$$

$$T_{1,2} = p \quad \text{and for } i \geq 2, \quad T_{i,i-1} = q, \quad T_{i,i+1} = p$$

where p and q are defined by

$$p = a/(a+b) \quad \text{and} \quad q = b/(a+b).$$

Notice that the stability condition of the fluid model associated with T is satisfied, that is $\rho = ra/cb < 1$.

After some algebra, we easily obtain the following relation between matrices G and T .

Lemma 6 *We have $G = T + U$ where $U = (p_0 - p)((r/c - 1)R + S)$ and $p_0 = \beta/(a+b)$.*

Since $\beta \geq a$, we have $p_0 \geq p$ and so $\nu(G) \geq \nu(T)$. Using Lemma 5, we obtain

$$\Phi_G(z) = \Phi_T(z) + z\Phi_G(z)U\Phi_T(z) \quad (3)$$

for every z such that $|z| < 1/\nu(G)$. We define the matrix $L(z)$ as

$$L(z) = U\Phi_T(z).$$

For $|z| < 1/\nu(T)$, we have $\nu(L(z)) = \nu(U\Phi_T(z)) \leq \nu(U)/(1 - |z|\nu(T))$, and so for every z such as $|z| < 1/(\nu(T) + \nu(U))$, we have $|z| < 1/\nu(L(z))$ which proves that $L(z) \in \mathcal{M}$. Lemma 4 applied to Relation (3) with $X(z) = \Phi_G(z)$, $H = \Phi_T(z)$ and $M = L(z)$ leads to

$$\Phi_G(z) = \Phi_T(z)\Phi_{L(z)}(z) \quad (4)$$

for $|z| < \min\{1/\nu(G), 1/(\nu(T) + \nu(U))\}$ where $\nu(U) = (p_0 - p)r/c$.

In order to derive an expression of the potential kernel Φ_G given in (4), we first recall in the next lemma the expression of Φ_T obtained in [Barbot and Sericola, 2002]. For that, we introduce, for z such that $|z| \leq 1/4$, the function $C(z) = (1 - \sqrt{1 - 4z})/2z$.

Lemma 7 *Let $|z| < 1$ and $\eta(z) = C(pqz^2)$. Let $X(z)$ and $Y(z)$ be the matrices defined by*

$$\begin{aligned} X_{i,j}(z) &= (qz\eta(z))^i (pz\eta(z))^j \\ Y(z) &= \sum_{k=0}^{\infty} W^k X(z) V^k. \end{aligned}$$

For every z such that $|z| < \min\{1/2, c/(qr + c)\}$, we have

$$\Phi_T(z) = \eta(z)Y(z) +$$

$$qz\eta^2(z) \frac{(1 + \rho - \rho qz\eta(z))X(z) - \frac{r}{c}WX(z)}{(1 - qz\eta(z))(1 - \rho qz\eta(z))}. \quad (5)$$

Theorem 8 *For every z such that $|z| < 1/2$, we have*

$$L(z) = u(z)RX(z) + \eta(z)(p_0 - p)RX(z)V, \quad (6)$$

$$\Phi_{L(z)}(z) = I + \frac{z}{1 - zu(z)}L(z), \quad (7)$$

where $u(z) = (p_0 - p)(r/c - 1) \frac{\eta(z)}{1 - \rho qz\eta(z)}$.

Proof. Let z be such that $|z| < 1/2$. Since $RW = 0$ and $SW = R$, we have by definition of $X(z)$ and $Y(z)$

$$RY(z) = RX(z), \quad SX(z) = qz\eta(z)RX(z)$$

and

$$SY(z) = qz\eta(z)RX(z) + RX(z)V.$$

Lemma 7 leads to

$$L(z) = \eta(z)(p_0 - p) \left((r/c - 1)R + S \right) \left(Y(z) + \right.$$

$$\left. qz\eta(z) \frac{(1 + \rho - \rho qz\eta(z))X(z) - \frac{r}{c}WX(z)}{(1 - qz\eta(z))(1 - \rho qz\eta(z))} \right)$$

and using the relations above, we obtain (6). Consider now the successive powers $L^k(z)$ of matrix $L(z)$. Observing that $VR = 0$ and

$$X(z)RX(z) = X(z), \quad (8)$$

we easily get from (6) that $L^2(z) = u(z)L(z)$. It follows by induction that for every $k \geq 0$,

$$L^{k+1}(z) = u^k(z)L(z).$$

Since $|z| < 1/2$, it is easy to check, from the definition of the function C , that $|\eta(z)| \leq 2$ and therefore $|qz\eta(z)| < 1$. Moreover, since $\rho_0 < 1$, we have $(p_0 - p)(r/c - 1) < q(1 - \rho)$ and so $|u(z)| < 1$. Thus, we obtain

$$\begin{aligned} \Phi_{L(z)}(z) &= I + z \sum_{k=0}^{\infty} (zu(z))^k L(z) \\ &= I + \frac{z}{1 - zu(z)} L(z). \end{aligned}$$

Theorem 9 For $|z| < \min\{1/2, c/(qr + c), 1/(\nu(G) + \nu(U))\}$, we have

$$\begin{aligned} \Phi_G(z) &= \eta(z)Y(z) \\ &+ \eta(z) \frac{qz\eta(z)(1 + \rho - \rho qz\eta(z)) + \frac{zu(z)}{1 - zu(z)}}{(1 - qz\eta(z))(1 - \rho qz\eta(z))} X(z) \\ &+ \left(\frac{c}{r - c} \right) \frac{zu(z)}{(1 - qz\eta(z))(1 - zu(z))} X(z)V \\ &- \left(\frac{r}{c} \right) \frac{qz\eta^2(z)}{(1 - qz\eta(z))(1 - zu(z))} WX(z) \\ &- \left(\frac{r}{r - c} \right) \frac{qz^2\eta^2(z)u(z)}{(1 - qz\eta(z))(1 - zu(z))} WX(z)V \quad (9) \end{aligned}$$

Proof. Let z be such that $|z| < \min\{1/2, c/(qr + c), 1/(\nu(G) + \nu(U))\}$. Replacing Relations (5) and (7) in (4), we obtain

$$\begin{aligned} \Phi_G(z) &= \eta(z) \left(I + \frac{z}{1 - zu(z)} L(z) \right) \left(Y(z) + \right. \\ &\left. qz\eta(z) \frac{(1 + \rho - \rho qz\eta(z))X(z) - \frac{r}{c}WX(z)}{(1 - qz\eta(z))(1 - \rho qz\eta(z))} \right). \quad (10) \end{aligned}$$

Now, since $VR = 0$, we obtain from (8) that $Y(z)RX(z) = X(z)$. We get from (6),

$$Y(z)L(z) = u(z)X(z) + \eta(z)(p_0 - p)X(z)V$$

and using (8), $X(z)L(z) = Y(z)L(z)$. Putting these relations in (10), we obtain (9).

3.2 Explicit Solution For A Single Buffer

We obtain in this section a closed-form expression for $G_{0,j}^n$ and so for $\Pr\{Q \leq x\} = \sum_{j=0}^{\infty} F_j(x)$. For

that purpose, we need the following well-known lemma which gives an analytical expression of the powers of $\eta(z)$. For the proof, see e.g. [Riordan, 1968] page 154.

Lemma 10 For every $k \geq 1$ and $|z| \leq 1/4$, we have $C^k(z) = \sum_{n=0}^{\infty} s(k, n)z^n$ where $s(k, n)$ are the ballot defined by

$$s(k, n) = k \frac{(2n + k - 1)!}{n!(n + k)!}.$$

Theorem 11 For every $x \geq 0$,

$$\begin{aligned} \Pr\{Q \leq x\} &= (1 - \rho_0) \sum_{n=0}^{\infty} e^{-\theta x} \frac{(\theta x)^n}{n!} \\ &\times \left(1 + (p_0 - p) \frac{\theta x}{n + 1} \right) \sum_{j=0}^n \left(\frac{p}{q} \right)^j \sum_{m=0}^{n-j} \gamma^m \\ &\times \sum_{k=0}^{\lfloor \frac{n-j-m}{2} \rfloor} s(n - 2k + 1, k) p^k q^{n-m-k} \\ &\times \sum_{h=0}^{n-j-m-2k} \frac{(m+h)!}{h!} \rho^h \end{aligned}$$

where $\lfloor u \rfloor$ denotes the largest integer less than or equal to the real number u and

$$\gamma = (p_0 - p)(r/c - 1) \in [0, 1].$$

Proof. Let z be such that $|z| < \min\{1/2, c/(qr + c), 1/(\nu(G) + \nu(U))\}$. Since the first row of the matrix $WX(z)$ has all its entries equal to zero, we have from (9), for every $j \in \mathbb{N}$,

$$\begin{aligned} (\Phi_G(z))_{0,j} &= \eta(z)Y_{0,j}(z) \\ &+ \eta(z) \frac{qz\eta(z)(1 + \rho - \rho qz\eta(z)) + \frac{zu(z)}{1 - zu(z)}}{(1 - qz\eta(z))(1 - \rho qz\eta(z))} X_{0,j}(z) \\ &+ \left(\frac{c}{r - c}\right) \frac{zu(z)}{(1 - zu(z))(1 - qz\eta(z))} (X(z)V)_{0,j} \end{aligned}$$

By definition of $X(z)$, $Y(z)$ and V , we can easily verify that

$$Y_{0,j}(z) = X_{0,j}(z) = (pz\eta(z))^j, \quad (X(z)V)_{0,0} = 0$$

and

$$(X(z)V)_{0,j} = (pz\eta(z))^{j-1}.$$

So, we obtain

$$(\Phi_G(z))_{0,0} = \frac{\eta(z)}{(1 - qz\eta(z))(1 - \rho qz\eta(z))(1 - zu(z))} \quad (11)$$

and for $j \geq 1$

$$\begin{aligned} (\Phi_G(z))_{0,j} &= \frac{p^j z^j \eta^{j+1}(z)}{(1 - qz\eta(z))(1 - \rho qz\eta(z))(1 - zu(z))} \\ &+ \left(\frac{c}{r - c}\right) \frac{p^{j-1} z^j \eta^{j-1}(z) u(z)}{(1 - qz\eta(z))(1 - zu(z))}. \quad (12) \end{aligned}$$

Before inverting the expressions (11) and (12), it must be remembered that for $|x| < 1$ and $n \in \mathbb{N}$

$$(1 - x)^{-n-1} = \sum_{l=0}^{\infty} \frac{(n+l)!}{l!} x^l.$$

For $|z| < 1/2$, we have $|u(z)| < 1$ and $|qz\eta(z)| < 1$ and therefore, using the Cauchy product of two series, we obtain

$$\begin{aligned} (\Phi_G(z))_{0,0} &= \frac{\eta(z)}{(1 - qz\eta(z))(1 - \rho qz\eta(z))} \sum_{n=0}^{\infty} (zu(z))^n \\ &= \frac{\eta(z)}{1 - qz\eta(z)} \sum_{n=0}^{\infty} (\gamma z\eta(z))^n (1 - \rho qz\eta(z))^{-n-1} \\ &= \frac{\eta(z)}{1 - qz\eta(z)} \sum_{n,l=0}^{\infty} (\gamma z\eta(z))^n \frac{(n+l)!}{l!} (\rho qz\eta(z))^l \\ &= \eta(z) \sum_{n,l=0}^{\infty} (\gamma z\eta(z))^n \sum_{h=0}^l \frac{(n+h)!}{h!} (\rho qz\eta(z))^h \\ &\quad \times (qz\eta(z))^{l-h} \\ &= \sum_{n,l=0}^{\infty} z^{n+l} \eta^{n+l+1}(z) \gamma^n q^l \sum_{h=0}^l \frac{(n+h)!}{h!} \rho^h. \end{aligned}$$

From Lemma 10, we have

$$\begin{aligned} \eta^{n+l+1}(z) &= C^{n+l+1}(pqz^2) \\ &= \sum_{k=0}^{\infty} s(n+l+1, k) p^k q^k z^{2k} \end{aligned}$$

which leads, by changing the order of summations, to

$$\begin{aligned} (\Phi_G(z))_{0,0} &= \sum_{n,l,k=0}^{\infty} z^{n+l+2k} s(n+l+1, k) \gamma^n p^k q^{l+k} \\ &\quad \times \sum_{h=0}^l \frac{(n+h)!}{h!} \rho^h \\ &= \sum_{n=0}^{\infty} z^n \sum_{m=0}^n \gamma^m \sum_{k=0}^{\lfloor \frac{n-m}{2} \rfloor} s(n-2k+1, k) p^k \\ &\quad \times q^{n-m-k} \sum_{h=0}^{n-m-2k} \frac{(m+h)!}{h!} \rho^h. \end{aligned}$$

Then, we have for every $n \in \mathbb{N}$,

$$\begin{aligned} G_{0,0}^n &= \sum_{m=0}^n \gamma^m \sum_{k=0}^{\lfloor \frac{n-m}{2} \rfloor} s(n-2k+1, k) p^k \\ &\quad \times q^{n-m-k} \sum_{h=0}^{n-m-2k} \frac{(m+h)!}{h!} \rho^h. \quad (13) \end{aligned}$$

Similarly, for $j \geq 1$, we obtain

$$\begin{aligned} (\Phi_G(z))_{0,j} &= z^j p^j \sum_{n=0}^{\infty} z^n \sum_{m=0}^n \gamma^m \sum_{k=0}^{\lfloor \frac{n-m}{2} \rfloor} p^k q^{n-m-k} \\ &\quad \times s(n-2k+j+1, k) \sum_{h=0}^{n-m-2k} \frac{(m+h)!}{h!} \rho^h \\ &+ z^j p^{j-1} (p_0 - p) \sum_{n=0}^{\infty} z^n \sum_{m=0}^n \gamma^m \sum_{k=0}^{\lfloor \frac{n-m}{2} \rfloor} p^k q^{n-m-k} \\ &\quad \times s(n-2k+j, k) \sum_{h=0}^{n-m-2k} \frac{(m+h)!}{h!} \rho^h. \end{aligned}$$

Therefore, $G_{0,j}^n = 0$ if $n < j$, and for $n \geq j$

$$\begin{aligned} G_{0,j}^n &= \left(\frac{p}{q}\right)^j \sum_{m=0}^{n-j} \gamma^m \sum_{k=0}^{\lfloor \frac{n-j-m}{2} \rfloor} p^k q^{n-m-k} \\ &\quad \times s(n-2k+1, k) \sum_{h=0}^{n-j-m-2k} \frac{(m+h)!}{h!} \rho^h \\ &+ \frac{p_0 - p}{q} \left(\frac{p}{q}\right)^{j-1} \sum_{m=0}^{n-j} \gamma^m \sum_{k=0}^{\lfloor \frac{n-j-m}{2} \rfloor} p^k q^{n-m-k} \\ &\quad \times s(n-2k, k) \sum_{h=0}^{n-j-m-2k} \frac{(m+h)!}{h!} \rho^h. \quad (14) \end{aligned}$$

Putting Relations (13) and (14) in (2), we obtain the result by summing over j and changing the order of summations.

4 THE FLUID CONTENT OF THE $(i+1)$ TH BUFFER

We suppose that the stability condition of the tandem fluid queues is satisfied, that is, $\rho_M < 1$. For $1 \leq i \leq M-1$, we derive the distribution of the stationary level Q_{i+1} of the $(i+1)$ th buffer.

Theorem 12 For every $x \geq 0$ and $1 \leq i \leq M-1$

$$\begin{aligned} \Pr\{Q_{i+1} \leq x\} &= (1 - \rho_i) \sum_{n=0}^{\infty} e^{-\theta_i x} \frac{(\theta_i x)^n}{n!} \\ &\times \left(1 + \frac{\lambda x}{(n+1)c_0(1 - c_{i+1}/c_i)} \right) \sum_{j=0}^n \left(\frac{p_i}{q_i} \right)^j \\ &\times \sum_{m=0}^{n-j} \gamma_i^m \sum_{k=0}^{\lfloor \frac{n-j-m}{2} \rfloor} s(n-2k+1, k) p_i^k q_i^{n-m-k} \\ &\times \sum_{h=0}^{n-j-m-2k} \frac{(m+h)!}{h!} \rho_i^h \end{aligned}$$

where

$$\begin{aligned} \lambda_i &= \lambda(N - c_i/c_0), \quad \mu_i = \mu c_i/c_0, \\ p_i &= \lambda_i/(\lambda_i + \mu_i), \quad q_i = 1 - p_i, \\ \theta_i &= (\lambda_i + \mu_i)/(c_i - c_{i+1}), \\ \rho'_i &= c_i \lambda_i / (c_{i+1} \mu_i), \\ \gamma_i &= (c_i/c_0)(c_i/c_{i+1} - 1)\lambda/(\lambda_i + \mu_i). \end{aligned}$$

Proof. We saw that the stationary level of the $(i+1)$ th buffer is equivalent to the stationary level of an infinite buffer with service rate c_{i+1} and fed by an MM1($N\lambda, \lambda_i, \mu_i, c_i$). We then apply Theorem 11 to this fluid model and set $\beta = N\lambda$, $a = \lambda_i$, $b = \mu_i$, $r = c_i$ and $c = c_{i+1}$.

REFERENCES

- Aalto S. 1998, "Output of a multiplexer loaded by heterogeneous on-off sources". *Commun. Statist.-Stochastic Models*, Vol. 14(4). Pp993–1005.
- Aalto S. and Scheinhardt W.R.W. 2000, "Tandem fluid queues fed by homogeneous on-off sources". *Op. Res. Letters*, Vol. 27. Pp73–82.

Anick D., Mitra D. and Sondhi M. M. 1982, "Stochastic Theory of a Data-Handling System with Multiple Sources". *Bell Syst. Tech. J.*, Vol. 61(8). Pp1871–1894.

Barbot N. and Sericola B. 2002, "Stationary solution to the fluid queue fed by an M/M/1 queue". *J. Appl. Prob.* Vol. 39. Pp359–369.

Boxma O.J. and Dumas V. 1998, "The busy period in the fluid queue". *Perf. Eval. Rew.* Vol. 26. Pp100–110.

Leguesdron P., Pellaumail J., Rubino G. and Sericola B. 1993, "Transient analysis of the M/M/1 queue". *Adv. Appl. Prob.* Vol. 25. Pp702–713.

Mitra D. 1988, "Stochastic theory of a fluid model of producers and consumers coupled by a buffer". *Adv. Appl. Prob.* Vol. 20. Pp646–676.

Riordan J. 1968, *Combinatorial identities*, John Wiley, New York.

Sericola B. and Tuffin B. 1999, "A fluid queue driven by a Markovian queue". *Queueing Systems* Vol. 31. Pp253–264.

BIOGRAPHY



NELLY BARBOT is a temporary associate professor in applied mathematics at ENSSAT, an engineering school (École Nationale Supérieure de Sciences Appliquées et Technologie), University of Rennes I (France). She received the Ph.D. degree in applied mathematics from the University of Rennes I in 2002. She is also part of the ARMOR (Networks ARchitectures and Modeling) group at IRISA-INRIA laboratory. Her research subject is communication systems performance evaluation with stochastic models.



BRUNO SERICOLA received the Ph.D. degree in computer science from the University of Rennes I in 1988. He has been with INRIA (Institut National de Recherche en Informatique et Automatique, a public research French laboratory) since 1989. His main research activity is in computer and communication systems performance evaluation, dependability and performability analysis of fault-tolerant architectures and applied stochastic processes.

PRODUCT FORM OVER ON-OFF COMPONENTS IN PEPA

NIGEL THOMAS

*Research Institute in Software Evolution
University of Durham, UK. Nigel.Thomas@durham.ac.uk*

Abstract: In the study of stochastic process algebra it is necessary to consider not only how systems are to be specified, but also how complex systems can be simplified and solved efficiently. In this paper a relationship between the behaviour of stochastic systems and a product form solution over components is explored. A number of characterisations of increasing complexity are derived which extend the class of model subject to product form solution that have been defined for Markovian process algebra.

Keywords: Product form, Markovian process algebra, PEPA, decomposition

1. INTRODUCTION

In recent years some effort has been made to identify efficient methods for analysing and solving stochastic process algebra models by decomposition (see [Hillston, 2001]). Such solutions are derived on the basis that the components in the model are statistically independent in their steady state behaviour and so the steady state solutions for components may be found in isolation without the need to generate the entire state space of the model. Clearly product form solutions are an extremely efficient mechanism in deriving important numerical solutions.

The aim of this paper is to address the issue of how the behaviour of the model may be used to directly show product form results in general models without relying on additional insight from the modeller. The product form solutions that are derived here are related to the class of model previously defined by Boucherie [Boucherie, 1994]. This class of model gives a product form over components interacting only through a resource, to which exclusive access is granted to one component at a time. No other interaction is possible between components and without the resource a component may only carry out internal actions.

The paper begins by re-introducing Hillston's Markovian process algebra, PEPA [Hillston, 1996], together with the set of concepts required to describe features of a model and then briefly discusses the notion

of behavioural independence and control. In Section 3 the exploitation of behavioural independence is made in relation to simple product form decomposition. In Section 4 this is developed to consider more complex classes of model. Finally some conclusions and future work directions are presented.

2. PEPA

A formal presentation of PEPA is given in [Hillston, 1996], in this section a brief informal summary is presented. PEPA, being a Markovian Process Algebra, only supports actions that are negative exponentially distributed at given rates. Specifications written in PEPA represent Markov processes and can be mapped to a continuous time Markov chain (CTMC). Systems are specified in PEPA in terms of *activities* and *components*. An activity (α, r) is described by the *type* of the activity, α , and the *rate* of the associated negative exponential distribution, r . This rate may be any positive real number, or given as *unspecified* using the symbol \top . The syntax for describing components is given as:

$$P ::= (\alpha, r).P \mid P + Q \mid P/L \mid P \underset{L}{\bowtie} Q \mid A$$

The component $(\alpha, r).P$ performs the activity of type α at rate r and then behaves like P . The component $P + Q$ behaves either like P or like Q , the resultant behaviour being given by the first activity to complete. The component P/L behaves exactly like P except that

the activities in the set L are concealed, their type is not visible and instead appears as the unknown type τ . Concurrent components can be synchronised, $P \bowtie_L Q$, such that activities in the *cooperation set* L involve the participation of both components. In PEPA the shared activity occurs at the slowest of the rates of the participants and if a rate is unspecified in a component, the component is passive with respect to the activities of that type. The parallel combinator \parallel is used as shorthand to denote synchronisation with no shared activities, i.e. $P \parallel Q \equiv P \bowtie_{\emptyset} Q$. $A \stackrel{\text{def}}{=} P$ gives the constant A the behaviour of the component P . A small number of addition definitions are required.

DEFINITION 1: Fertile action. An action γ is said to be fertile in derivative P_i if $P_i \xrightarrow{\gamma} P_j$ and $i \neq j$.

DEFINITION 2: Current fertile action type set. The current fertile action type set of P , denoted $\mathcal{A}_f(P)$, is the set of all action types of actions that are fertile in the current derivative of P .

DEFINITION 3: Complete fertile action type set. The complete fertile action type set of P , denoted $\vec{\mathcal{A}}_f(P)$, is the set of all action types of actions that are fertile in at least one derivative of P .

3. BEHAVIOURAL INDEPENDENCE

Put simply the notion of *behavioural independence* is simply that a component in a model behaves identically regardless of the current behaviour of the other components in the model.

DEFINITION 4: Behavioural Independence. The component P is said to be behaviourally independent in the model $P \bowtie_L Q$ if for every $P_i \in ds(P)$
 $\forall Q_j, Q_k \in ds(Q)$ s.t. $(P_i \bowtie_L Q_j), (P_i \bowtie_L Q_k) \in ds(P \bowtie_L Q)$

$$\begin{aligned} & \text{Act} \left((P_i \bowtie_L Q_j) / \{ \mathcal{A}(P_i \bowtie_L Q_j) / \{ \mathcal{A}_f(P_i) \cap L \} \} \right) \\ & \quad = \\ & \text{Act} \left((P_i \bowtie_L Q_k) / \{ \mathcal{A}(P_i \bowtie_L Q_k) / \{ \mathcal{A}_f(P_i) \cap L \} \} \right) \end{aligned}$$

Obviously the trivial case for behavioural independence is where there are no shared actions, i.e. $P \parallel Q$, however this is not the only case where components may be considered to be behaviourally independent. Furthermore, the fact that no actions are shared between two components does not mean they will always

be behaviourally independent in the presence of other components. For example, in $(P \parallel Q) \bowtie_L R$ the interaction between P and R may influence the interaction between Q and R , causing P and Q to be behaviourally *dependent*. If a component is not behaviourally independent then it must be dependent on some other component to perform one of more actions during its evolution. This dependence is referred to by saying that component P *controls* component Q over actions $K \subset L$ in $P \bowtie_L Q$ if the rate at which an action of type $k \in K$ can happen in $Q_i \in ds(Q)$ depends on the current derivative of P . Clearly, if P controls Q over K then Q cannot be behaviourally independent, but the independence, or otherwise, of P is not known by this statement.

3.1 Exploiting behavioural independence

It is clear that if a component is behaviourally independent (even it is also controlling) then it may be studied in isolation without affecting its behaviour, subject to the rates of shared actions being set. If a shared action is not enabled by the partner in the cooperation then the rate of that action will be zero when that component is considered in isolation, otherwise the rate of the shared action will be determined by the rates specified in each participating component. To illustrate such a situation, consider a number of queues in sequence. If all the queues are basic $M/M/1/\infty$ then the system is clearly a simple Jackson network and has a product form solution. However, even if this is not the case then the first queue will still be behaviourally independent (unless there is blocking at the server) and so may be studied in isolation.

As well as being used to identify independent behaviour leading to decomposition, behavioural independence and control can also be used to identify cases where product form solutions exist. Such a case is the queueing model with breakdowns illustrated below.

$$\begin{aligned} \text{Queue}_0 & \stackrel{\text{def}}{=} (\text{arrival}, \top). \text{Queue}_1 \\ \text{Queue}_i & \stackrel{\text{def}}{=} (\text{arrival}, \top). \text{Queue}_{i+1} \\ & \quad + (\text{service}, \top). \text{Queue}_{i-1} \\ & \quad , 1 \leq j \leq N-1 \\ \text{Queue}_N & \stackrel{\text{def}}{=} (\text{service}, \top). \text{Queue}_{N-1} \\ \text{Server}_{on} & \stackrel{\text{def}}{=} (\text{fail}, \xi). \text{Server}_{off} \\ & \quad + (\text{arrival}, \lambda). \text{Server}_{on} \\ & \quad + (\text{service}, \mu). \text{Server}_{on} \end{aligned}$$

$$Server_{off} \stackrel{def}{=} (repair, \eta).Server_{on}$$

$$Queue_0 \boxtimes_{\{service, arrival\}} Server_{on}$$

It is clear that the *Server* component is behaviourally independent in this model as neither of the shared actions affects its evolution. Similarly it is clear that the *Server* component controls the *Queue* component over the actions *service* and *arrival*. A number of other important factors are also apparent: all the actions of *Queue* are shared actions, all shared actions are enabled in *Server_{on}*, no shared actions are enabled in *Server_{off}*, the action *fail* does not alter the derivative of *Queue*. These six factors mean that a product form solution exists over the *Server* and *Queue* components such that the joint steady state probabilities are given as $\pi_{(Server_j, Queue_i)} = \pi_{Server_j} \cdot \pi_{Queue_i}$ where $j \in \{on, off\}$ and $0 \leq i \leq N$.

The model illustrated in here is reversible and it is possible to derive a product form solution using the characterisation derived in [Hillston and Thomas, 1998]. In fact an example with the same structure, referred to as *the drinking gambler* appeared in [Hillston and Thomas, 1998]. That approach required the identification of reversible components and the application of restrictions on the cooperation between them. This requires a detailed study of both the components and the interface, whereas the approach described here only requires a simple inspection of the components, only adherence to the five criteria.

1. Component, A , of a pair $A \boxtimes_L B$ is behaviourally independent.
2. Component, B , is controlled by A over all the actions in the cooperation set, $\mathcal{K}(B) = L$.
3. The complete action type set of A , $\vec{\mathcal{A}}(B)$ is contained within its interface, $\vec{\mathcal{A}}(B) = L$.
4. All actions in the cooperation set, L , are enabled in exactly one derivative of A , A_i .
5. No actions in the cooperation set, L , are enabled in any other derivative of A .
6. Any action α such that $A_i \xrightarrow{\alpha} A_j$, $A_i \neq A_j$ is not fertile in B .

This product form solution relies on the fact that component A turns the interaction in the model off and on; and when it turns back on the system returns to exactly the same global state as it was before it turned

off. Hence, as long as these 5 stated conditions are not broken then the model illustrated above can be easily adapted to incorporate additional (non-reversible) features, such as batch service, without compromising the product form solution.

$$Queue_0 \stackrel{def}{=} (arrival, \top).Queue_1$$

$$Queue_i \stackrel{def}{=} (arrival, \top).Queue_{i+1} + (service, \top).Queue_0, \quad 1 \leq j \leq N-1$$

$$Queue_N \stackrel{def}{=} (service, \top).Queue_0$$

$$Server_{on} \stackrel{def}{=} (fail, \xi).Server_{off} + (arrival, \lambda).Server_{on} + (service, \mu).Server_{on}$$

$$Server_{off} \stackrel{def}{=} (repair1, \eta_1).Server_{standby}$$

$$Server_{standby} \stackrel{def}{=} (repair2, \eta_2).Server_{on}$$

$$Queue_0 \boxtimes_{\{service, arrival\}} Server_{on}$$

4. PARTIAL BEHAVIOURAL INDEPENDENCE

The simple product form developed in Section 3 can be extended by considering parts of components as behaviourally independent and parts which exert control. This may be achieved by observing that component P controls Q over the set of actions K , but that the actions in K are not affected by changes in derivative within a subset of P .

DEFINITION 5: Partial Behavioural Independence

The component P is said to have partial behavioural independence in the model $P \boxtimes_L Q$ with respect to the subset $\mathcal{D}(Q) \subset ds(Q)$, if for every $P_i \in ds(P)$ $\forall Q_j, Q_k \in \mathcal{D}(Q)$ s.t. $(P_i \boxtimes_L Q_j), (P_i \boxtimes_L Q_k) \in ds(P \boxtimes_L Q)$

$$Act \left((P_i \boxtimes_L Q_j) / \{ \mathcal{A}(P_i \boxtimes_L Q_j) / \{ \mathcal{A}_f(P_i) \cap L \} \} \right)$$

$$=$$

$$Act \left((P_i \boxtimes_L Q_k) / \{ \mathcal{A}(P_i \boxtimes_L Q_k) / \{ \mathcal{A}_f(P_i) \cap L \} \} \right)$$

This definition states that the P will behave identically as long as Q behaves as some derivative $Q_i \in \mathcal{D}(Q)$. Clearly there may be many such subsets for any given component. In product form solution introduced in Section 3.1, the component A has two subsets; one subset consists of the single derivative where all actions

in the cooperation set, L , are enabled, and the other subset consists off all the other derivatives where no actions in the cooperation set are enabled. The single "on" behaviour restriction from A can be relaxed as long as the result of returning to "on" always returns to exactly the same derivative of A as immediately before turning "off".

1. Component, A , of a pair $A \bowtie_L B$ is behaviourally independent.
2. Component, B , is controlled by A over all the actions in the cooperation set, $\mathcal{K}(B) = L$.
3. The complete action type set of B , $\vec{\mathcal{A}}(B)$ is contained within its interface, $\vec{\mathcal{A}}(B) = L$.
4. It is possible to divide the derivatives of A into N subsets $\mathcal{D}_1(A), \dots, \mathcal{D}_N(A)$, $N \geq 2$, such that $\bigcup_{i=1}^N \mathcal{D}_i(A) = ds(A)$, $\mathcal{D}_i(A) \cap \mathcal{D}_j(A) = \emptyset$, $i \neq j$, and B has partial behavioural independence in $A \bowtie_L B$ with respect to $\mathcal{D}_i(A)$, $i = 1, \dots, N$.
5. All actions in the cooperation set, L , are enabled in all derivatives in $\mathcal{D}_1(A)$.
6. No actions in the cooperation set, L , are enabled in any derivative in $\mathcal{D}_i(A)$, $i \geq 2$.
7. For any $l \geq 2$, there exists at most one derivative, $A_i \in \mathcal{D}_1(A)$ such that $A_i \xrightarrow{\alpha} A_j$ and $A_k \xrightarrow{\alpha} A_i$ where $A_j, A_k \in \mathcal{D}_l(A)$.
8. For any $i \neq j, \geq 2$ there are no derivatives $A_k \in \mathcal{D}_i(A)$ and $A_l \in \mathcal{D}_j(A)$ such that $A_k \xrightarrow{\alpha} A_l$.
9. Any action α such that $A_i \xrightarrow{\alpha} A_j$, $A_i \in \mathcal{D}_1(A)$, $A_j \notin \mathcal{D}_1(A)$, α is not fertile in B .

This set of rules allows for multiple "on" behaviours and multiple "off" behaviours, by partitioning the "off" behaviours into distinct subsets that do not have any single actions linking them. However, this definition still imposes the restriction the "on" and "off" behaviours are controlled by a single behaviourally independent component. It is possible to relax even this restriction, however, this can be done only if the actions in the "on" subset, $\mathcal{D}_1(A)$ are restricted to the actions in the cooperation set. This further restriction would mean that two components A and C , could both control B over L , and also that A controls C over L and C controls A over L . In fact it is not necessary for the component B to be constrained by its interface, and it can in fact be an on-off component in the same way as A .

DEFINITION 6: Restricted Partial Behavioural Independence The component P is said to have restricted partial behavioural independence in the model $P \bowtie Q$ with respect to the subsets $\mathcal{D}(Q) \subset ds(Q)$ and $\mathcal{D}(P) \subset ds(P)$, if for every $P_i \in \mathcal{D}(P)$
 $\forall Q_j, Q_k \in \mathcal{D}(Q)$ s.t. $(P_i \bowtie_L Q_j), (P_i \bowtie_L Q_k) \in ds(P \bowtie_L Q)$

$$\begin{aligned} & Act \left((P_i \bowtie_L Q_j) / \{ \mathcal{A}(P_i \bowtie_L Q_j) / \{ \mathcal{A}_f(P_i) \cap L \} \} \right) \\ & = \\ & Act \left((P_i \bowtie_L Q_k) / \{ \mathcal{A}(P_i \bowtie_L Q_k) / \{ \mathcal{A}_f(P_i) \cap L \} \} \right) \end{aligned}$$

Thus it is possible to derive a product form solution in a model without a resource component subject to the following conditions.

1. Two components, A and B in $A \bowtie_L B$ are controlled and controlling over all the actions in the cooperation set, $\mathcal{K}(B) = \mathcal{K}(A) = L$.
2. It is possible to divide the derivatives of A into N distinct subsets $\mathcal{D}_1(A), \dots, \mathcal{D}_N(A)$, $N \geq 2$, and the derivatives of B into M subsets $\mathcal{D}_1(B), \dots, \mathcal{D}_M(B)$, $M \geq 2$ such that B has restricted partial behavioural independence in $A \bowtie_L B$ with respect to $\mathcal{D}_i(A)$ and $\mathcal{D}_1(B)$, $i = 1, \dots, N$ and A has restricted partial behavioural independence in $A \bowtie_L B$ with respect to $\mathcal{D}_j(B)$ and $\mathcal{D}_1(A)$, $j = 1, \dots, M$.
3. All actions in the cooperation set, L , are enabled in all derivatives in $\mathcal{D}_1(A)$ and all derivatives in $\mathcal{D}_1(B)$.
4. No actions in the cooperation set, L , are enabled in any derivative in $\mathcal{D}_i(A)$, $i \geq 2$, or any derivative in $\mathcal{D}_j(B)$, $j \geq 2$.
5. The current action type set of all $A_i \in \mathcal{D}_1(A)$, $\mathcal{A}(A_i)$, is contained within its interface; $\mathcal{A}(A_i) \subset L, \forall A_i \in \mathcal{D}_1(A)$.
6. The current action type set of all $B_j \in \mathcal{D}_1(B)$, $\mathcal{A}(B_j)$, is contained within its interface; $\mathcal{A}(B_j) \subset L, \forall B_j \in \mathcal{D}_1(B)$.
7. For any $l \geq 2$, there exists at most one derivative, $A_i \in \mathcal{D}_1(A)$ such that $A_i \xrightarrow{\alpha} A_j$ and $A_k \xrightarrow{\alpha} A_i$ where $A_j, A_k \in \mathcal{D}_l(A)$.
8. For any $i \neq j, \geq 2$ there are no derivatives $A_k \in \mathcal{D}_i(A)$ and $A_l \in \mathcal{D}_j(A)$ such that $A_k \xrightarrow{\alpha} A_l$.

9. For any $l \geq 2$, there exists at most one derivative, $B_i \in \mathcal{D}_1(B)$ such that $B_i \longrightarrow B_j$ and $B_k \longrightarrow B_i$ where $B_j, B_k \in \mathcal{D}_l(B)$.
10. For any $i \neq j, \geq 2$ there are no derivatives $B_k \in \mathcal{D}_i(B)$ and $B_l \in \mathcal{D}_j(B)$ such that $B_k \longrightarrow B_l$.
11. Any action α such that $A_i \xrightarrow{\alpha} A_j, A_i \in \mathcal{D}_1(A), A_j \notin \mathcal{D}_1(A)$ is not fertile in B .
12. Any action β such that $B_i \xrightarrow{\alpha} B_j, B_i \in \mathcal{D}_1(B), B_j \notin \mathcal{D}_1(B)$ is not fertile in A .

These conditions follow as a simple consequence of the earlier discussion. All interaction between A and B is restricted to the subsets $\mathcal{D}_1(A)$ and $\mathcal{D}_2(B)$. If A enters a behaviour outside $\mathcal{D}_1(A)$, then all actions of B will be blocked until A returns to $\mathcal{D}(A)$. Furthermore, if A was behaving as $A_1 \in \mathcal{D}_1(A)$ and the action α occurred causing A to behave as $A'_1 \notin \mathcal{D}_1(A)$ then B will be blocked until A is once again behaving as A_1 . Likewise B will block the actions of A if its behaviour leaves the subset $\mathcal{D}_1(B)$. Clearly therefore this characterisation gives rise to a separable system, but in general this does not mean a product form solution will exist. This is only guaranteed if each of the activities in L are fertile in at most one component in every derivative of $A \bowtie_L B, A' \bowtie_L B'$, such that $A' \in \mathcal{D}_1(A)$ and $B' \in \mathcal{D}_1(B)$ (the same condition is present in the Boucherie product form). This gives a product form solution of the following structure.

$$\pi_{(A_j, B_k)} = \frac{1}{X} \pi_{A_j} \cdot \pi_{B_k}$$

where $1/X$ is the normalising constant resulting from the fact that not all combinations of derivatives of A and B are reachable in $A \bowtie_L B$.

4.1 Security guards example

In this simple example there are a pair of security guards. The only stipulation is that at least one must be awake and on duty at any time. Thus, any guard may choose to go to sleep only if another of his colleagues is awake. The correctness is held by the fact that in order to make the transition from $G_x Awake$ to $G_x Asleep$, each guard must have the (passive) cooperation of his colleague. Once asleep however, the guards are incapable of communicating, and so the guard who is awake must remain so.

$$G_A Awake \stackrel{def}{=} (afallAsleep, r_2).G_A Asleep +$$

$$\begin{aligned} & (bfallAsleep, \top).G_A Awake \\ G_A Asleep & \stackrel{def}{=} (wakeup, r_4).G_A Awake \\ G_B Awake & \stackrel{def}{=} (bfallAsleep, r_2).G_B Asleep \\ & + (afallAsleep, \top).G_B Awake \\ G_B Asleep & \stackrel{def}{=} (wakeup, r_6).G_B Awake \end{aligned}$$

$$G_A Awake \bowtie_{\{afallAsleep, bfallAsleep\}} G_B Awake$$

Thus the model has the excluded state of $G_A Asleep G_B Asleep$, and a trivial product form over the remaining states, given by,

$$\pi_{(G_A Y, G_B Z)} = \frac{1}{X} \pi_{G_A Y} \cdot \pi_{G_B Z}$$

where $Y, Z = \{Awake, Asleep\}$ and $X = \pi_{G_A Awake} \cdot \pi_{G_B Awake} + \pi_{G_A Awake} \cdot \pi_{G_B Asleep} + \pi_{G_A Asleep} \cdot \pi_{G_B Awake}$. The model can be made slightly more interesting if guards go out on patrol. There are a number of possibilities in this regard.

- Both guards patrol together at all times.

$$\begin{aligned} G_A Awake & \stackrel{def}{=} (goOut, r_7).G_A Patrol \\ & + (afallAsleep, r_2).G_A Asleep \\ & + (bfallAsleep, \top).G_A Awake \\ G_A Patrol & \stackrel{def}{=} (goBack, r_8).G_A Awake \\ G_B Awake & \stackrel{def}{=} (goOut, r_7).G_B Patrol \\ & + (bfallAsleep, r_2).G_B Asleep \\ & + (afallAsleep, \top).G_B Awake \\ G_B Patrol & \stackrel{def}{=} (goBack, r_8).G_B Awake \end{aligned}$$

$$G_A Awake \bowtie_{\{goOut, goBack, afallAsleep, bfallAsleep\}} G_B Awake$$

- One or both of the guards go out, the other must be awake.

$$\begin{aligned} G_A Awake & \stackrel{def}{=} (goOut, r_7).G_A Patrol \\ & + (goOut, \top).G_A Awake \\ & + (afallAsleep, r_2).G_A Asleep \\ & + (bfallAsleep, \top).G_A Awake \\ G_A Patrol & \stackrel{def}{=} (goBack, r_8).G_A Awake \\ G_B Awake & \stackrel{def}{=} (goOut, r_9).G_B Patrol \\ & + (goOut, \top).G_B Awake \\ & + (bfallAsleep, r_2).G_B Asleep \\ & + (afallAsleep, \top).G_B Awake \\ G_B Patrol & \stackrel{def}{=} (goBack, r_8).G_B Awake \end{aligned}$$

$$G_A Awake \underset{\{a fall Asleep, b fall Asleep, go Out\}}{\boxtimes} G_B Awake$$

Here both components might be simultaneously passive on *goOut*, however this is not a fertile action in this case. This case only has a product form if $r_7 = r_9$.

- *Either guard may go out at anytime regardless of whether the other is awake or not.*

$$\begin{aligned} G_A Awake &\stackrel{def}{=} (goOut, r_7).G_A Patrol \\ &\quad + (goOut, \top).G_A Awake \\ &\quad + (a fall Asleep, r_2).G_A Asleep \\ &\quad + (b fall Asleep, \top).G_A Awake \end{aligned}$$

$$\begin{aligned} G_A Patrol &\stackrel{def}{=} (goBack, r_8).G_A Awake \\ &\quad + (b fall Asleep, \top).G_A Awake \end{aligned}$$

$$\begin{aligned} G_B Awake &\stackrel{def}{=} (goOut, r_9).G_B Patrol \\ &\quad + (goOut, \top).G_B Awake \\ &\quad + (b fall Asleep, r_2).G_B Asleep \\ &\quad + (a fall Asleep, \top).G_B Awake \end{aligned}$$

$$\begin{aligned} G_B Patrol &\stackrel{def}{=} (goBack, r_8).G_B Awake \\ &\quad + (a fall Asleep, \top).G_B Awake \end{aligned}$$

$$G_A Awake \underset{\{a fall Asleep, b fall Asleep\}}{\boxtimes} G_B Awake$$

This final model is not captured by the characterisation presented here, but instead belongs to a further class of model where independent actions are allowed to continue. Capturing this class of model remains ongoing work.

5. CONCLUSIONS

In this paper a discussion of the notions of behavioural independence and control has been presented in relation to product form solution. It is probable that many additional classes of product form solution will have subclasses that can be defined using behavioural independence and control, although this remains to be proved. By exploring these subclasses of solutions it will be possible to gain greater understanding of the links between different product form solutions and near and non-product form solutions and where they may overlap. The class of product form here is related to the class defined by Boucherie [Boucherie, 1994] and is thus related to an earlier characterisation in PEPA [Hillston and Thomas, 1999], although it is conceptually somewhat simpler than either of these cases. These

earlier characterisations have some advantages over that presented here, however the class defined here differs from those earlier characterisations in three important ways. The equivalent of the resource component used here is not redundant. This means that queues (and other state dependent structures) can be used under a Boucherie-like framework. Furthermore, by developing the notion of *restricted partial behavioural independence* it has been possible to achieve a characterisation without an explicit resource. Given the Boucherie result it should be possible to relax the on-off behaviour such that component B in $A \boxtimes B$ will still be able to perform internal actions even if A is in some derivative $A' \notin \mathcal{D}_1(A)$. This remains as future work.

REFERENCES

- Boucherie R. 1994, A Characterisation of Independence for Competing Markov Chains with Applications to Stochastic Petri nets, *IEEE Transactions on Software Engineering* **20**(7).
- Clark G. Gilmore S. Hillston J. and Thomas N. 1999, Experiences with the PEPA performance modelling tools, *IEE Proceedings - Software*, **146**(1).
- Harrison P.G. and Hillston J. 1995, Exploiting Quasi-reversible Structures in Markovian Process Algebra Models, *The Computer Journal*, **38**(7), pp. 510–520.
- Hillston J. 1996, *A Compositional Approach to Performance Modelling*, Cambridge University Press.
- Hillston J. 2001, Exploiting Structure in Solution: Decomposing Composed Models, in: *FMPA Lecture Notes*, Springer-Verlag.
- Hillston J. and Thomas N. 1999, Product Form Solution for a Class of PEPA Models, *Performance Evaluation*, **35**(3-4), pp. 171-192.
- Hillston J. and Thomas N. 1998, A Syntactic Analysis of Reversible PEPA Models, in: *Proceedings of the Sixth International Workshop on Process Algebra and Performance Modelling*.
- Kelly F.P. 1979, *Reversibility and Stochastic Networks*, Wiley.
- Thomas N. 2002, Exploiting behavioural independence and control in Markovian process algebra, in: *Proceedings of the 1st Workshop on Process Algebra with Stochastic Timed Activities*, University of Edinburgh.

APPROXIMATE SOLUTION OF A CLASS OF QUEUEING NETWORKS WITH BREAKDOWNS

NIGEL THOMAS*, DAVID THORNLEY[†] and HARF ZATSCHLER[†]

Abstract: In this paper we study a class of open queueing network where servers suffer breakdowns and are subsequently repaired. The network topology is a pipeline with feedback from the final node to the first. Each node consists of a number of queues each with an unreliable server. There are no losses from the queues in this system, however jobs are routed according to the distribution of operational servers at each node in the pipeline. This model is in general intractable, however an iterative technique is presented which combines a number of earlier results to generate an approximation to steady state measures found by simulation.

Keywords: Queueing theory, breakdowns, approximation, decomposition

1. INTRODUCTION

Queueing networks with breakdowns are a class of problem that are of obvious practical interest and have consequently been considered for many years. However, the vast majority of studies that have been made concern only single queue models or solve more general topologies using simulation. A number of papers have addressed the problem of queues in parallel, most notably Mitrani and Wright [Mitrani and Wright, 1994] who analysed a system of nodes in parallel which suffered failures that caused all jobs to be lost, incoming jobs were then routed away from failed nodes, this resulted in an interesting trade off in performance between response time and job loss. Models without loss on failure are not without practical application, particularly in transaction processing and manufacturing. Thomas and Mitrani [Thomas and Mitrani, 1995] started with the same basic model as [Mitrani and Wright, 1994], but changed the nature of the failure so that queues were preserved during repair periods. The same authors also considered an extension to their model [Thomas and Mitrani, 1998] where a pipeline was constructed where each node was a system of parallel queues. It was not possible to solve this model exactly, instead they considered each node in series and compared a simple Poisson approximation with

a Markov-modulated arrival process based on the configuration of operational servers at the previous node in the pipeline.

In this paper we present an extension to the model presented in [Thomas and Mitrani, 1998] to consider the existence of a feedback loop which returns jobs to the start of the pipeline with a given probability. The existence of such a loop means that the approach used previously will no longer be applicable because all the nodes are now dependent of their predecessor, whereas in [Thomas and Mitrani, 1998] the first node had only external arrivals. We employ an iterative approach recently applied to Markovian process algebra [Thomas et al, 2003]. In the context of the queueing systems described here this iterative method is extremely close to that applied recently in [Harrison et al, 2002]. The approach described in [Thomas et al, 2003] requires that all shared actions (in queueing terms this refers to departures from one node which become arrivals at another) are represented in a reduced model to estimate the marginal distribution for each component (node) in turn. The method is repeated until convergence over a particular measure is reached and hence all the marginal distributions are found. In this paper the reduced model is a single queue with Markov-modulated arrivals and convergence is required of the parameters of the arrival process at each node. As in earlier studies these marginal queue size distributions do not in general give rise to a product form solution, but nevertheless can be used to find many performance measures of interest, such as average response time and utilisation.

*Department of Computer Science, University of Durham, UK.
nigel.thomas@durham.ac.uk

[†]Department of Computing, Imperial College London, UK.
{djt,hz3}@doc.ic.ac.uk

2 THE MODEL

Jobs arrive into the system in a Poisson stream with rate λ . There are K nodes in series and in node i there are N_i servers in parallel, each with an associated unbounded queue, to which incoming jobs may be directed. Server j at node i goes through alternating independent operative and inoperative periods, distributed exponentially with means $1/\xi_{i,j}$ and $1/\eta_{i,j}$ respectively. While it is operative, the jobs in its queue receive service of an exponentially distributed duration with mean $1/\mu_{i,j}$, and leave the node upon completion to proceed to the next (if any) node in the pipeline. When a server becomes inoperative (breaks down), the corresponding queue, including the job in service (if any), remains in place. Services that are interrupted in this way are eventually resumed from the point of interruption. On completion of service at the final node a proportion of jobs, $0 < \phi \leq 1$, leave the system and the remainder return to the first node. The system model is illustrated in Figure 1.

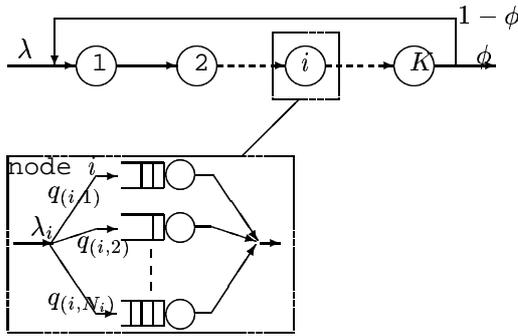


Figure 1. A single source to a pipeline of K nodes, split between the queues in each node

The external arrival rate is given in Figure 1 as λ_i , and since no jobs are lost the overall arrival rate at all nodes will be the same, namely, λ/ϕ . However, since the arrivals at node i depend on the departures from node $i - 1$ then the arrival stream will, in general, cease to be Poisson (the only case where this is not true is for node 1 if $\phi = 1$). The *system configuration* at any moment is specified by the subset, σ , of servers that are currently operative (that subset may be empty, or it may be the set of all servers): $\sigma \subset \Omega_N$, where $\Omega_N = \{(1, 1), (1, 2), \dots, (1, N_1), (2, 1), \dots, (K, N_K)\}$, where the pair $\{i, j\}$ represents server j at node i . There are of course 2^N possible system configurations, where $N = \sum_{i=1}^K N_i$. In general it is more convenient to consider the subset σ_i whose elements are those servers at node i which are operative. The set of

all servers at node i is denoted by Ω_{N_i} . Clearly $\sigma_i \subset \Omega_{N_i} \subset \Omega_N$ and $\sigma_i \subset \sigma$. The steady-state marginal probability, p_{σ_i} , of configuration σ_i at node i is given by

$$p_{\sigma_i} = \prod_{j \in \sigma_i} \frac{\eta_{i,j}}{\xi_{i,j} + \eta_{i,j}} \prod_{j \in \bar{\sigma}_i} \frac{\xi_{i,j}}{\xi_{i,j} + \eta_{i,j}}, \quad \sigma_i \subset \Omega_{N_i},$$

And the steady-state marginal probability, p_σ , of configuration σ is given by

$$p_\sigma = \prod_{i,j \in \sigma} \frac{\eta_{i,j}}{\xi_{i,j} + \eta_{i,j}} \prod_{i,j \in \bar{\sigma}} \frac{\xi_{i,j}}{\xi_{i,j} + \eta_{i,j}}, \quad \sigma \subset \Omega_N,$$

where $\bar{\sigma}_i$ is the complement of σ_i with respect to Ω_{N_i} , $\bar{\sigma}$ is the complement of σ with respect to Ω_N and an empty product is by definition equal to 1. These expressions follow from the fact that servers break down and are repaired independently of each other.

If, at the time of arrival at node i , a new job finds the node in configuration σ_i , then it is directed to the queue at server j with probability $q_{i,j}(\sigma_i)$. These decisions are independent of each other, of past history, of the sizes of the various queues and of the state of any other node in the pipeline. Thus, a routing policy at node i is defined by specifying 2^{N_i} vectors,

$$\mathbf{q}_i(\sigma_i) = [q_{i,1}(\sigma_i), q_{i,2}(\sigma_i), \dots, q_{i,N_i}(\sigma_i)], \quad \sigma_i \subset \Omega_{N_i},$$

such that for every σ_i ,

$$\sum_{j=1}^{N_i} q_{i,j}(\sigma_i) = 1$$

There are clearly many strategies that can be employed using this system and a number have been studied previously. Intuitively, it seems better not to send jobs to queues where the server is inoperative, unless that is unavoidable. This suggests the following strategy: If the subset of operative servers at node i in the current system configuration is σ_i , and that subset is non-empty, send jobs to queue j only if $j \in \sigma_i$, with probability proportional to q_j :

$$q_j(\sigma_i) = \frac{q_j}{\sum_{l \in \sigma} q_l}, \quad j \in \sigma.$$

If σ is empty (i.e. all servers are broken), send jobs to queue j with probability q_j ($j = 1, 2, \dots, N_i$). Note that this strategy does not take account of the states of servers at other nodes in the system. However the existence of other nodes may have an effect on the optimal routing vector for a given strategy, for instances

in spreading jobs when all preceding servers are operative, but directing jobs only to fast servers when few preceding servers are operative.

The system state at time t is specified by the pair $[I(t), \mathbf{J}(t)]$, where $I(t)$ indicates the current configuration (the configurations can be numbered, so that $I(t)$ is an integer in the range $0, 1, \dots, 2^N - 1$), and $\mathbf{J}(t)$ is an integer vector whose k 'th element, $J_k(t)$, is the number of jobs in queue k ($k = 1, 2, \dots, N$). The integer k is used here instead of the pair i, j for simplicity, the relationship between k and i, j is a simple 1 to 1 mapping such that

$$j + \sum_{x=1}^{i-1} N_x = k$$

Under the assumptions that have been made, $X = \{[I(t), \mathbf{J}(t)], t \geq 0\}$ is an irreducible Markov process. The condition for ergodicity of X is that, for every queue i, j , the overall arrival rate is lower than the overall service capacity:

$$\sum_{\forall \sigma_i} \lambda_i p_{\sigma_i} q_{i,j}(\sigma_i) < \mu_{i,j} \frac{\eta_{i,j}}{\xi_{i,j} + \eta_{i,j}}$$

$, i = 1, 2, \dots, K, j = 1, 2, \dots, N_i.$

When the routing probabilities at each node depend on the system configuration, the process X is not separable (i.e., it does not have a product-form solution). Consequently, the problem of determining its equilibrium distribution is intractable in general. On the other hand, the quantities of principal interest are expressed in terms of averages only; they are the steady-state mean queue sizes, L_k , and the overall average response time, W , given by

$$W = \frac{1}{\lambda} \sum_{i=1}^K \sum_{j=1}^{N_i} L_{i,j}.$$

To determine those performance measures, it is not necessary to know the joint distribution of all queue sizes; the marginal distributions of the N queues in isolation are sufficient. Unfortunately, the isolated queue processes, $\{J_k(t), t \geq 0\}$ ($k = 1, 2, \dots, N$), are not Markov. As mentioned earlier the arrival stream at any node i is not Poisson since it depends on the activity of all the previous nodes (*ad infinitum* given $\phi < 1$), this makes an exact solution of the marginal queue size distributions almost as intractable a problem as solving the joint distribution of all queue sizes. However, it is possible to obtain good approximate solutions for the marginal queue size distributions by assuming the

arrival stream at node i to be Markov-modulated Poisson. However, unlike the pipeline model presented in [Thomas and Mitrani, 1998], there is no explicit start point in this model where the arrival process at a node is known, therefore an iterative solution is employed and a further approximation is used to start the process. This iterative process and the formation of the Markov-modulated arrival processes are discussed in Sections 3 and 4.

Consider the stochastic processes $Y_{i,j}$,

$$Y_{i,j} = \{[I^*(t), J_{i,j}(t)], t \geq 0\}$$

$, i = 1, 2, \dots, K, j = 1, 2, \dots, N_i,$

which model the joint behaviour of the configuration and the size of an individual queue i, j , where $I^*(t)$ indicates the current mode of the Markov-modulated arrival process (MMPP). The state space of $Y_{i,j}$ is infinite in one dimension only, which simplifies the solution considerably and makes it tractable for reasonably large values of the number of modes in the MMPP, I_{max} . The important observation here is that, with the assumption of a Markov-modulated Poisson process, $Y_{i,j}$ is an irreducible Markov process, for every i, j . This is because the arrivals into, and departures from queue i, j during a small interval $(t, t + \Delta t)$ depend only on the approximated system configuration and the size of queue i, j at time t , and not on the sizes of the other queues. As mentioned earlier, without the approximation of the arrival stream to a Markov-modulated Poisson process, this statement would not be true, since a job only arrives at node $i + 1$ after successfully completing service at node i , therefore making the queue size at any node dependent on all previous nodes of service. It is then necessary to find the equilibrium distribution of $Y_{i,j}$:

$$p_{i,j}(x, y) = \lim_{t \rightarrow \infty} P[I^*(t) = x, J_{i,j}(t) = y]$$

$, x = 0, 1, \dots, I_{max} - 1, y = 0, 1, \dots$

Given the probabilities $p_{i,j}(x, y)$, the average size of queue i, j is obtained from

$$L_{i,j} = \sum_{y=1}^{\infty} y \sum_{x=0}^{I_{max}-1} p_{i,j}(x, y)$$

There are three established approaches to solving systems of this kind, matrix geometric methods [Neuts, 1981], solution by generating functions and spectral expansion [Mitrani and Chakka, 1995]. We have employed spectral expansion due primarily to familiarity with this technique and this choice is somewhat arbitrary. The use of spectral expansion has some issues

regarding stability with respect to deriving eigenvalues, although the method is well known, elegant and efficient. Since it appears in detail elsewhere we do not present the application of the spectral method here and the interested reader is directed to that earlier work [Mitrani and Chakka, 1995].

3 APPROXIMATION USING AN MMPP

In the previous section it was stated that the arrivals at node i could be approximated by a Markov-modulated arrival process. In the study of the simpler pipeline model [Thomas and Mitrani, 1998] a comparison was made between a simple Poisson approximation and an MMPP where the modes correspond to the operational state at the preceding node, i.e. σ_{i-1} , thus the MMPP used at node i will have $2^{N_{i-1}}$ modes. In each mode the arrival rate is calculated as the sum of the departure streams in that operational state:

$$\sum_{j=1}^{N_i} \mu_{i,j} (p_{\sigma_i} - p_{i,j,0}(\sigma_i))$$

If this model was specified using Markovian process algebra and the technique described in [Thomas et al, 2003] applied then there would in fact be $4^{N_{i-1}}$ modes in the MMPP. This is because the separation of *actions* in process algebra gives rise to separate modes not only when each server is operational or not, but also whether its queue is empty or not. Hence the mode in the MMPP is described by the superset of the pairs (o_i, e_i) , where $o_i \in \{0, 1\}$ indicates whether the server i is operational or not and $e_i \in \{0, 1\}$ indicates whether the queue at i is empty or not. The arrival rate in each mode is the sum of service rates for each server that is both working and has a non-empty queue. The transitions between modes in this case are somewhat more complex. Clearly the transitions arising from a change in operational state are the same as previously, and the transition rate from empty to non-empty is simply the average arrival rate into that queue in that operational state. The transition from non-empty to empty is calculated as the service rate multiplied by the probability that there is exactly one job in the queue given that it is non-empty. Clearly this requires knowing the probabilities $p_{i-1,j}(\sigma_{i-1}, 0)$ and $p_{i-1,j}(\sigma_{i-1}, 1)$ for all j and σ_{i-1} , except $\sigma_i = \emptyset$.¹ The number of modes in the MMPP clearly has implications for the amount of work required to solve the model. Therefore a move from $2^{N_{i-1}}$ to $4^{N_{i-1}}$ is clearly not

¹Clearly the predecessor of node 1 is node K , hence we interpret $1 - 1$ as K in this instance.

desirable when N_{i-1} is large except if there is a significant increase in accuracy of the approximation.

4 ITERATIVE SOLUTION

If the probability of leaving the system after the last node is not certain, $\phi < 1$, then there is no node for which the input process is entirely known. To address this problem an iterative approach is adopted (from [Thomas et al, 2003]) to tackle this problem as follows.

1. Calculate the rate of an equivalent Poisson arrival process at node 1.
2. Solve to find the approximate marginal queue size probabilities at node 1.
3. Use the calculated queue empty probabilities to generate a Markov-modulated arrival process at the next node.
4. Use this to calculate the approximate marginal queue size probabilities at the next node.
5. Repeat steps 3 and 4 until convergence criterion is satisfied (or abandon).

The convergence criterion employed is that the same MMPP is calculated twice in succession (to some number of decimal places) for any given node. The equivalent Poisson stream at node 1 is easily shown to have rate $\frac{1}{\phi}\lambda$ in steady state since there is no job loss.

5 NUMERICAL RESULTS

Figures 2 and 3 show illustrate that the MMPP approximation based on the operative state at the preceding node is generally very successful at predicting average response time except when the periods of operation and inoperation were very long (Figure 3) and at high load (Figure 2). One of the reasons for this inaccuracy is that during long inoperative periods the number of jobs at the preceding node will become much larger (all the servers may be broken or sufficient such that the remainder are saturated) and so on repair there will be a period of continuous service before steady state behaviour can eventually be resumed. The more complex MMPP approximation including states where the preceding queues are empty are similar in accuracy to the MMPP case. The advantage of these two methods of approximation is that they appear to offer an upper

and lower bound respectively, but only in the absence of feedback.

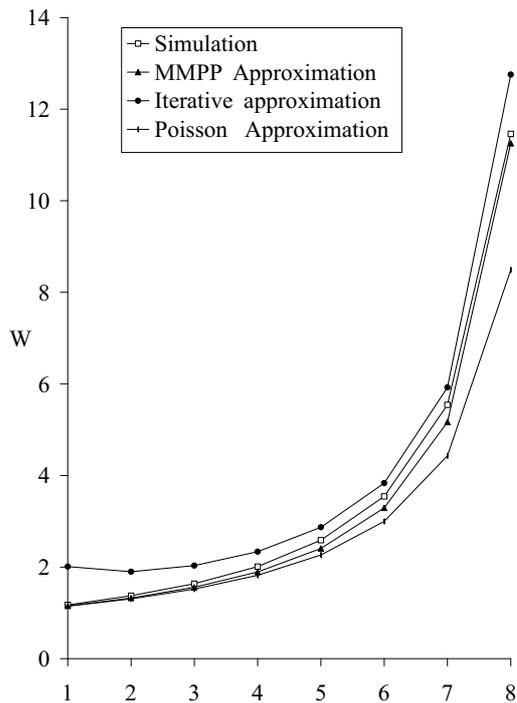


Figure 2. Average response time at node 2 against arrival rate; $\mu_i = 10, \eta_i = 0.1, \xi_i = 0.01, \phi = 1$

The divergence between a simple Poisson approximation and the MMPP approximation becomes more exaggerated when the feedback probability ϕ is decreased. Under these conditions the traffic at each node becomes increasingly less Poisson and the accuracy of both approximations diminishes dramatically. Unlike the pipeline case these approximations give an over estimate of the average response time in the presence of feedback. This is due to the tightly coupled nature of this model. The key point to observe is when the bulk of the arrivals occur into a queue, not how bursty they are. If node 1 breaks down then all arrivals into node 2 are blocked and so (after a number of services at node 2) is the feedback. This means that during most of the breakdown period the arrivals into node 1 are just the external arrivals. The same happens during breakdowns at node 2, although obviously here the external jobs have to pass through node 1 before reaching node 2. Thus in both cases relatively few jobs arrive at a node when it is broken so the queue sizes don't grow too much. Contrast this to the Poisson approximation; here jobs are assumed to arrive at a node regardless of its state, a constant rate of λ/ϕ . Hence in the feedback case the

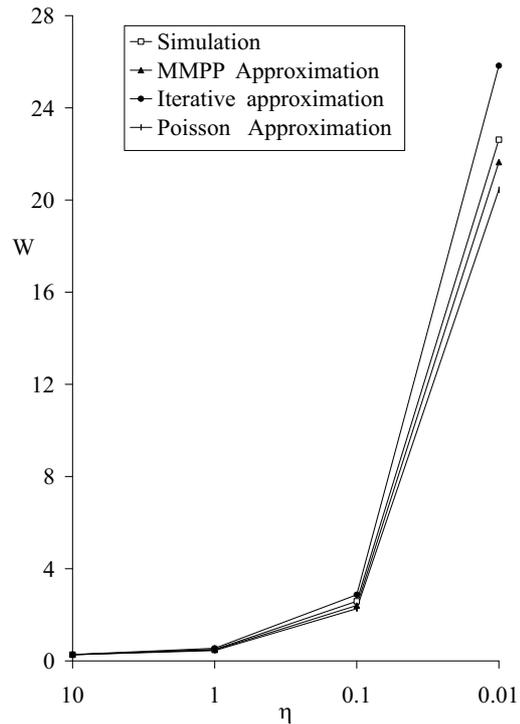


Figure 3. Average response time at node 2 against repair rate; $\mu_i = 10, \lambda = 5, \xi_i = \eta_i/10, \phi = 1$

Poisson approximation ceases to be a lower bound. The more complex approximation captures some of this behaviour, but it's still fairly crude.

Figures 4 and 5 show the behaviour of the same two node model where $\phi = 0.5$. In Figure 4 the average system response time is shown when the external arrival rate is varied. Note that the effective rate of arrivals is in fact twice the rate given on average, but the arrivals are made much more bursty by the occurrence of breakdowns. At low load the iterative approach clearly gives the best approximation, but this becomes less accurate as load increases. At even higher load, nearer saturation, the Poisson approximation becomes more accurate. This is due to the fact that the queue lengths are generally so large that the queues rarely empty during a breakdown at the other node, and so the "steady" arrivals of the Poisson approximation are more realistic.

6 CONCLUSIONS

The method proposed here seeks to extend the pipeline model to include a feedback which is incorporated into

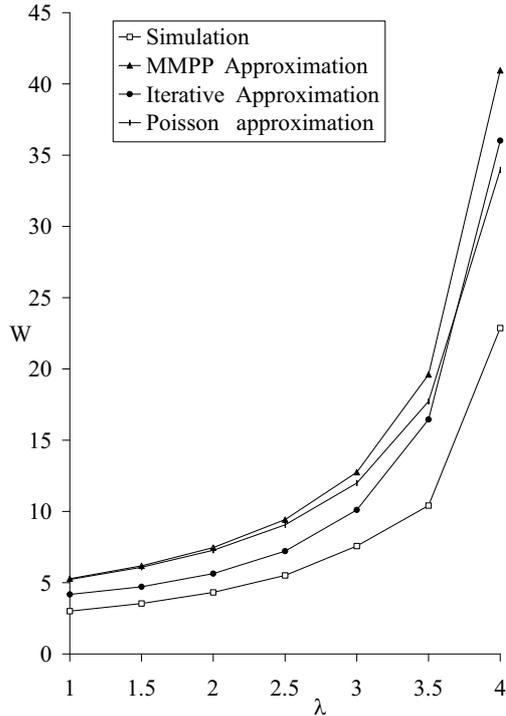


Figure 4. Average system response time against arrival rate; $\mu_i = 10, \eta_i = 0.1, \xi_i = 0.01, \phi = 0.5$

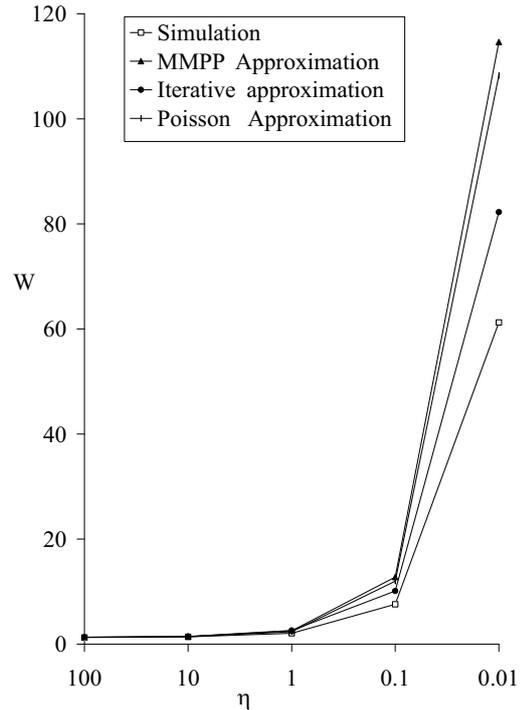


Figure 5. Average system response time against repair rate; $\mu_i = 10, \lambda = 3, \xi_i = \eta_i/10, \phi = 0.5$

the solution method using an iterative approach. The class of models that it is possible to consider using this method includes more general network structures, although it is evaluated here with a simple loop. Simple approximations work well under most conditions but become increasingly less accurate as load and inoperative periods increase. These approximations form a lower bound to the exact solution when there is no feedback, but become overly pessimistic when feedback exists. The more complex approximation with iterative solution method performs somewhat better under these conditions, but there is still considerable room for improvement, particularly at high load. Incorporating a burst of services following repair may increase the accuracy of this method, this remains an area of continuing investigation.

REFERENCES

Harrison P. Thornley D. and Zatschler H. 2002, Geometrically Batched Networks, in: *Proceedings of the Seventeenth International Symposium On Computer and Information Sciences*.

Mitrani I. and Chakka R. 1995, Spectral Expansion Solution for a Class of Markov Models: Application and Comparison with the Matrix-Geometric Method, *Performance Evaluation*, 23, pp. 241-260.

Mitrani I. and Wright P. 1994, Routing in the Presence of Breakdowns, *Performance Evaluation* 20, pp. 151-164.

Neuts M. 1981, *Matrix geometric solutions in stochastic models*, John Hopkins Univ Press.

Thomas N. and Mitrani I. 1995, Routing among different servers, in: F. Baccelli, A. Jean-Marie and I. Mitrani, eds., *Quantitative Methods in Parallel Systems*, Springer-Verlag.

Thomas N. and Mitrani I. 1998, Approximate solution of a pipeline with server vacations, in: *Proceedings of 12th European Simulation Multiconference*.

Thomas N. Bradley J. and Thornley D. 2003, An approximate solution of PEPA models using component substitution, *IEE Proceedings Computers and Digital Techniques*.

FUNCTIONAL MODELLING AND PERFORMANCE EVALUATION FOR TWO CLASS DIFFSERV ROUTER USING STOCHASTIC PROCESS ALGEBRA

ABDELMALEK BENZEKRI and OSMAN SALEM

*Institut de recherche en informatique de Toulouse,
Université Paul Sabatier,
118 Route de Narbonne - 31062 Toulouse Cedex 04 - France
Téléphone: +33 05 61 55 60 86 - Télécopie: +33 05 61 52 14 58
E-mail: {benzekri, osman}@irit.fr*

Abstract: This paper describes the use of stochastic process algebra to model and to evaluate the performance of a two class DiffServ router. This specification is done by means a set of powerful operators of Extended Markovian Process Algebra (EMPA) language, and then studied from the functional and the performance point of view.

Keywords: Stochastic Process Algebra; Markov Models; Performance Evaluation; DiffServ.

1. INTRODUCTION

Network devices can be described using classical process algebra which provides a means for constructing an abstract model of the device in question [Bernardo, 1998]. This model is used only to establish the correct functional behaviour by deriving qualitative properties such as freedom from deadlock or livelock [Benzekri, 2002]. Performance evaluation of the model was in a separate phase, after the fully design and implementation of the model. Consequently, if the performance is detected to be poor, the model will be redesigned with negative consequences for both design cost and time lost, where the need of integrating the performance analysis into the design process.

Stochastic Process Algebra (SPA) has been developed for this purpose. It is a formal specification technique which extends classical process algebras via the inclusion of timing information by using random variables in the generated models, in order to express the durations of an activity [Benzekri, 2002; Brinksma and Hermanns, 2001]. Once the model has been defined and parameterized, it can be used to investigate numerically the performance parameters.

Designing Diffserv [Nichols and Blake, 1998] routers using SPA is a complicated task because there are many factors to be considered such as penalty policy of malicious traffic inside a class. This policy may fluctuate from delaying to dropping packets from this flow. In contrast, formal specification can be a valuable aid to routers designers as it allows a range of options for configuration to be explored in a precise setting, such as policy requirements which may be clarified during the performance evaluation of a router model, because it becomes evident what

information about the state of the model is required to ensure that it operates effectively. For example, it would be possible for the designer to demonstrate that under any constraints, a minimum threshold for throughput and delay may be satisfied for such class of traffic.

This paper is organized as follows. In section 2 we recall the syntax and semantics of Extended Markovian Process Algebra EMPA. In section 3 we recall the principle of the DiffServ technology. In section 4 we give the specification of a DiffServ router. In section 5 we analyze the performance of this model by using EMPA algebraic reward method and the CTMC diagram derived from model specification. Finally, conclusions work is presented.

2. STOCHASTIC PROCESS ALGEBRA

Stochastic Process Algebras are formal descriptions techniques used to describe the functionality of concurrent and distributed systems and to analyze their related performance [Benzekri, 2002; Brinksma and Hermanns, 2001]. Several SPA languages have been appeared in the literature, these include PEPA [Hillston, 1996], TIPP [Herzog, 1993], EMPA [Bernardo, 1998]. These languages have been introduced as an extension to classical process algebras like CCS [Milner, 1989] and CSP [Hoare, 1985]. They are abstract languages constructed from a small set of powerful operators where it is possible to construct algebraic models whose key features are: compositionality (which allows the designer to build a complex model from smaller ones by means of languages operators, and to study the behaviour of each component separately), and abstraction (which allows the internal details of a system description to

be hidden from an external observer at analysis time). In these languages, systems are modeled as a collection of entities, called agents or processes, which execute actions. These actions are the building blocks of these languages and they are used to describe sequential behaviours which may run concurrently by synchronizations or by communications between them.

These languages propose the same approach to performance modeling: a random variable is associated with each action, representing its duration. This random variable is assumed to be exponentially distributed and this leads to a clear relationship between the process algebra model and a Continuous Time Markov Chains (CTMC). Via this underlying CTMC derived from the model semantic description [Benzekri, 2002], different types of analysis may be performed, like steady-state and transient probability distribution. This analysis is done through the compilation of the infinitesimal generator matrix of the Markov diagram.

In this paper, we will use the Extended Markovian Process Algebra (EMPA) language [Bernardo, 1998] which is supported by a tool called TwoTowers. EMPA is inspired and developed on the basis of PEPA (Performance Evaluation Process Algebra [Hillston, 1996]) and TIPP (TImed Processes and Performability evaluation) languages [Herzog, 1993]. It extends these languages by including three different kinds of actions: exponentially timed actions, passive actions and prioritized weighted immediate actions. In addition to this reason, EMPA allows one to specify performance measures with the algebraic specifications of the system through atomic rewards attached to states and transitions of the Markov chains (MC for short). This leads to an automatic derivation of performance measures and may avoid a full scan of the CTMC diagram. The syntax of EMPA can be summarized by the following expression:

$$P = 0 \mid \langle a, \lambda \rangle . P \mid \langle a, \infty_{L,W} \rangle . P \mid \langle a, * \rangle . P \mid P/L \mid P[\phi] \mid P + P \mid P \parallel_s P \mid A$$

Since the deadlock term "0", the prefix operator " $\langle a, \lambda \rangle .$ ", the functional abstraction operator " $_/L$ ", the functional relabeling operator " $_{[\phi]}$ ", the alternative choice operator " $+_$ ", the cooperation operator " \parallel_s " and the constant operator are the same operators used in classical process algebras. Due to lack of space, the reader is referred to [Bernardo, 1998] for an extensive presentation of EMPA syntax and semantics.

In EMPA, every activity is represented by $\langle a, \lambda \rangle$ which means the execution of action "a" after exponential distributed delay with rate " λ " (denoted by $F(t)=1-e^{-\lambda t}$). An immediate action is represented

with rate $\lambda = \infty$ or " $\langle a, \infty_{L,W} \rangle$ ", where L is used to express the priority level and W is used for the probability weight. In some cases, the rate of an action is outside the control of this component, such actions are carried out jointly with another component in order to model activities waiting for synchronization, so this component is playing a passive role and is recorded by the distinguished symbol "*".

The choice in the alternative composition operator " $+_$ " is governed by the race policy, where the action with least duration will be executed. In this situation, immediate actions take precedence over exponentially distributed actions and over other immediate actions having small priority level with respect to their levels. If two immediate actions have the same priority level, they will be executed according to the probability associated with each one.

3. THE DIFFSERV ROUTER

Differentiated services (DiffServ) [Nichols and Blake 1998] is a set of technologies which allow network service providers to offer services with different kinds of network quality of service (QoS) to different customers and their traffic streams, depending to a contract (Service Level Agreement or SLA) between them.

DiffServ work by dividing traffic into many classes by marking a field in the IP packet header, called the Differentiated Services Code Point (DSCP) field. Its value depends on the customer profile and the traffic requirement. Network elements serve these classes with different priorities with respect to the DSCP field content. Applications requiring low loss, low latency, low jitter and assured bandwidth service generally send data as expedited forwarding (EF) class packets. This class is used for loss and delay sensitive applications such as voice over IP (VoIP). Assured forwarding (AF) class offers a lower priority service from the previous one (EF), and itself is subdivided into four subclasses and each of these four classes is also divided into three subclasses (gold, silver, bronze) [Nichols and Blake 1998]. Generally AF carries best effort TCP data, such as HTTP and FTP traffic applications.

Like we have seen that a DiffServ router has a large number of classes defined, but the most essential use of DiffServ is to provide support for the two most common applications: voice and video traffic with high priority level, and best effort data (TCP) with low priority level. This is why in the rest of this paper, we will be concerned only with the modelling of such a two classes router and we will denote these classes by H (high priority level) and L (low priority level), instead of modeling all classes in order to prevent a huge number of state and the state space

explosion problem when analyzing the model by existing tools.

The DiffServ router is composed from a classifier and a traffic conditioner. Traffic conditioners may contain meters, markers, droppers and shapers. We must notice that some of these blocks may be aggregated in another block or may be omitted. For example, in the case where no traffic profile is in effect, packets may only pass through a classifier and a marker, and in the case of core routers, marker may be omitted because packets were coded at the ingress router. Readers interested about the DiffServ technology can refer to [Nichols and Blake, 1998; Blake, 1998].

4. THE SPECIFICATION

We take advantage from the compositional feature of SPA in order to model the DiffServ router which appears in figure 1. This feature allows us to deal with five entities: classifier, marker, meter, dropper and priority queuing.

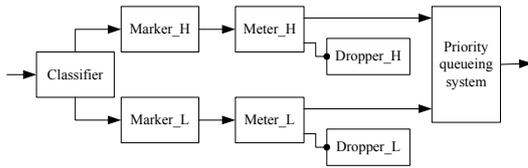


Figure 1. An ingress DiffServ router

In this specification, we suppose that the customer has a service level specification (SLS) which specifies 2 service levels, to be identified to the provider by DSCP High and DSCP Low. Each components of this DiffServ router model [Bernet and Blake, 2002] can be specified as follows:

Classifier: It takes a single traffic stream as input and generates N logically separated traffic streams as output. Figure 2 show a classifier that separates input traffic into one of two output streams based on matching filters:

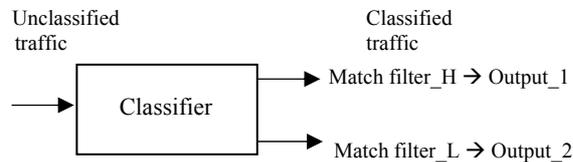


Figure 2. Classifier

The specification of this agent using EMPA is:

```
Classifier=
  <packet_arrival,λ>. <check_pkt_header,θ>.
  <classify,∞₁₁>. (<send_message_H,λ₁>.Classifier
  + <send_message_L,λ₂>.Classifier);
```

Marker: The marker sets the DS field of each packet received from the classifier to a particular code-point

(e.g., DSCP). It can be represented logically by a box with one input one output like appear in figure 3.

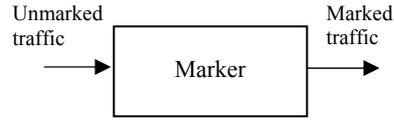


Figure 3. Marker

In our model, two markers are needed, one for the high priority level traffic and the other for the low priority level traffic. Their specification is the following:

```
Marker_H = <send_message_H,*>. <mark_DSCP_H,α>.
  <send_to_meter_H,∞₁₁>.Marker_H;
Marker_L = <send_message_L,*>. <mark_DSCP_L,α>.
  <send_to_meter_L,∞₁₁>.Marker_L;
```

Meter: It is used to monitor the traffic stream and sends malicious packets to the dropper agent, in order to prevent high level traffic from monopolizing the network resource. Figure 4 illustrates a simple meter with two levels of conformance. It will measure the rate of each traffic to determine its conformance. So, if the packet is judged conformed, then it will be sent to the priority queuing system in order to be served (forwarded to next hop), else the packet will be sent to the dropper. This agent can be specified by the following:

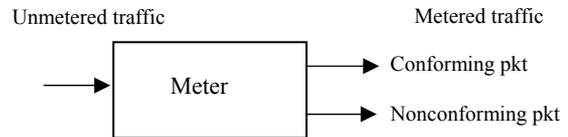


Figure 4. Meter

```
Meter0_H = <send_to_meter_H,*>. <start_timer,∞₁₁>.
  <arr_H,λ_H>.Meter1_H;
Meter1_H = <send_to_meter_H,*>.
  <get_current_time,∞₁₁>.
  <compute_elapsed_time,∞₁₁>.
  (<time_elapsed,∞₁₁>.
  <average_conform,∞₁₁>. <reset_timer,∞₁₁>.
  <arr_H,λ_H>.Meter1_H
  + <time_not_elapsed,∞₁₁>.
  <avrg_not_conform,∞₁₁>.
  <send_to_dropper_H,γ>.Dropper_H
  + <timeout,η>. Meter0_H);
Meter0_L = <send_to_meter_L,*>. <start_timer,∞₁₁>.
  <arr_L,λ_L>.Meter1_L;
Meter1_L = <send_to_meter_L,*>.
  <get_current_time,∞₁₁>.
  <compute_elapsed_time,∞₁₁>.
  (<time_elapsed,∞₁₁>.
  <average_conform,∞₁₁>. <reset_timer, ∞₁₁>.
  <arr_L,λ_L>.Meter1_L
  + <time_not_elapsed,∞₁₁>.
  <avrg_not_conform,∞₁₁>.
  <send_to_dropper_L,γ>.Dropper_L
```

+ <timeout, η >.Meter0_L);

Dropper: The dropper discards some or all malicious packets in a traffic stream according to the service provider policy. The dropper can be implemented as a special case of shaper by setting the buffer size to zero. It can be represented logically by a box with one input one output and its specification is the following:

```
Dropper_H = <send_to_dropper,*>.<discard, $\infty$ 11>.
Meter1_H
Dropper_L = <send_to_dropper,*>.<discard, $\infty$ 11>.
Meter1_L
```

Priority queueing: The final agent is the queue where packets wait before being served (forwarded to next hop). We have taken a preemptive queue (e.g., arriving of high priority packet will interrupt the service of low priority packet already in service phase) with priority inter-arrival policy in the sense that if the queue is full and a high priority packet arrives, it will drop a low priority packet (if the queue contains at least one) in order to accept the high priority packet, but if it contains only a high priority packet, the arrival packet will be lost (dropped). The specification of this queue (M/M/1/N) [Thomas and Hillston, 1997] is the following:

```
Queue0,0 = <arrH,*>. Queue1,0 + <arrL,*>. Queue0,1;
Queuei,0 = <arrH,*>. Queuei+1,0 + <arrL,*>. Queuei,1
+ <deliverH,*>. Queuei-1,0; (if 0<i<N-1)
Queue0,j = <arrH,*>. Queue1,j + <arrL,*>. Queue0,j+1
+ <deliverL,*>. Queue0,j-1; (if 0<j<N-1)
Queuei,j = <arrH,*>. Queuei+1,j + <arrL,*>. Queuei,j+1
+ <deliverH,*>. Queuei-1,j
+ <deliverL,*>. Queuei,j-1;
(if i,j>0 and i+j<N-1)
QueueN-1,0 = <deliverH,*>. QueueN-2,0;
Queue0,N-1 = <deliverL,*>. Queue0,N-2
+ <arrH,*>.<looseL, $\infty$ 2,1>. Queue1,N-2;
Queuei,j = <deliverH,*>. Queuei-1,j
+ <deliverL,*>. Queuei,j-1
+ <arrH,*>.<looseL, $\infty$ 2,1>. Queuei+1,j-1;
(if i,j>0 and i+j = N-1)
Pre_Server = <deliverH, $\infty$ 2,1>.<serve, $\mu$ >.Pre_Server
+ <deliverL, $\infty$ 1,1>.ServePLLow;
ServePLLow = <serve, $\mu$ >.Pre_Server
+ <deliverH, $\infty$ 2,1>.<serve, $\mu$ >. ServePLLow;
```

In order to obtain the complete DiffServ router specification, the individual agents described above need to be composed like in the following expression:

```
DiffServ = Classifier ||T Markers ||S Meters ||M
Droppers ||Arr Queue00 ||Del Pre_emp_Server
Markers = Marker_L || Marker_H
Meters = Meter0_H || Meter0_L
Droppers = Dropper_L || Dropper_H
T = {send_msgH, send_msgL};
```

```
S = {send_to_meterH, send_to_meterL};
M = {send_to_dropperH, send_to_dropperL};
Arr = {arrH, arrL};
Del = {deliverH, deliverL};
```

When the model is loaded in TwoTowers [Bernardo, 1998], its descriptions are syntactically and semantically analyzed using a parser for detecting errors, then TwoTowers will find all possible states and transitions or in another expression the labelled transition diagram.

5. PERFORMANCE ANALYSIS

TwoTowers will give us the steady state and the transient state distribution probability vector. So given the CTMC diagram and the value of the probability distribution in steady and transient states, we can evaluate the performance of the system by using queueing system theory. For example, the throughput which is given by the service rate multiplied by the stationary probability of being in a state where service action can be provided is given by the following formula:

$$T = \sum_{i=2}^N \mu \cdot \pi_i$$

The utilization rate was defined to be the percentage of time the router spent in doing useful work by the fraction of time, and which is the sum of the stationary probabilities of states where there is at least one packet in the system. It is given by the following formula:

$$U = \sum_{i=2}^N (1 \cdot \pi_i) = \sum_{i=2}^N \pi_i$$

The vector distribution probability π_i for all states is given by TwoTowers and the utilisation rate can be calculated by a simple addition.

The Markovian analyzer implemented in TwoTowers allow an automatic derivation of performance model and may allow us to avoid the full scan to the CTMC diagram, which will be exceedingly expensive, especially if we have a large number of states. The performance aspects of a system model in EMPA, can be taken into account in the early stages of its design (with algebra description), where performance measures can be specified by attaching a yield reward y_i to every state i , which expresses the rate at which reward is accumulated at state i , and by attaching a bonus reward $b_{i,j}$ to every transition from state i to state j , which expresses the instantaneous gain due to the execution of the transition from state i to state j . Readers interested about yield and bonus rewards can refer to [Bernardo, 1997]. Actions with reward will be specified by $\langle a,r,y,b \rangle$ instead of $\langle a,r \rangle$, and the desired stationary performance measure can be computed in EMPA according to the following formula:

$$\sum_{i=1}^N y_i \cdot \pi_i + \sum_{i=1}^N \sum_{j=1}^N b_{ij} \cdot \pi_i \cdot q_{ij}$$

Many performance measures can be obtained using this formula, for example: if we want to compute the throughput, we must replace every action of the form $\langle \text{serve}, \mu \rangle$ with $\langle \text{serve}, \mu, \mu, 0 \rangle$ (e.g., $y_i = \mu$ and $b_{ij} = 0$) for obtaining the following equation:

$$\sum_{i=1}^N y_i \cdot \pi_i + \sum_{i=1}^N \sum_{j=1}^N b_{ij} \cdot \pi_i \cdot q_{ij} = \sum_{i=2}^N \mu_i \cdot \pi_i$$

EMPA will take into account all states that provide the action "serve" and will assign a reward to them. The first state is where no packet in the router and this is why it can not provide the action "serve".

As a performance measure, we have computed the throughput and the router utilization by using the reward technique of EMPA. This is done by replacing every action $\langle \text{serve}, \mu \rangle$ by $\langle \text{serve}, \mu, \mu, 0 \rangle$ in our semantics model for obtaining the throughput, and by replacing every action $\langle \text{serve}, \mu \rangle$ by $\langle \text{serve}, \mu, 1, 0 \rangle$ in order to obtain the utilization rate. In contrast, the algebra based method (reward technique) fails to determine the mean number and the mean waiting time of packets for each class due to the additivity assumption of transition labeled with "serve" action, and values for these performance aspects were calculated by a manual full scan to the transformed specification (CTMC diagram) and by using the probability distribution vector given by TwoTowers.

The mean number of packet in the system can be obtained by using the following formula:

$$\text{The mean number of packet} = \sum_{i=2}^N i \cdot \pi_i$$

And the mean packet delay (MPD) for each class is found by using Little's law:

$$\text{MPD} = \frac{\text{Mean system size}}{\text{Mean packet arrival rate}}$$

$$\text{Mean packet arrival rate} = \sum_{i=1}^N \lambda_i \cdot \pi_i$$

Where λ_i take the value of λ_H for packet with high priority and λ_L for packet with low priority.

The throughput was 2.38 packet/s, the utilisation rate was 33.34% and the mean waiting time was 1.12s for a packet in class high and 2.48s for packet in class low. These unacceptable results lead us to a set of experiment in order to detect the effect of each component at its performance. We begin by examine the effect of changing the rate (speed) of the marker at the system performance.

Figure 5.a shows that the throughput increases by increasing marker speed but reaches a threshold

afterwards there is no effect of increasing its speed at the system performance, and this effect can be explained by the limited speed of the classifier. However the utilisation rate of the marker decreases significantly by the fact of speeding the marker, because packets will spend a less time before being forwarded to next stage. The mean packets waiting time decreases slightly when we decrease the rate of this component because packets spend less time in this component.

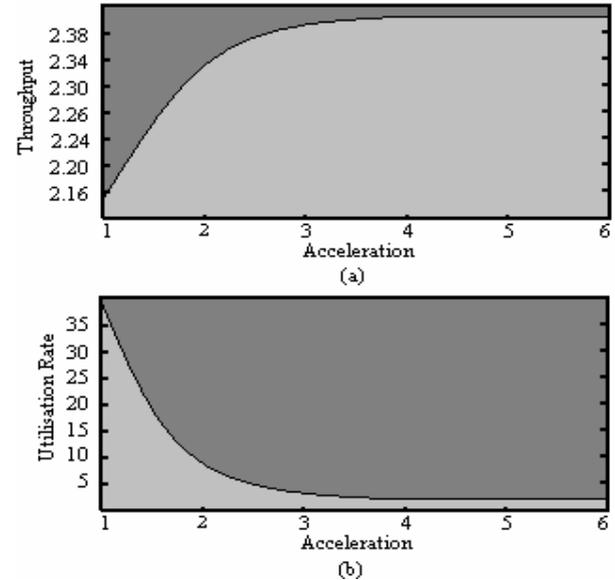


Figure 5. Effect of speeding up the marker

We have experimented the effect of speeding up the meter (increasing its rate) at the performance model. Like expected the throughput and the utilisation rate of the model increase and reach quickly a maximum threshold value (curves variation are similar to those in Figure 5).

The same experiment was repeated for the classifier and the queueing server. We have found that there was little profit from speeding up any of these components apart at the model performance.

The result of these experiments motivate us to another set of experiments in order to investigate the effect of speeding up many components at the system performance, because every time we have increased the rate of a component we have found: the throughput of the system increases, the utilisation rate fluctuate, and the mean packet delay decreases for each class but still inside a specific margin.

Figure 6 shows the result obtained by speeding up the marker and the queueing server, it can be seen from this figure that the increase of throughput is not at the expense of utilisation rate like we have seen when speeding up one component alone.

We discover from these experiments that we can use relatively a slow classifier with no big influence at

the system throughput in contrast like we have been thought. This result can be interpreted by the time that packets spend in other components especially in the server queueing.

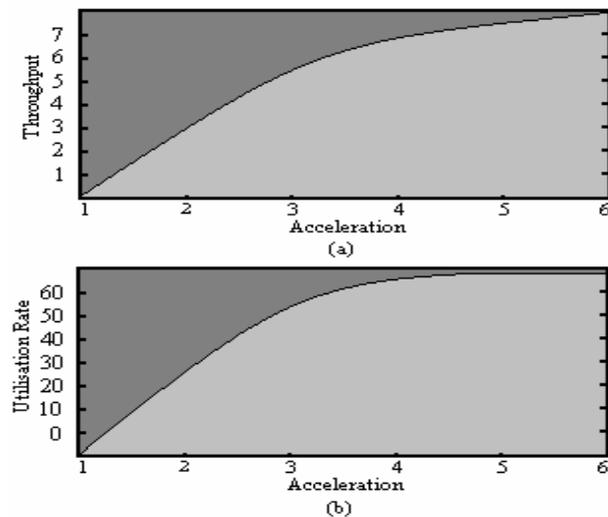


Figure 6. Effect of speeding up the marker and the server

6. CONCLUSIONS

The main aim of this work was to present a simple DiffServ router model and to analyze its functional and performance properties using stochastic process algebra. In order to achieve that, we begin by an algebraic description of this router using stochastic process algebra then we use EMPA tool (TwoTowers) for analyzing and detecting a misbehavioural functional error such as freedom from deadlock. After qualitative verification, a set of experiment has been done in order to detect the effect of each component at the performance of this model. Fortunately, the reward-based method in EMPA provides an automatic derivation for some performance aspect in our model (like throughput, utilisation rate...), but unfortunately not all. Therefore, a full scan to the CTMC diagram derived from the EMPA supported tool is necessary.

These experiments lead us to discover some interesting information about which component limits the throughput and other performance aspect. It demonstrates that we can use a relatively slow component in the model with no significant difference in the throughput in contrast to that it has been thought originally.

The coexistence of three kinds of actions in EMPA and especially the prioritized weighted action was a great characteristic because these actions are not taken in account in the performance semantic model (CTMC diagram) and this aids us to include only actions which are important for determining performance aspects.

REFERENCES

- Blake S. 1998, "An Architecture for Differentiated Services", RFC 2475.
- Benzekri A. 2002, "Qualitative and Quantitative Evaluation using Process Algebra", The 17th International Symposium on Computer and Information Sciences, Florida USA, Pp415-418.
- Bernardo M. 1997, "An Algebra Based Method to Associate Rewards with EMPA Terms", in Proc. of the 24th Int. Coll. on Automata, Languages and Programming (ICALP), P.Degano, Lecture Notes in Computer Science, Bologna, Pp358-368.
- Bernardo M. 1998, "A Tutorial on EMPA: A Theory of Concurrent Processes with Nondeterminism, Priorities, Probabilities and Time", Theoretical Computer Science, Pp1-54.
- Bernet Y. and Blake S. 2002, "An Informal Management Model for DiffServ Routers", RFC 3290.
- Brinksma Ed and Hermanns Holger 2001, "Process Algebra and Markov Chains", Lecture on Formal Methods and Performance Analysis, Nijmegen, Pp183-231.
- Herzog U. 1993, "TIPP: A Language for Timed Processes and Performance Evaluation", Proceedings of the First International Workshop on PA and PM, University of Edinburgh, UK.
- Hillston J. 1996, "A Compositional Approach to Performance Modelling", Cambridge University Press.
- Milner R. 1989, "Communication and Concurrency", Prentice-Hall.
- HOARE C.A.R 1985, "Communicating Sequential Processes", Prentice-Hall.
- Nichols K. and Blake S. 1998, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474.
- Thomas N. and Hillston J. 1997, "Using Markovian Process Algebra to Specify Interactions in Queueing Systems", Technical Report, University of Edinburgh, Pp151-164.

AUTHOR BIOGRAPHY



OSMAN SALEM is a first year PhD student in Université Paul Sabatier, UPS, Toulouse, France. He received his engineering diploma in June 2001 and his master degree in network and telecommunication in September 2002.

His research and major interesting areas are formal specification using stochastic process algebra, quality of service protocol engineering and performance evaluation.

Impatient service in a G-network

P. Bocharov, C. D'Apice, B. D'Auria

Department of Probability Theory and Mathematical Statistics,

Peoples' Friendship University of Russia

Department of Information Engineering and Applied Mathematics,

University of Salerno, Italy

E- mail: pbocharov@sci.pfu.edu.ru

E-mail: {dapice,dauria}@diima.unisa.it

Abstract

An open exponential queueing network with signals and impatient service is considered. Upon completion of service at a node, a positive customer passes to another node with fixed probabilities either as a positive customer or as a signal, or quits the network. Every signal is activated during a random exponentially distributed amount of time. Activated signals with fixed probabilities either move a customer from the node they arrive to another node or kill a positive customer. Each customer can be served in a node at most a random time ("patient" time) distributed exponentially. When the patient service is

finished, the customer with fixed probabilities either goes to another node or quits the network. The stationary state probabilities for such a G-network in which positive customers are processed in each node by a single server is derived in product form. The solution for an analogous symmetrical G-network in which service rate of a positive customer at each node depends on the number of positive customers in this node is expressed in product form too.

Keywords: G-networks, positive customers, impatient service and product form solution.

1 Introduction

In the last years queueing networks with "negative and positive customers, triggers, signals", called G-networks have been studied and solution in product form have been obtained [Gelenbe and Pujolle, 1998]. Positive customers are ordinary customers and they are served by the server in the normal way, instead a negative customer deletes (kills) a positive customer, or a trigger, which moves a positive customer from a node to another one; a signal combines these two kinds of customers and can act either as a negative customer or as a trigger. The analysis of this new class of queueing networks was first inspired by the study of neural networks. G-networks can be applied too in performance evaluation of computer networks, to model, for example, the effect of flow control [Gelenbe and Pujolle, 1998]. A vast review of papers concerning with G-networks is done in [Artalejo 2000].

We consider queueing networks with positive customers and signals. External arrival flows of positive customers and signals are independent Poisson processes. The service times of positive customers at each node are exponentially distributed. Upon completion of service at a node, a positive customer passes to another node with fixed probabilities either as a positive customer or as a signal, or quits the network. Every signal is activated during a random exponentially distributed amount of time. Ac-

tivated signals with fixed probabilities either move a customer from the node they arrive to another node or kill a positive customer. We assume additionally that each customer can be served in a node at most a random time ("patient" time) distributed exponentially. When the patient service is finished, the customer with fixed probabilities either goes to another node or quits the network.

A G-network with instantaneous signal activation is studied in [Gelenbe and Pujolle, 1998]. An analogous G-network with random signal activation period without impatient time is considered in [Bocharov 2002]. The stationary state distributions for the networks considered in [Gelenbe and Pujolle, 1998], [Bocharov 2002], were derived in product form. In this paper, the stationary state distribution for a G-network with random delay of signals and impatient service in which positive customers are processed by a single server is derived in product form. Moreover, the solution for an analogous symmetrical G-network in which service rate of a positive customer at each node depends on the number of positive customers in this node is expressed in product form too.

2 Mathematical model

We deal with G-networks with M nodes, positive customers and signals. Positive customers and signals arrive from outside (from node 0) according to inde-

pendent Poisson processes. We denote, respectively, with λ_{0i}^+ and λ_{0i}^- the arrival rate of external positive customers and external signals at node i .

The service of a positive customer is completed at node i with probability $\mu_i^+(k)\Delta + o(\Delta)$ in a time interval $(t, t + \Delta)$, provided that k positive customers are present at this node at instant t .

Upon completion of service at node i , a positive customer goes from node i to node j with probability p_{ij}^+ as a positive customer, and with probability p_{ij}^- as a signal. He leaves the network with probability $p_{i0} = 1 - \sum_{j=1}^M (p_{ij}^+ + p_{ij}^-)$.

Every signal is activated during a random time. A signal arriving at node i is activated in a time interval $(t, t + \Delta)$ with probability $\mu_i^-(n)\Delta + o(\Delta)$, provided that n non activated signals are present at this node at instant t .

After the completion of the activation period a signal:

- with probability q_{ij}^+ moves a positive customer from node i to node j and retains him as a positive customer (in this case, a signal acts as a trigger);
- with probability q_{ij}^- moves a positive customer from node i to node j and retains him as a signal;
- with probability q_{i0} kills a positive customer at node i and he vanishes (in this case, the signal acts as a negative customer).

If there are not positive customers at node i an activated signal at the node disappears.

The impatience time of a customer in the node i is completed in a time interval $(t, t + \Delta)$ with probability $\gamma_i(k)\Delta + o(\Delta)$, provided that k positive customers are present at this node at instant t . Then a positive customer with probability r_{ij}^+ goes from node i to node j as a positive customer, with probability r_{ij}^- as a signal, and with probability $r_{i0} = 1 - \sum_{j=1}^M (r_{ij}^+ + r_{ij}^-)$ he leaves the network.

3 Equilibrium equations

Let us denote with $P^+, P^-, Q^+, Q^-, R^+, R^-$ the matrices with elements $p_{ij}^+, p_{ij}^-, q_{ij}^+, q_{ij}^-, r_{ij}^+, r_{ij}^-$, respectively, $i, j = \overline{1, M}$, and let us set $P = P^+ + P^-$, $Q = Q^+ + Q^-$, and $R = R^+ + R^-$.

The stochastic behaviour of the queueing network under consideration can be described by an homogeneous Markov process $\{X(t), t \geq 0\}$ with the following state space:

$$\mathcal{X} = \{((k_1, n_1), \dots, (k_M, n_M)), k_i \geq 0, n_i \geq 0, i = \overline{1, M}\}.$$

The state $((k_1, n_1), (k_2, n_2), \dots, (k_M, n_M))$ means that at any instant there are k_1 positive customers and n_1 non-activated signals at node 1, k_2 customers and n_2 signals at node 2, ..., and finally, k_M customers and n_M signals at node M .

Introducing vectors $\vec{k} = (k_1, k_2, \dots, k_M)$ and $\vec{n} = (n_1, n_2, \dots, n_M)$, let us take $(\vec{k}, \vec{n}) = ((k_1, n_1), (k_2, n_2), \dots, (k_M, n_M))$. We also introduce

the vector \vec{e}_i with i -th component equal to 1 and other components equal to 0. We also use the notation $\lambda_0^+ = \sum_{i=1}^M \lambda_{0i}^+$ and $\lambda_0^- = \sum_{i=1}^M \lambda_{0i}^-$.

Let $p(\vec{k}, \vec{n})$ denote the stationary probability of the state (\vec{k}, \vec{n}) . If the stationary distribution $\{p(\vec{k}, \vec{n}), \vec{k}, \vec{n} \geq \vec{0}\}$ of the process $\{X(t), t \geq 0\}$ exists, then the following system of equilibrium equations holds:

$$\begin{aligned}
& p(\vec{k}, \vec{n})(\lambda_0^+ + \lambda_0^- + \sum_{i=1}^M \mu_i^+(k_i)(1 - p_{ii}^+) + \sum_{i=1}^M \mu_i^-(n_i)) \\
& + \sum_{i=1}^M \gamma_i(k_i)(1 - \gamma_{ii}^+) = \sum_{i=1}^M p(\vec{k} - \vec{e}_i, \vec{n})\lambda_{0i}^+ u(k_i) + \\
& \sum_{i=1}^M p(\vec{k}, \vec{n} - \vec{e}_i)\lambda_{0i}^- u(n_i) + \\
& \sum_{i=1}^M p(\vec{k} + \vec{e}_i, \vec{n})\mu_i^+(k_i + 1)p_{i0} + \\
& \sum_{i=1}^M p(\vec{k} + \vec{e}_i, \vec{n} + \vec{e}_i)\mu_i^-(n_i + 1)q_{i0} + \\
& \sum_{i=1}^M p(\vec{k} + \vec{e}_i, \vec{n})\gamma_i(k_i + 1)r_{i0} + \\
& \sum_{i=1}^M p(\vec{k}, \vec{n} + \vec{e}_i)\mu_i^-(n_i + 1)(1 - u(k_i)) + \\
& \sum_{i=1}^M \sum_{j=1, j \neq i}^M p(\vec{k} + \vec{e}_i - \vec{e}_j, \vec{n})\mu_i^+(k_i + 1)p_{ij}^+ u(k_j) + \\
& \sum_{i=1}^M \sum_{j=1}^M p(\vec{k} + \vec{e}_i, \vec{n} - \vec{e}_j)\mu_i^+(k_i + 1)p_{ij}^- u(n_j) + \\
& \sum_{i=1}^M \sum_{j=1}^M p(\vec{k} + \vec{e}_i - \vec{e}_j, \vec{n} + \vec{e}_i)\mu_i^-(n_i + 1)q_{ij}^+ u(k_j) + \\
& \sum_{i=1}^M \sum_{j=1, j \neq i}^M p(\vec{k} + \vec{e}_i, \vec{n} + \vec{e}_i - \vec{e}_j)\mu_i^-(n_i + 1)q_{ij}^- u(n_j) + \\
& \sum_{i=1}^M p(\vec{k} + \vec{e}_i, \vec{n})\mu_i^-(n_i)q_{ii}^- + \\
& \sum_{i=1}^M \sum_{j=1, j \neq i}^M p(\vec{k} + \vec{e}_i - \vec{e}_j, \vec{n})\gamma_i(k_i + 1)r_{ij}^+ u(k_j) + \\
& \sum_{i=1}^M \sum_{j=1}^M p(\vec{k} + \vec{e}_i, \vec{n} - \vec{e}_j)\gamma_i(k_i + 1)r_{ij}^- u(n_j), \\
& (\vec{k}, \vec{n}) \in \mathcal{X},
\end{aligned} \tag{1}$$

where $\mu_i^+(0) = 0$, $\mu_i^-(0) = 0$, $\gamma_i(0) = 0$ and $u(x)$ is a unit Heavyside function.

4 Solution in product form

We can not possible find the general product-form of the system of equations (1). Nevertheless, solutions for two important cases are given.

4.1 Service of positive customers by a single server

Consider a network in which positive customers are served at every node by a single server and the service time at node i is exponentially distributed with parameter μ_i^+ . Therefore

$$\mu_i^+(k_i) = u(k_i)\mu_i^+, \quad i = \overline{1, M}. \tag{2}$$

We also assume that

$$\gamma_i(k_i) = u(k_i)\gamma_i, \quad i = \overline{1, M}. \tag{3}$$

Let us introduce the following notations:

$$q_i = \frac{\lambda_i^+}{\lambda_i^- + \mu_i^+ + \gamma_i}, \quad \rho_i^-(j) = \frac{\lambda_i^-}{\mu_i^-(j)}, \quad i, j = \overline{1, M}. \tag{4}$$

$$\lambda_i^+ = \lambda_{0i}^+ + \sum_{j=1}^M q_j(\mu_j^+ p_{ji}^+ + \lambda_j^- q_{ji}^+ + \gamma_j r_{ji}^+), \quad i = \overline{1, M},$$

$$\lambda_i^- = \lambda_{0i}^- + \sum_{j=1}^M q_j(\mu_j^+ p_{ji}^- + \lambda_j^- q_{ji}^- + \gamma_j r_{ji}^-), \quad i = \overline{1, M}. \tag{5}$$

As in [1] we can prove that there exists a unique positive solution λ_i^+ , λ_i^- , $i = \overline{1, M}$ of the system of equations (5).

Besides let us denote

$$\Lambda_0 = \sum_{j=1}^M q_j \mu_j^+ p_{j0} + \sum_{j=1}^M q_j \lambda_j^- q_{j0} + \sum_{j=1}^M q_j \gamma_j r_{j0}. \quad (6)$$

From (4) - (6) we obtain

$$\begin{aligned} \Lambda_0 + \sum_{j=1}^M (\lambda_j^+ + \lambda_j^-) &= \lambda_0^+ + \lambda_0^- + \sum_{j=1}^M q_j (\mu_j^+ + \lambda_j^- + \gamma_j) = \\ &= \lambda_0^+ + \lambda_0^- + \sum_{j=1}^M \lambda_j^+. \end{aligned}$$

Therefore

$$\Lambda_0 + \sum_{j=1}^M \lambda_j^- = \lambda_0^+ + \lambda_0^-. \quad (7)$$

The following theorem holds.

Theorem 1 *If the matrices P , Q , and R are irreducible, conditions (2) and (3) hold, and a unique positive solution of equations (5) exists such that*

$$\lambda_i^+ < \lambda_i^- + \mu_i^+ + \gamma_i, \quad i = \overline{1, M},$$

$$G_i = \sum_{n_i=0}^{\infty} \prod_{j=1}^{n_i} \rho_i^-(j) < \infty, \quad i = \overline{1, M},$$

then the Markov process $\{X(t), t \geq 0\}$ is ergodic and its stationary distribution is represented in a product form as

$$p(\vec{k}, \vec{n}) = \prod_{i=1}^M p(k_i, n_i), \quad (8)$$

where

$$p(k_i, n_i) = (1 - q_i) q_i^{k_i} G_i^{-1} \prod_{j=1}^{n_i} \rho_i^-(j), \quad k_i, n_i \geq 0. \quad (9)$$

and $\prod_{j=1}^0 \equiv 1$.

Proof. The substitution of expressions (8), (9), (4) for the stationary distribution of the process $\{X(t), t \geq 0\}$ into the equilibrium system of equations (1) leads to the following equalities:

$$\begin{aligned} \lambda_0^+ + \lambda_0^- + \sum_{i=1}^M \mu_i^+ u(k_i) + \sum_{i=1}^M \mu_i^-(n_i) + \sum_{i=1}^M \gamma_i u(k_i) = \\ \sum_{i=1}^M \frac{\lambda_{0i}^+}{q_i} u(k_i) + \sum_{i=1}^M \frac{\mu_i^-(n_i)}{\lambda_i^-} \lambda_{0i}^- + \sum_{i=1}^M q_i \mu_i^+ p_{i0} + \\ \sum_{i=1}^M q_i \lambda_i^- q_{i0} + \sum_{i=1}^M q_i \gamma_i r_{i0} + \sum_{i=1}^M \lambda_i^- (1 - u(k_i)) + \\ \sum_{i=1}^M \sum_{j=1}^M \frac{q_i}{q_j} \mu_i^+ p_{ij}^+ u(k_j) + \\ \sum_{i=1}^M \sum_{j=1}^M q_i \frac{\mu_j^-(n_j)}{\lambda_j^-} \mu_i^+ p_{ij}^- + \sum_{i=1}^M \sum_{j=1}^M \frac{q_i}{q_j} \lambda_i^- q_{ij}^+ u(k_j) + \\ \sum_{i=1}^M \sum_{j=1}^M q_i \frac{\mu_j^-(n_j)}{\lambda_j^-} \lambda_i^- q_{ij}^- + \sum_{i=1}^M \sum_{j=1}^M \frac{q_i}{q_j} \gamma_i r_{ij}^+ u(k_j) + \\ \sum_{i=1}^M \sum_{j=1}^M q_i \frac{\mu_j^-(n_j)}{\lambda_j^-} \gamma_i r_{ij}^-. \end{aligned} \quad (10)$$

The latter equality takes place for all $(\vec{k}, \vec{n}) \in \mathcal{X}$.

Let us denote by

$$\begin{aligned} A = \sum_{i=1}^M \frac{\mu_i^-(n_i)}{\lambda_i^-} \lambda_{0i}^- + \sum_{i=1}^M \sum_{j=1}^M q_i \frac{\mu_j^-(n_j)}{\lambda_j^-} \mu_i^+ p_{ij}^- + \\ \sum_{i=1}^M \sum_{j=1}^M q_i \frac{\mu_j^-(n_j)}{\lambda_j^-} \lambda_i^- q_{ij}^- + \sum_{i=1}^M \sum_{j=1}^M q_i \frac{\mu_j^-(n_j)}{\lambda_j^-} \gamma_i r_{ij}^-. \end{aligned}$$

Taking into account (5) we obtain

$$A = \sum_{j=1}^M \mu_j^-(n_j). \quad (11)$$

Further let us denote by

$$\begin{aligned} B = \sum_{i=1}^M \frac{\lambda_{0i}^+}{q_i} u(k_i) + \sum_{i=1}^M \sum_{j=1}^M \frac{q_i}{q_j} \mu_i^+ p_{ij}^+ u(k_j) + \\ \sum_{i=1}^M \sum_{j=1}^M \frac{q_i}{q_j} \lambda_i^- q_{ij}^+ u(k_j) + \sum_{i=1}^M \sum_{j=1}^M \frac{q_i}{q_j} \gamma_i r_{ij}^+ u(k_j). \end{aligned}$$

After some transformations of the right part with

combination with (5) we obtain

$$B = \sum_{j=1}^M \frac{\lambda_j^- + \mu_j^+ + \gamma_j}{\lambda_j^+} [\lambda_{0j}^+ + \sum_{i=1}^M q_i (\mu_i^+ p_{ij}^+ + \lambda_i^- q_{ij}^+ + \gamma_i r_{ij}^+)] u(k_j) = \sum_{j=1}^M (\lambda_j^- + \mu_j^+ + \gamma_j) u(k_j). \quad (12)$$

Finally let us introduce

$$C = \Lambda_0 + \sum_{i=1}^M \lambda_i^- (1 - u(k_i)). \quad (13)$$

Then the right part of equalities (10) can be represented as $A + B + C$. Then we have

$$\begin{aligned} A + B + C &= \sum_{j=1}^M \mu_j^- (n_j) + \sum_{j=1}^M (\lambda_j^- + \mu_j^+ + \gamma_j) u(k_j) + \Lambda_0 + \sum_{i=1}^M \lambda_i^- - \sum_{i=1}^M \lambda_i^- u(k_i) = \\ &= \sum_{j=1}^M \mu_j^- (n_j) + \sum_{j=1}^M (\mu_j^+ + \gamma_j) u(k_j) + \Lambda_0 + \sum_{i=1}^M \lambda_i^- = \\ &= \lambda_0^+ + \lambda_0^- + \sum_{j=1}^M \mu_j^- (n_j) + \sum_{j=1}^M (\mu_j^+ + \gamma_j) u(k_j). \end{aligned}$$

This coincides with the left part of equalities (10).

Thus the substitution of (8), (9) into the system of equations (1) - (3) leads to a system of identities for all $(\vec{k}, \vec{n}) \in \mathcal{X}$. Under the assumptions of the theorem the expressions (8), (9) determine a positive solution of the equilibrium system of equations (1) - (3) and this solution is bounded. Moreover under theorem assumptions the process $\{X(t), t \geq 0\}$ is irreducible. Therefore, according to Foster's theorem the process is ergodic and the relations (8), (9) give us its unique stationary distribution. Thus, the theorem is proved. ■

4.2 Symmetrical network

We consider the network described in the section 2 with $(i, j = \overline{1, M})$

$$p_{ij}^+ = q_{ij}^+ = r_{ij}^+, p_{ij}^- = q_{ij}^- = r_{ij}^-, p_{i0} = q_{i0} = r_{i0}. \quad (14)$$

It is convenient to call a queueing network under these conditions as a symmetrical network.

Let us introduce the following notations:

$$q_i(j) = \frac{\lambda_i^+}{\lambda_i^- + \mu_i^+(j) + \gamma_i(j)}, \rho^-(j) = \frac{\lambda_i^-}{\mu_i^-(j)}, \quad i, j = \overline{1, M}. \quad (15)$$

$$\begin{aligned} \lambda_i^+ &= \lambda_{0i}^+ + \sum_{j=1}^M \lambda_j^+ p_{ji}^+, \quad i = \overline{1, M}, \\ \lambda_i^- &= \lambda_{0i}^- + \sum_{j=1}^M \lambda_j^+ p_{ji}^-, \quad i = \overline{1, M}. \end{aligned} \quad (16)$$

If the matrix P is irreducible, the system (16) has a unique positive solution for $\lambda_i^+, \lambda_i^-, i = \overline{1, M}$.

Let us denote

$$\Lambda_0 = \sum_{i=1}^M \lambda_i^+ p_{i0}. \quad (17)$$

From (16) and (17) we obtain

$$\Lambda_0 + \sum_{j=1}^M (\lambda_j^+ + \lambda_j^-) = \lambda_0^+ + \lambda_0^- + \sum_{j=1}^M \lambda_j^+.$$

This yields

$$\Lambda_0 + \sum_{j=1}^M \lambda_j^+ = \lambda_0^+ + \lambda_0^-. \quad (18)$$

The relation (18) formally coincides with relation (7) obtained for the case of single-server processing of positive customers but the values of λ_i^+ and λ_i^- for

the symmetrical network are determined from another system of equations which is a linear one.

Theorem 2 *If matrix P is irreducible and the following conditions hold ($i = \overline{1, M}$):*

$$F_i = \sum_{k_i=0}^{\infty} \prod_{j=1}^{k_i} q_i(j) < \infty, \quad G_i = \sum_{n_i=0}^{\infty} \prod_{j=1}^{n_i} \rho^-(j) < \infty,$$

then the Markov process $\{X(t), t \geq 0\}$ is ergodic and its stationary distribution is represented in a product form as

$$p(\vec{k}, \vec{n}) = \prod_{i=1}^M p(k_i, n_i), \quad (19)$$

where

$$p(k_i, n_i) = F_i^{-1} G_i^{-1} \prod_{j=1}^{k_i} q_i(j) \prod_{l=1}^{n_i} \rho_l^-(j), \quad k_i, n_i \geq 0. \quad (20)$$

Proof. We make the substitution of (19), (20) into the system of equations (1), for which the assumptions (14) take place.

After some algebraic transformations we obtain the

equality

$$\begin{aligned} \lambda_0^+ + \lambda_0^- + \sum_{i=1}^M \mu_i^+(k_i) + \sum_{i=1}^M \mu_i^-(n_i) + \sum_{i=1}^M \gamma_i(k_i) = \\ \sum_{i=1}^M \mu_i^+(k_i) p_{ii}^+ + \sum_{i=1}^M \gamma_i(k_i) p_{ii}^+ + \sum_{i=1}^M \frac{\lambda_{0i}^+}{q_i(k_i)} u(k_i) + \\ \sum_{i=1}^M \frac{\mu_i^-(n_i)}{\lambda_i^-} \lambda_{0i}^- + \\ \sum_{i=1}^M q_i(k_i + 1) \mu_i^+(k_i + 1) p_{i0} + \sum_{i=1}^M q_i(k_i + 1) \lambda_i^- p_{i0} + \\ \sum_{i=1}^M q_i(k_i + 1) \gamma_i^+(k_i + 1) p_{i0} + \sum_{i=1}^M \lambda_i^-(1 - u(k_i)) + \\ \sum_{i=1}^M \sum_{j=1, j \neq i}^M \frac{q_i(k_i + 1)}{q_j(k_j)} \mu_i^+(k_i + 1) p_{ij}^+ u(k_j) + \\ \sum_{i=1}^M \sum_{j=1}^M \frac{\mu_j^-(n_j)}{\lambda_j^-} q_i(k_i + 1) \mu_i^+(k_i + 1) p_{ij}^- + \\ \sum_{i=1}^M \sum_{j=1, j \neq i}^M \frac{q_i(k_i + 1)}{q_j(k_j)} \lambda_i^- p_{ij}^+ u(k_j) + \\ \sum_{i=1}^M \lambda_i^- p_{ii}^+ u(k_i) + \sum_{i=1}^M \sum_{j=1}^M \frac{\mu_j^-(n_j)}{\lambda_j^-} q_i(k_i + 1) \lambda_i^- p_{ij}^- + \\ \sum_{i=1}^M \sum_{j=1, j \neq i}^M \frac{q_i(k_i + 1)}{q_j(k_j)} \gamma_i(k_i + 1) p_{ij}^+ u(k_j) + \\ \sum_{i=1}^M \sum_{j=1}^M \frac{\mu_j^-(n_j)}{\lambda_j^-} q_i(k_i + 1) \gamma_i(k_i + 1) p_{ij}^-. \end{aligned} \quad (21)$$

This equality is true for all $(\vec{k}, \vec{n}) \in \mathcal{X}$.

Similarly to the proof of the theorem of the previous case we transform the right part of the equality (21). Let us denote by

$$\begin{aligned} A = \sum_{i=1}^M \frac{\mu_i^-(n_i)}{\lambda_i^-} \lambda_{0i}^- + \\ \sum_{i=1}^M \sum_{j=1}^M \frac{\mu_j^-(n_j)}{\lambda_j^-} q_i(k_i + 1) \mu_i^+(k_i + 1) p_{ij}^- + \\ \sum_{i=1}^M \sum_{j=1}^M \frac{\mu_j^-(n_j)}{\lambda_j^-} q_i(k_i + 1) \lambda_i^- p_{ij}^- + \\ \sum_{i=1}^M \sum_{j=1}^M \frac{\mu_j^-(n_j)}{\lambda_j^-} q_i(k_i + 1) \gamma_i(k_i + 1) p_{ij}^-. \end{aligned}$$

Taking into account the relation

$$q_i(k_i + 1) [\lambda_i^- + \mu_i^+(k_i + 1) + \gamma_i(k_i + 1)] = \lambda_i^+,$$

we obtain

$$A = \sum_{j=1}^M \mu_j^-(n_j). \quad (22)$$

Further let us denote by

$$\begin{aligned} B = & \sum_{i=1}^M \mu_i^+(k_i) p_{ii}^+ + \sum_{i=1}^M \frac{\lambda_{0i}^+}{q_i(k_i)} u(k_i) + \\ & \sum_{i=1}^M \sum_{j=1, j \neq i}^M \frac{q_i(k_i+1)}{q_j(k_j)} \mu_i^+(k_i+1) p_{ij}^+ u(k_j) + \\ & \sum_{i=1}^M \sum_{j=1, j \neq i}^M \frac{q_i(k_i+1)}{q_j(k_j)} \lambda_i^- p_{ij}^+ u(k_j) + \sum_{i=1}^M \lambda_i^- p_{ii}^+ u(k_i) + \\ & \sum_{i=1}^M \gamma_i(k_i) p_{ii}^+ + \sum_{i=1}^M \sum_{j=1, j \neq i}^M \frac{q_i(k_i+1)}{q_j(k_j)} \gamma_i(k_i+1) p_{ij}^+ u(k_j). \end{aligned}$$

After some transformations of the right part with combination with (15) and (16) we obtain

$$B = \sum_{j=1}^M (\lambda_j^- u(k_j) + \mu_j^+(k_j) + \gamma_j(k_j)). \quad (23)$$

Finally, introducing

$$C = \Lambda_0 + \sum_{i=1}^M \lambda_i^- (1 - u(k_i)) \quad (24)$$

we represent the right part of equalities (21) as $A + B + C$.

Using (22) - (24), where Λ_0 , λ_i^+ and λ_i^- are determined by relations (16) and (17), we represent the right part of the equality (21) in the following form:

$$\begin{aligned} A + B + C = & \sum_{j=1}^M \mu_j^-(n_j) + \sum_{j=1}^M (\lambda_j^- u(k_j) + \mu_j^+(k_j) + \gamma_j(k_j)) + \\ & \Lambda_0 + \sum_{i=1}^M \lambda_i^- - \sum_{i=1}^M \lambda_i^- u(k_i) = \\ & \sum_{j=1}^M \mu_j^-(n_j) + \sum_{j=1}^M (\mu_j^+(k_j) + \gamma_j(k_j)) + \Lambda_0 + \sum_{i=1}^M \lambda_i^- = \\ & \lambda_0^+ + \lambda_0^- + \sum_{j=1}^M \mu_j^-(n_j) + \sum_{j=1}^M (\mu_j^+(k_j) + \gamma_j(k_j)). \end{aligned}$$

Thus the substitution of (19), (20) into the system of equations (1), (14), for all $(\vec{k}, \vec{n}) \in \mathcal{X}$, leads to a system of identities. Therefore, the expressions (19), (20) give a solution of the equilibrium system of equations (1), (14) which under the assumptions of the theorem is positive and bounded. As a consequence of this result the process $\{X(t), t \geq 0\}$ is ergodic, thus the theorem is proved. ■

4.3 Conclusion

G-networks provide a versatile class to model complex systems in various applications fields such as computer network and telecommunication systems. In this paper we extended the results of G-networks with product form solution introducing the impatient service. We provided a proof of the product form results for a network in which positive customers are processed by a single server at every node and for a simmetrical networks.

References

- [1] Gelenbe E. and Pujolle G. 1998, "Introduction to Queueing Networks", *John Wiley, New York*.
- [2] Gelenbe E and Fourneau J.M. 2002, "G-Networks with reset". In *Proc. IFIPWG 7.3 /ACM SIGMETRIX Performance'02 Conf.*, Rome, Italy. Vol. 49, Pp 179-192.

- [3] Artalejo J.R., 2000, "G-networks: a versatile approach for work removal in queueing networks". In *Europ. J. Oper. Res.*, V. 126, Pp 233-249.
- [4] Bocharov P.P., 2002, "Queueing networks with signals and random signal activation". In *Automation and Remote Control*, N. 9, Pp 85-96.

Study of Neighborhood search operators for unitation functions

Roberto Santana

Institute of Cybernetics, Mathematics, and Physics (ICIMAF)

Calle 15, e/ C y D, Vedado

CP-10400, La Habana, Cuba

rsantana@cidet.icmf.inf.cu

Abstract- In this paper we study the behavior of neighborhood search algorithms in optimization of unitation functions. The influence of two neighborhood search strategies is analyzed. The expected number of steps required by these algorithms to reach the optimum is derived. The analytical results achieved correspond to previous simulations.

Keywords: Stochastic optimization

1 Introduction

Unitation functions are functions defined in the finite n -dimensional binary space whose values are related to the number of components set to 1 (see section 2 for a formal definition) and have a number of attributes that make them particularly appealing for the investigation of different stochastic optimization algorithms. These functions have received a particular attention in Genetic Algorithms (GAs) [1] and Estimation Distribution Algorithms (EDAs) [5], two Population Based Search Methods that use Selection (PBSMS), commonly used as optimization methods. The performance of GAs and EDAs for unitation functions has been extensively investigated.

Unitation functions are also important because they allow the study of the behavior of optimization algorithms in the presence of multiple local and global optima. They are useful to understand, for example, how EDAs optimize functions by transforming the original landscape in the landscape given by the average fitness of the population, a result that was presented in [4].

One important question that arises is how single searchers behave for these functions. In [6] the computational complexity of Simulated Annealing (SA) [3] for a fixed temperature and neighborhood sizes was investigated in the framework of the optimization of unitation functions. A theoretical comparison of stochastic local search using large neighborhoods with a local search using optimal temperature schedules was done. In [7] a comparison between EDAs and some local searchers for a number of unitation problems was presented.

In this paper we theoretically analyze the influence of the way the neighborhood is defined in the performance of neighborhood based local searchers. We de-

rive a formula for estimating the probability of reaching the optimum in one single step of the search for a subclass of unitation functions. Results are applied to the calculation of the number steps needed by the local searcher to reach the optimum in the case of unitation functions with gaps. The outline of the paper is as follows: Next section introduces the class of unitation functions. Section 3 presents a neighborhood based search algorithm based on the Boltzmann distribution. The analysis of two types of neighborhood search operators is developed in section 4. Section 5 shows how the derived results can be used to estimate the number of steps to reach the optimum for one particular function, the results on the approximation are validated comparing with previous simulations. We present our conclusions in section 6.

2 Unitation functions

Let $X = (X_1, \dots, X_n)$ be a tuple of random variables and $X \in B^n$ where B^n is the finite n -dimensional binary space. We will use x to denote a value of X , and x_i to denote a value of X_i , the i -th component of X . Let f be a function such that $f(x) : B^n \mapsto R^{\geq 0}$, and let $u(x) = \sum_{i=1}^n x_i$.

$f(x)$ is a unitation function if $\forall x, y \in B^n, u(x) = u(y) \Rightarrow f(x) = f(y)$

$u(x)$ is itself a unitation function, usually called *Onemax* because it reaches the maximum at $x = (1, \dots, 1)$. A unitation function can be defined in terms of its unitation value $u(x)$ or, in a simpler way, u . One example is the *Jump* function (1) [6]. The parameter *gap* ($gap \in N, 0 \leq gap < n$) of this function defines the number of steps one has to go downhill in order to reach the unique maximum. For $gap = 0$ we have the very simple *Onemax* function. As the parameter *gap* increases, so does the difficulty of the function. The graphic of function *Jump* is shown in figure 1.

$$Jump(n, gap, u) = \begin{cases} u & u < n - gap \\ 2 * (n - gap - 1) - u & n - gap \leq u \\ n & u = n \end{cases} \quad (1)$$

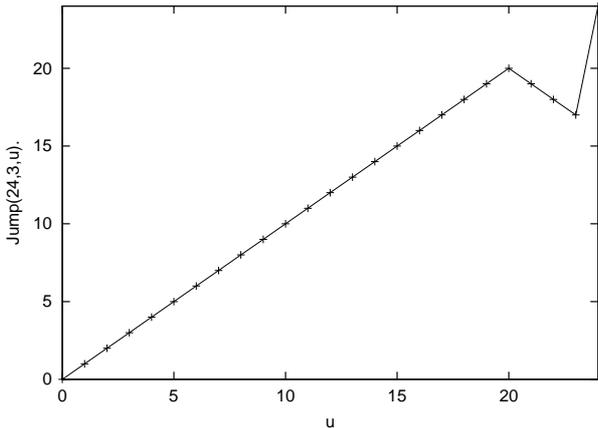


Figure 1: Function Jump, $n = 24$, $gap = 3$.

3 Neighborhood based search algorithms

Neighborhood based search algorithms are stochastic local search methods that start from a single point, and proceed the search for the optimum making transitions to points in a predefined neighborhood of the current one. The ways in which a neighborhood is defined, the probabilities of visiting the different of visiting the different points in the neighborhood, and the criteria to accept transitions to the neighbors determine the dynamics of the search. We will consider neighborhood search algorithms defined on B^n . For this space the size of the neighborhood is defined as the number of solutions that belong to the neighborhood.

Some neighborhood search methods use a dynamical variable to determine transitions in the space of solutions. This is the case Simulated Annealing (SA) which uses the temperature as a dynamical variable that changes with time, and may allow the system to make transitions which would be improbable at a fixed temperature. In SA the neighborhood of a point is usually comprises the set of points that are in an 1-bit distance from the current point.

Recently Mühlenbein and Zimmermann [6] have shown that for stochastic local search algorithms like SA the size of the neighborhood is more important than the temperature schedule. In fact, in the field of local optimization, it has been reported that updating more than one variable at the time can be a good heuristic for escaping local optima when sequential optimization algorithms are used. This is the case for example of GSAT [2], a very effective local optimization strategy used to solve the Satisfiability problem, where the number of variables to be updated in each step is not fixed. We will constrain our analysis to the case of the algorithms that use neighborhood search, where the neighborhood

size s is fixed. A transition from the current state to the next state is done by changing at most s variables of the current state. The search can be done in two steps.

- The s variables that will be changed are selected.
- The values of all or some of the variables are changed.

Let Ng represent a set of variables to be updated, s is the number of variables in Ng , and we will refer to the variables that are not in Ng as X/Ng . For $s = n$ we have $X/Ng = \emptyset$. x_{Ng} is the sub-vector of the vector x formed by variables in Ng . The close neighborhood $v_{Ng}(x)$ of x includes x and the set of points that can be accessed by changing the values in x_{Ng} . From now on we understand a neighborhood as a closed neighborhood.

In the neighborhood search algorithm we use, the set Ng of s variables is uniformly selected from X without replacement. Available information may be use to select the neighbors in a "convenient" biased way. Results for the case where the structure of the function is used in the selection of the blocks can be found in [7]. The new configuration of variables in Ng is found sampling from a neighborhood probability P that is an input of the algorithm. P is defined in the space of the 2^s binary configurations, and it is fixed for any set Ng .

The neighborhood search algorithm is shown in algorithm 1. The algorithm receives as a parameter the block size s , that can be also understood as the maximum number of variables that will change their values together.

Algorithm 1: Neighborhood based searcher

- 1 Set $t \leftarrow 0$. Generate a random initial point x^0 .
 - 2 **do** {
 - 3 Select a set Ng of s different uniformly selected variables of X .
 - 4 Propose a new point x' such that $x'_i = x_i$ if $X_i \in X/Ng$ and x'_i is sampled from the neighborhood probability P for $X_i \in Ng$
 - 5 if $f(x') \geq f(x)$ then $x^{t+1} = x'$ else $x^{t+1} = x$.
 - 6 $t \leftarrow t + 1$.
 - 7 } **until** Termination criteria are fulfilled
-

We have used two different ways of defining transition probabilities. These two ways actually define different types of neighborhoods, and we will refer to them as *Neig1* and *Neig2*. In *Neig1* a uniformly random value x'_{Ng} is selected among the $2^s - 1$ possible values. *Neig2* has been implemented as in [6], the probability

is uniform in the space of the number of the variables that can change their value together. Table 3 shows an example of how the neighborhood $Neig1$ and $Neig2$ are constructed for a point x . Note that while P_{Neig1} is uniform in the 2^s neighbors, P_{Neig2} is not, however P_{Neig2} is uniform in the space of the unitation. Finally if the value of the function at the proposed point x' is not worse than at the current one the transition is made.

4 Analysis of the neighborhood operators

We consider again a maximization problem. Let us suppose that the optimum of $f(x)$ is located at point $x = (1, 1, \dots, 1)$. We want to estimate the probability of making a transition from a point with unitation u to the optimum using a neighborhood search algorithm with neighborhood size s . Let us call this probability P_{trans} . To hit the optimum the following facts have to occur.

- The $(n - u)$ variables with value 0 are selected among the s variables.
- The new proposal changes the values of these $(n - u)$ variables and keep the values of the rest of the variables intact.
- The new proposal is accepted.

Then P_{trans} can be factorized as:

$$P_{trans} = P_{sel} \cdot P_{opt} \cdot P_{acceptance}$$

$$P_{sel} = \frac{\binom{s-(n-u)}{s}}{\binom{n}{s}} \quad (2)$$

Where P_{sel} is the probability of having the $(n - u)$ variables among the s variables selected. P_{opt} is the probability of changing the $(n - u)$ variables to 1, while keeping the remaining $s - (n - u)$ in their current values. P_{opt} depends on the way the neighborhood structure is defined. Finally, given that x' is the new proposal, x' will be accepted if $f(x') \geq f(x)$. Thus, the probability of hitting the optimum is equal to the probability of selecting the optimum as the next proposal. If the current solution has unitation u , and $n - u > s$, the probability of generating the optimum as the new proposal is 0 because in this case it is impossible to make the transition in just one step. It could be the case that the $n - u$ variables that need to be changed are among the set of s variables selected. In this case the transition to the optimum will be done, i.e. $P_{opt} = 1$. The analysis leads to the following theorem.

Theorem 1. *The probability that a random neighborhood based search algorithm with neighborhood size s and that always accept better points reaches the optimum in one step is given by:*

$$P_{trans} = \frac{\binom{s-(n-u)}{s}}{\binom{n}{s}} \cdot P_{opt} \quad (3)$$

Moreover, equation (2) gives the maximum probability of reaching the optimum in one step.

Now let us consider P_{opt} for $Neig1$ and $Neig2$ introduced in the previous section. In $Neig1$ an assignment for the s variables is uniformly random generated in the space of neighbors. This is:

$$P_{trans}^{Neig1} = \frac{\binom{s-(n-u)}{s}}{\binom{n}{s}} \cdot \frac{1}{2^s} \quad (4)$$

This would correspond to a search algorithm with uniform transition rules. But uniform transition rules do not imply a uniform search of the space. For the particular case of uniform search, the $Neig2$ case, the probability of selecting a new assignment for the s variables must satisfy that all the points of the neighborhood are visited with the same probability.

Let us consider the probability of having a neighbor y of x whose Hamiltonian distance from x is h (i.e. $H(x, y) = h$). Obviously, the probability for $h > s$ is zero. For $h \leq s$ this probability is:

$$P_{H(x,y)=h} = \frac{\binom{n}{h}}{\sum_{h'=0}^s \binom{n}{h'}} \quad (5)$$

Then P_{opt}^{Neig2} has to be calculated according to the distance to the optimum h_{opt} . If the current solution has unitation u then $h_{opt} = n - u$ and

$$P_{opt}^{Neig2} = \frac{\binom{n}{n-u}}{\sum_{h'=0}^s \binom{n}{h'}} \cdot \frac{1}{\binom{s}{n-u}}$$

where the first term in the expression corresponds to the probability of changing exactly $n - u$ components while the second expression is the probability of finding the right $n - u$ variables among the s variables selected.

Theorem 2. *The probability that a random neighborhood based search algorithm $Neig2$ with neighborhood size equal s reaches the optimum in one step is:*

$$P_{trans}^{Neig2} = \frac{1}{\sum_{h'=0}^s \binom{n}{h'}} \quad (6)$$

Proof: Substituting (6) in (3) we get:

$$P_{trans}^{Neig2} = \frac{\binom{s-(n-u)}{s}}{\binom{n}{s}} \cdot \frac{\binom{n}{n-u}}{\sum_{h'=0}^s \binom{n}{h'}} \cdot \frac{1}{\binom{s}{n-u}}$$

$Ng = \{1, 3, 5\}$	$v_{\{1,3,5\}}(x)$							
$x = (00000)$	00000	10000	00100	00001	10100	10001	00101	10101
u	0	1			2			3
$P_{Neig1}(x)$	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125
$\sum_x P_{Neig1}(x)$	0.125	0.375			0.375			0.125
$P_{Neig2}(x)$	0.250	0.083	0.083	0.083	0.083	0.083	0.083	0.250
$\sum_x P_{Neig2}(x)$	0.250	0.250			0.250			0.250

Table 1: Definition of two different neighborhood transition probabilities for $v_{1,3,5}(00000)$.

Considering the case when $s = n - u + a$, $a \geq 0$ and substituting in (7) we arrive to:

$$\begin{aligned}
P_{trans}^{Neig2} &= \frac{\binom{u}{a}}{\binom{n}{n-u+a}} \cdot \frac{\binom{n}{n-u}}{\sum_{h'=0}^s \binom{n}{h'}} \cdot \frac{1}{\binom{n-u+a}{n-u}} \\
&= \frac{u!}{(u-a)!a!} \cdot \frac{(n-u+a)!(u-a)!}{n!} \cdot \frac{n!}{(n-u)!u!} \\
&\quad \cdot \frac{(n-u)!a!}{(n-u+a)!} \cdot \frac{1}{\sum_{h'=0}^{n-u+a} \binom{n}{h'}} \\
&= \frac{1}{\sum_{h'=0}^{n-u+a} \binom{n}{h'}} \quad (7)
\end{aligned}$$

Finally, substituting $a = s - n + u$ in (7) we obtain the expression (6).

Corollary 3. P_{trans}^{Neig2} is maximal when $s = n - u$.

5 Structure of the neighborhood, Jump function

We investigate now the role of the neighborhood structure in the case of the *Jump* function. For this function most of the steps spent by a neighborhood search algorithm with neighborhood size s are used to pass from a local optimum with unitation $n - gap - 1$ to the optimum of unitation n .

The expected number of steps to reach the optimum can be calculated as the inverse of the probability for reaching it. Using (4), and (7) we estimate the total number of steps to reach the optimum of the *Jump* function with gap equal gap when *Neig1* and *Neig2* are used. We substitute u by $n - gap - 1$ and s by $gap + 1 + a$.

$$N_{Neig1} = \frac{\binom{n}{gap+1+a}}{\binom{n-gap-1}{a}} \cdot 2^{gap+1+a} \quad (8)$$

$$N_{Neig2} = \sum_{h'=0}^{gap+1+a} \binom{n}{h'} \quad (9)$$

For the *Jump* function it is clear that the minimal number of steps to reach the optimum needed by the

Neig2 (equation (9)) is achieved when $s = gap + 1$, otherwise the number of steps is incremented. This result demonstrates the following theorem presented as an empirical law in [6].

The expected number of steps N_{Neig2} for the *Jump* function and different number of variables are presented in table 2. The predictions are compared with results of the simulations appeared in [6]. As N_{Neig2} is just the number of the steps required to jump from the local optimum to the optimum, it is only a lower bound of the expected passage time τ . Nevertheless it can be appreciated in the table how close is the prediction.

To analyze the case of *Neig1* we consider a simplification of equation (8) for N_{Neig1} .

$$\begin{aligned}
N_{Neig1} &= \frac{n!a!(n - (gap + 1 + a))!2^{gap+1+a}}{(n - (gap + 1 + a))!(gap + 1 + a)!(n - gap - 1)!} \\
&= \frac{n!a!}{(gap + 1 + a)!(n - gap - 1)!} 2^{gap+1+a} \\
&= \frac{a!(n - gap - 1)! \prod_{i=1}^{gap+1} (n - gap - 1 + i)}{(n - gap - 1)!a! \prod_{i=1}^{gap+1} (a + i)} 2^{gap+1+a} \\
&= \left(\prod_{i=1}^{gap+1} \frac{n - gap - 1 + i}{a + i} \right) 2^{gap+1+a} \quad (10)
\end{aligned}$$

In (10) it can be seen that the number of steps depends on two terms. When a is increased the first term decreases but the second one gets exponentially higher. When $a = n - gap - 1$ the first term is 1 and N_{Neig1} is equal to the size of the space. If $a = 0$ then the number of steps becomes:

$$N_{Neig1} = \binom{n}{gap + 1} 2^{gap+1}$$

and this value can be even higher than 2^n . So, for the *Jump* function the *Neig2* makes a more efficient search. Figure 2 shows how N_{Neig1} and N_{Neig2}

n	g	τ	N_{Neig2}	g	τ	N_{Neig2}	g	τ	N_{Neig2}
8	1	47.6	37.0	2	96.7	93.0	3	164.6	163.0
16	1	191.0	137.0	2	718.6	697.0	3	2538.5	2517.0
24	1	430.0	301.0	2	2379.1	2325.0	3	13026.6	12951.0
32	1	764.6	529.0	2	5590.5	5489.0	3	41633.0	41449.0
64	1	3059.4	2081.0	2	44202.2	43745.0	3	680863.5	679121.0

Table 2: Comparison of the simulation results τ for the function *Jump* with the expected number of steps N_{Neig2}

scale when the number of variables is increased for the *Jump* function, $gap = 1, 5$. The size of neighborhood for *Neig1* and *Neig2* are respectively $(2gap + 1)$ and $(gap + 1)$. In the figure the y axes is log-scaled, vertical lines show the first n for which the computation of the number of steps is possible ($n > s$). It can be seen in the figure that the number of steps is always higher for *Neig2*. The difference between the number of steps needed by both algorithms can be intuitively appreciated if we realize that the average number of variables that change their value in every step of the *Neig1* is less than the average for *Neig2*. As a consequence the first algorithm needs more steps for finding a way to cross the gap.

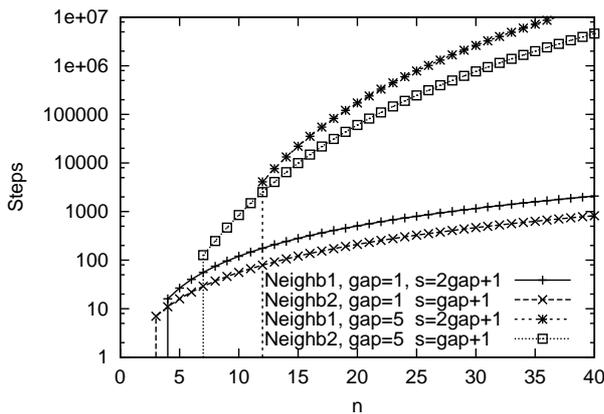


Figure 2: Expected number of steps of the random blocked Gibbs Sampling for the *Jump* function when n is increased.

6 Conclusions

For unitation functions we have calculated which is the maximum probability for a random neighborhood based search algorithm with neighborhood size s to reach the optimum in one step. This formula allows to estimate the average number of steps needed by the algorithm to jump from a local suboptimum of unitation u to the optimum. We have also presented the formu-

lae for calculating this probability for two commonly used neighborhood structures, *Neig1* and *Neig2*. In the case of *Neig2* we have also shown that the optimal choice of the neighborhood size is $s = n - u$. Results have been applied to derive the number of steps needed by function *Jump* to reach the optimum, demonstrating the conjecture presented in [6]. Concerning the differences between *Neig1* and *Neig2* algorithms, an important conclusion of our analysis is that the way transition probabilities of the neighborhood are defined is as critical for the efficiency of the search as the own choice of the neighborhood size is.

7 Acknowledgments

The author thanks to Heinz Muehlenbein for having introduced him to the problem and to Li-Vang Lozada Chang for his comments on the paper, valuable discussion and insight.

Bibliography

- [1] J. H. Holland. *Adaptation in natural and artificial systems*. University of Michigan Press, Ann Arbor, MI, 1975.
- [2] K. Kask and R. Dechter. GSAT and local consistency. In *Proceedings of the 14th IJCAI*, pages 616–622, Montreal, Canada, 1995.
- [3] S. Kirkpatrick, C. D. J. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, May 1983.
- [4] H. Mühlenbein and T. Mahnig. Evolutionary computation and beyond. In Y. Uesaka, P. Kanerva, and H. Asoh, editors, *Foundations of Real-World Intelligence*, pages 123–188. CSLI Publications, Stanford, California, 2001.
- [5] H. Mühlenbein and G. Paaß. From recombination of genes to the estimation of distributions I. Binary parameters. In A. Eiben, T. Bäck, M. Shoenauer, and H. Schwefel, editors, *Parallel Problem*

Solving from Nature - PPSN IV, pages 178–187, Berlin, 1996. Springer Verlag.

- [6] H. Mühlenbein and J. Zimmermann. Size of neighborhood more important than temperature for stochastic local search. In *Proceedings of the 2000 Congress on Evolutionary Computation CEC00*, pages 1017–1024, La Jolla Marriott Hotel La Jolla, California, USA, 2000. IEEE Press.
- [7] R. Santana and H. Mühlenbein. Blocked stochastic sampling versus Estimation of Distribution Algorithms. In *Proceedings of the 2002 Congress on Evolutionary Computation*, volume 2, pages 1390–1395. IEEE press, 2002.

RANDOM SUMMATION AND ITS APPLICATION TO THE PERFORMANCE MODELLING OF COMPUTER SYSTEM

S.L. FRENKEL

*The Institute of Informatics Problems, Russian Academy of Sciences,
Vavilova 44,2, 117333, Moscow, Russia, E-mail: slf-ipiran@mtu-net.ru*

Abstract. This paper presents some views on the problem of application-specific systems (ASCS) performance modelling based on statistical measurement during a program's run time. The probabilistic model considered in this paper defines a performance measure relative to a given domain of application tasks, and it may be used as a base for performance verification. This measure follows a random sum of random summands, corresponding to various runtime component execution times. We consider theoretically whether the current state-of-the-art of random sums' theory can be used as the basis of applied ASCS performance modelling. Relying on the results of the analysis, it is possible to construct a mathematical model which is based on a random sum of the program's operations execution times

Keywords: performance modeling, performance evaluation, random sums, statistical prediction.

1. INTRODUCTION

Performance analysis to predict the execution time of target programs is the basis of the effective design of various application-specific computer systems [ASCS]. Let us consider the prototyping stage (either physical or virtual) of the design process (which may follow the high-level synthesis, simulation). Usually, a designer has a hardware platform variant by this time, as well as a sufficient suite of software modules. In the design phase, the designer tries to define the performance parameters of the application software by using a trace analysis tool. Let us call as "measurement-based", any approach to the execution time prediction, which is based on the results of the tracing and the profiling of a prototype of system design. Ideally, the timing performance verification at this design stage requires a mathematical model to represent the time in terms of some system characteristics, namely, in terms of the measured times of execution of some of the runtime components (e.g., function calls, basic blocks). Various mathematical models that enable the execution time prediction in terms of these measurements results have been suggested by [Saavedra and Smith, 1989; Li, 1996; Gautama, 1998]. Some of these models are deterministic in nature, and they just try to compute the execution time in terms of the various performance "indexes" (e.g., "cycles per instruction" (CPI) or their a high-level analog) [Ferrari, 1983; Saavedra, 1996; Hennessy, 1996]. Since applications' behavior are stochastic (e.g., due to data dependency in programs), the probabilistic performance models are most appropriate for this aim [Gautama, 1998]. However, these probabilistic models are not very practical, since on the one hand, they are based on some specific program models, and, on the other hand, they are fairly time-consuming. Since, in general, the execution time is a random sum of random durations

of the above mentioned runtime components, which may be random values, thereby, the number of these components are also often random, the random sums theory would be very helpful for the performance modelling. This paper considers theoretically whether the current state-of-the-art of random sums theory can be used as a basis of applied ASCS performance modelling. The rest of the paper is organized as follows. Section 2 describes some related work concerning programs' execution times prediction. Section 3 describes the structure of execution time. Section 4 describes random sums concerning the execution time definition. Section 5 describes a random sum based mathematical model in terms of programs' operation execution time. Section 6 describes an example of an application-specific system performance analysis, which is based on the model. This paper, because of limited size, does not contain detailed statistical techniques description, but it only briefly explains some ideas.

2. MEASUREMENT-BASED PERFORMANCE MODELS. RELATED WORKS

Presently the prevalent approach to ASCS performance prediction is based on CPI conception (mentioned above), and represents the time spent by a processor to complete the task [Hennessy, 1996]:

$$\text{CPU_time} = \text{Clock_Cycle_Time} \sum_{i=1, n} (\text{CPI}(i) * \text{IC}(i)) \quad (2.1)$$

where $\text{CPI}(i)$ is the average cycles per instruction for i -th instruction class (e.g. float-pointed, arithmetic, etc.), $\text{IC}(i)$ is the counter of i -th instruction, $i=1, 2, \dots, n$ are the indexes of the instructions set.

More advanced deterministic ways of execution time prediction have been suggested in [Saavedra, 1996], where the machine model consists of a set of abstract

operations of some particular programming language. The predicted execution time of program A on machine M is performed by using detailed measured information about the execution time of these constructs. Since the models are not probabilistic, they may not provide relevant devices to estimate the performance prediction accuracy for some target applications set. Since the goal of our work is to predict the execution time of a program on an application-specific system, we have to be able to estimate both the accuracy of the prediction and the representative features of the data sample used. It is clearly an impossibility to provide it in the framework of such deterministic conceptual model.

Many probabilistic models of execution time estimation (applicable both for sequential and concurrent systems) have been suggested recently [Li, 1996; Gautama, 1998]. These models have been developed in the framework of a "task graph" conception [Adve, 1993]. A task graph is a directed acyclic graph (DAG) in which nodes represent some subtasks and arcs represent the data-dependencies among the subtasks (in terms of which the program is represented). Correspondingly, DAG-based models consider the programs as a set of tasks on a particular input, and consider the program trace in terms of a task set. In [Adve, 93] the focus of the work is on the behaviour of a parallel program for a single input data set, rather than across different input sets, that takes place in case of ASCS designing. In this case, a characteristic of execution time concerning some input domain would be more interesting. The program in this model is decomposed into computationally homogeneous subtasks, and the computational requirements for each subtask are determined. The application is assumed to consist of a set of non-overlapping code segments that are totally ordered in time. The total execution time of the application is the sum of the execution times of all its code segments.

In fact, current publications [Adve, 93; Gautama, 1998] either assume that the execution time is normally distributed, or try to estimate the distributions by computing of moments of the program execution time. However, since by using various probability models we deal with decision rules like inequalities $\text{Prob}(S < t)$, $\text{Prob}(S > t)$, where S is a random value, t is a real one, dealing with so-called "distribution tails", the assumptions about distributions mentioned above may lead to situation of the dramatic loss of accuracy. Secondly, the rate of convergence to the normal distribution may turn out rather slow, so the using the sum may be not suitable if we consider a trace as a sequence of several quite big tasks. Besides, these models deal, in fact, only with the deterministic sums of random items, whereas, the designers often deal with unknown numbers of sequential-performed activities that determine the total time execution, for

example, as a result of a non-deterministic programs behaviour due to branching, data dependencies.

In the above mentioned approaches, the programs-and-data used for the model parameters estimation are assumed as sufficient to reflect the behaviour of all application domains of the designed system. The question is whether indeed set benchmarks are able to predict the machine performance on programs not included in the benchmark suite? For example, formula (2.1) assumes, in fact, that the CPI (i) values have to be estimated from a benchmark suite, while there is no any statistical model for its rational choice.

3. STRUCTURE OF A PROGRAM EXECUTION TIME

Let us consider that we can apply special language features to split the program into intervals and to get performance characteristics for each interval. In general, the execution time of parallel programs on multiprocessor computers is determined by the various factors, e.g., a part of parallel calculations in the total volume of calculations time, or degree of overlapping of interprocessor communications with calculations. Ultimately, execution time is the maximum of the times of the program execution on each processor.

Let us call as "Application Domain" (AD) any set of applied programs with all possible input data to be executed on the designed applied platform. So, the application domain is defined by set of application programs $\{AP\}$ with defined domains of their possible input data, that is an AD is a set of pairs $\{AP_i, ID_i\}$, i is some integer, where each of AP_i is an applied program under input data ID_i .

Let O_1, O_2, \dots, O_r be a set of some specific items ("basic operations" or some disjoint works, functions call, basic blocks), in terms of which we represent a program behaviour as a trace in a sequential manner. Each of O_i is characterized both by the type "i" ($i=1, \dots, r$) and the random time of execution X_i with a distribution function $G_i(X_i)$ (in particular, they may be measured on some system prototype). If the system can execute some operations simultaneously, thanks to either multiprocessing or availability of several executive units in one uni-processor (for example, as it is performed in Alpha 21264 [Alpha, 2000]), then we may consider corresponding individual combinations of operations that can be executed simultaneously. The time of the operations execution should be considered as some random variable. Note, that this randomness may take place both in multi-processors and single-processor systems. The cache hit/miss, pipeline stalls due to hazard, may be considered as corresponding technical factors in a single processor.

When we are able to describe a program runtime as a sequence of some activity pieces (which, in general, may be some aggregations of overlapped operations [Li, 1996]), the execution time of a program API can be expressed as a sums of some summands, corresponding to the execution times, that is expressed as N_p

$$T_p = \sum_{i=1, N_p} (X_j) \quad (3.1)$$

where X_j is the duration of j-th sequential actions in the trace, N_p is the trace length, which should be considered as a random variable, because the different input data suites from a given program input domain may generate different paths in the program execution.

For example, let us consider the distributed execution of any applications in a p-processors's computer system [Ivannikov et al, 2000]. The application execution may be considered as an ordered system of n processes, where each of the processes is an activity dealing with a block of the application running, thereby there is a linear order of executions over the blocks set 1..s, where s is the numbers of the blocks. Under some certain condition about the processes-and-block interactions (e.g. it is impossible to process each block more then on one processor simultaneously, each j-th blocks is distributed to j-th processor), and synchronization conditions (the end of a block in i-th processor coincides with the start of the next block execution in (i+1) processors), the time of n concurrent processes execution is

$$T = \sum_{i=1, n-1} \max_{1 \leq u \leq p} [\sum_{j=1, u} t_{ij} - \sum_{j=1, u-1} t_{i+1, j}] + \sum_{j=1, p} t_{nj}$$

where t_{ij} is an execution time of j-th block of the i-th process.

In practice, such times can be random ones, for example, because of some pipelining effects, the number of processes n may also be dependent of the program input data, .so we deal with the random sums.

4. RANDOM SUMS MODEL OF PERFORMANCE CHARACTERIZATION

Let us consider some possibilities to estimate execution time using the random sums properties.

We can consider the set of operation execution durations (which are the summands) as a triangular array of independent non-negative integer-valued rv's $\{\tau_{ij}\}_{1 \leq j \leq r}$, defined on a probability space (Ω, F, P) , where F is a sigma-algebra, generated by random variables τ_{ij} , r is a number of operation types. "The columns" $j=1, 2, \dots$ define the operations location in the sequence (in a program trace, in fact). In general, we may

consider sampling sums deals with this "triangle array" scheme of random variables

$$S_n = \sum_{m(j)} v_{i,j} \tau_{ij} \quad (4.1)$$

where $m(j)$ is a sequence of integer- random rv's such that $m(j)$ is a stopping time with respect to F_i^j , that is $\{m(j) \leq 1\} \in F_i^j$, $v_{i,j}$ are random variables taking values 0 or 1, $v_{i,j}$, τ_{ij} are independent in each row. This sum corresponds to the scheme of random sampling from a population of real numbers without replacement, where the "population" is a set of arrays $\{\tau_{ij}\}$, each of which corresponds to a program trace. Note, that for some computer architectures, the operations duration can depend on the operation execution order because of pipelining and/or caching influence [Alpha, 2000]. This circumstance is reflected in the sum (4.1), as $v_{i,j}$, which determines implicitly the location of the operation "i" on a place "j" in program trace considered.

So, in fact, (4.1) means that any program trace is a random sample from all possible traces, generated by input data (part of which, of course, can be understood as some "controls"). Any differences between traces are reflected in their operations composition, and rv's $v_{i,j}$ just define if an operation "i" is present on j-th place in the trace, or it is absent.

The key question of the execution time model choice is whether there exists a mathematical technique to compute such a distribution.

4.1 Random Sums Theory: an Applied View

The classical random sums theory results rely essentially on the Kolmogorov-Lindeberg assumption [Zolotarev, 1997] about summands' smallness that is, in the general case, not to be justified for the execution times of program's operations. Some new results [Rahimov, 1995] concern the asymptotic behaviour of (4.1) sums distribution, where the number of array rows ("i") $\rightarrow \infty$, where the above smallness assumptions have been transformed into more weak ones; that is the summand variances are decreasing as $O(i^{-2})$. However, this result is also not very practical, because it is asymptotical in nature, and, in fact, it requires to estimate a distribution of very sophisticated random value.

To obtain more practical results we should include in the estimation model the knowledge about the distribution of numbers of summand N_p in (3.1). For some creditable assumptions about the Poisson distribution of the number of operations, a uniform estimator for the sum distribution deviation about normal law has been obtained (The Berry-Essen inequality for Poisson random sums) [Bening and

Korolev, 2002]. However, this result is also very difficult in practice.

Since our main goal is to characterize the performance relative to some set of programs ("application domain"), and, ultimately, presently the most practical execution time prediction techniques are various regression models (parametric, non-parametric), which deal with some conditional expectations values [Iverson, 1999], studying the expected value of execution time calculation issue is, perhaps, even more important than the probability distribution.

4.2 Expected Value of Random Sums

Speaking about expected values of execution time, we, in fact, should deal with various averaging techniques, that is just what we would have dealt with using any regression techniques of execution time estimation over a benchmarks set [Iverson et al, 1999]. One of the theoretical problems concerning the expected value definition is the Law of Large Numbers (LLN) conditions for the random sums. Strictly speaking, unlike the classical case of sums with non-random summands number, the limit for the "arithmetic mean" is a random value even for the independent identically distributed (i.i.d.) summands, thereby its distribution is completely determined by the asymptotic behavior of the random number of summands [Bening and Korolev, 2002]. So, one of questions is to provide some appropriate averaging.

Formally, we can rely on fact that the above triangular array model corresponds to i.i.d. summands, where the distribution is a mixture of operation type duration distributions $G_r(x)$ (Section 3) that is

$$G(x) = \text{Prob}(\tau_{ij} \leq x) = \sum_{i=1,r} p_i G_i(x) \quad (4.2.1)$$

where p_i are probabilities of appearance of each of the r operations type in the program traces.

Following the Wald identity, we can express the expected value of the execution time as

$$E(T_E) = E(N_p)E(x) \quad (4.2.2)$$

where $E(N_p)$ is the expected value of the trace length N_p over an application domain considered,

$$E(X) = \int_{\Gamma_x} x G(x) dx = \sum_{i=1,r} p_i \int_{\Gamma_x} x G_i(x) dx = \sum_{i=1,r} p_i E(\tau_i),$$

Γ_x is an integration domain (corresponding to x definition).

So, the expected value of the execution time can be expressed as

$$E(T_E) = E(N_p) \sum_{j=1,r} p_j E(\tau_j) \quad (4.2.3)$$

Taking into account above remark about logical difficulties of LLN performing due to randomness of the numbers of summands, the main question deals with the $\{p_j\}$ estimation, which, in classical case (when N_p is a deterministic value) would be estimated as K_i / N_p , where K_i is the numbers of i -th operations appearances in a representative set of traces from an application domain considered. To overcome this problem let us consider an obvious way of the execution time T_E representation in terms of the i.i.d. rv's, corresponding to the execution time structure (Section 3)

$$T_E = \sum_{i=1,r} \sum_{j=1, K_i} \tau_{ij} \quad (4.2.4)$$

where the summands in the inner sum are i.i.d. rv's, thereby, all the numbers of i -th operation appearances K_i are mutually dependent (but they are independent of the rv's τ_{ij}), $K_1 + \dots + K_r = N_p$. The Wald identity can be expressed in this case [Khokhlov, 2003] as

$$E(T_E) = \sum_{i=1,r} E(\tau_i) E(K_i) \quad (4.2.5)$$

where single index "i" is used just to stand for the fact, that all operations of "i" types have the same distribution. Dividing the right side of (4.2.3) by $E(N_p)$, we obtain

$$E(T_E) / E(N_p) = \sum_{i=1,r} (E(K_i) / E(N_p)) E(\tau_i) \quad (4.2.6)$$

Comparing (4.2.3) and (4.2.6) we can obtain the probability estimator $p_i = E(K_i) / E(N_p)$.

The fact of such averaging over the total numbers of the operations in each of traces, as well as over set of each of operations type is correlated with the above mentioned character of LLN for the random sums.

The main question is what is a set of traces (program instances), over which this averaging should be performed. The answer is in interpretation both $\{K_i\}$ set and N_p as a statistics from a representative set of serially run programs which are a rational "benchmarks" set.

5. PERFORMANCE VERIFICATION

Since our main goal is a computer system performance prediction, the question is whether we are able to predict the execution time T_E of a program P of a length N_p with a sufficient accuracy as

$$T_E \approx N_p \sum_{i=1,r} p_i E(\tau_i) \quad (5.1)$$

We may consider this approximation as a regression of T_E on N_p that is as a conditional expected value given N_p . Correspondingly, we may represent the execution time T_p estimator as

$$T_E = N_p \sum_{i=1,r} p_i E(\tau_i) + \varepsilon \quad (5.2)$$

where ε , is stochastic, that is represents the possible errors of time prediction [Iverson99].

But what is a set of traces (program instances), over which this averaging should fulfillment to maximize (5.1) accuracy? In the most general sense this accuracy is determined by both operation times distribution and accuracy of the (p_1, \dots, p_r) vector estimations. It would be attractive to reduce the ε error analysis (5.2) to the rate of convergence for the Law of Large Numbers (e.g., rely on Kolmogorov inequality). However, while for the non-random sums case summands dispersions impact on convergence rate follows the Kolmogorov inequality [Feller,66], the situation for random sum is considerably sophisticated [Bening and Korolev, 2002]. Note, that following the Wald identity, dispersion of the random sum can be calculated by the formula [Bening and Korolev, 2002]

$$DT_E = EN_p D_x + DN_p (Ex)^2 \quad (5.3)$$

where its random summands have the distribution $G(x)$ (4.2.1).

The calculation of this mean and dispersion measure can be carried out over a statistics obtaining from a considered application domain, represented by a program's set. When we are able to estimate the mean and the variance of the execution time, we will be able to estimate the accuracy of the (5.1) representation in a standard statistical manner, calculating the T_E confidence interval size as a function of the variance [Pollard, 1977]. The problem is how to choose the programs/input data ("benchmarks"), to provide these estimations. If the times τ_j are measured very accurately, than the dispersions are defined only by the values of corresponding operations K_i that are used in a program. So, we can choose a program/input data set for the performance verification so as to provide the (p_1, \dots, p_r) vector estimation (where p_1, \dots, p_r are from (4.2.6)) with a suitable accuracy. To achieve this we should know the distribution of vectors, representing the frequency of operations occurring in the program's trace [Frenkel, 1998]. It is easy to see that the distribution is a multinomial. Indeed, as we do not consider any information about the structure of any selected programs, then any events corresponding to the appearance of the basic operation O_i is independent of each other O_j , $i \neq j$, and the appearance of each of them does not change the appearance probability. (In other words, the independence of these events means ignoring the program's semantics).

So, knowing the frequency distribution, we can reduce the problem of benchmarks choice to the well-known problem of providing suitable confidence ellipsoids for the vectors [Pollard,77]. We may define the size and

the structure of a suite of programs, that provides a suitable closeness of the frequencies, calculated over this suite, to the probabilities, used in the above formulas. This (under the above mentioned assumptions about operation's time execution) may ensure the closeness of (5.1) execution time prediction to the true value, allowing performance verification in terms of any target program execution time estimation at the design stages, when the programs may be run only on a host machine, but the operation's duration on the target HW are known with a good accuracy. Operations duration variability impact on prediction ability depends mostly on the times distribution. Briefly, it depends mostly on the "tails" of the distributions. If their cumulative distribution functions $F(x)$ obey the condition $1-F(x) \sim cx^{-\alpha}$, $0 < \alpha < 2$, where \sim means a limit (by $x \rightarrow \infty$) of the fraction of the functions on the left and the right is 1. If $F(x)$ is heavy tailed then the operations' durations values shows a very high variability.

6. AN EXAMPLE

The methodology of performance evaluation was used in our practice for both FX!32 translator (in emulation mode) [Sites, 1992] of x86 applications on the Alpha platform performance optimization and the performance optimization of a RTL ("register-transfer - level) model of a designed microprocessor. The emulator corresponds to a ASCS since it has been designed given all Alpha processor both hardware issues (external cache memory, register file, etc.) and Alpha Windows NT operating system. Some x86 applications sets (for example, MS-Office) could be considered as the system application domain. We can state the problem of FX!32 performance evaluation relative to the application domain. Since we have to define a probabilistic space to operate with the model, we have to understand what the randomness of the operations means in this case. Duration distributions (measured on the Alpha platform) depend on both environmental factors (Dcache miss) and some architectural properties of the emulator (e.g., how much the emulator tables match the x86 instruction structures). Some statistics can be found in <http://www.ipi.ac.ru/~lab24/frenkel>.

7. DISCUSSION AND CONCLUSION

It is well-known that random summation has just as great a role in probabilistic models applications [Zolotorev, 1997; Bening and Korolev, 2002]. However, there is less evidences of its use in such important areas of contemporary computer science as "performance evaluation". In this paper we have discussed a probabilistic and statistical models for the prediction of execution times of sequential and parallel programs in a given operational and hardware environment. These models deal mostly with the some

random sums of random summands. Therefore, the main question of the modelling is whether present-day state-of-the-art of such theory, mostly concerning the LLN and CLT issues, allows us to build a well-grounded model of execution times prediction. This is possible based on expected values of the time predicted. Relying on this analysis, suggests an approach which discharges the necessary to consider summation of enormous numbers of operation times.

The results are in contrast to the traditional approaches (like [Hennessy, 1996]) which express the program execution time in terms of average of cycles per instructions. The traditional approach does not provide any devices both to estimate the accuracy and choose a reasonable benchmarks set, like the considered model, which enables these things. In fact, we encounter the usual difference between some naive statistical approaches, based only on the average numbers and counters of some events, and strong statistical methods, based on a well-grounded probabilistic model, when we can reduce the problem of benchmarks choice to the well-known problem of providing a suitable confidence ellipsoid for the estimated probabilities vector. So, in spite of numerous unresolved questions in the random sums theory, its principal results can be used effectively for performance prediction.

ACKNOWLEDGMENTS

I am very grateful to professors of Moscow State University V.Yu. Korolev and Yu.S. Khokhlov for their comments on some mathematical issues concerning this paper.

REFERENCES

Adve V. 1993, Analyzing the Behavior and Performance of Parallel Programs, Ph.D Thesis Department, Computer Sciences University of Wisconsin-Madison.

Alpha 21264 Microprocessor Hardware Reference Manual, Compaq Computer Corporation, 2000.

Benning V.E., Korolev V.Yu. 2002, "Generalized Poisson Models", VSP, Utrecht.

Feller W. 1966, An Introduction to Theory of Probability and its Applications, Willey & Sons, Inc, New York.

Ferrari D. et al 1983, Measurement and Tuning of Computer Systems, Prentice-Hall.

Frenkel S. L. 1998, Performance Measurement Methodology-and-Tool for Computer System with Migrating Applied Software, *BRICS Notes Series*, NS-98-4, Aalborg, Denmark, June, Pp.83-86.

Gautama H. 1998, A Probabilistic Approach to the Analysis of Program Execution Time Technical Report No. 1-68340-44(1998)06, Faculty of Information Technology and Systems Delft University of Technology.

Hennessy J. and Patterson D. 1996, Computer Architecture: A Quantitative Approach, Second Edition, Morgan Kaufmann Publishers Inc.

Ivannikov V. P. et al 2000, On the Minimal Time Required for Execution of Distributed Concurrent Processes in Synchronous Modes, "Programming and Computer Software vol.26, N5.

Iverson M. et al 1999, Statistical Prediction of Task Execution Times through Analytic Benchmarking for Scheduling in a Heterogeneous Environment IEEE Trans. on comp., Vol 48, N 12, Pp1374-1379.

Khokhlov Yu. S. 2003, Private Communication, Moscow State University.

Li Y. A. 1996, A probabilistic framework for estimation of execution time in heterogeneous computing systems, Ph.D Thesis, the Faculty of Purdue University.

Pollard J.H. 1977, A Handbook of Numerical and Statistical Techniques, Cambridge University.

Rahimov I 1995, Random Sums and Branching Stochastic Processes, Lecture Notes in Statistics, Springer Verlag.

Saavedra R.H. and Smith A.J. 1996, Analysis of benchmark characteristics and benchmark performance prediction, ACM Transactions on Computer Systems, vol. 14, Pp. 344-384.

Sites R L. et al 1992, "Binary Translation", *Digital Technical Journal*, Vol. 4, No. 4.

Zolotarev V.M. 1997, Modern Theory of Summation of Random Variables, VSP, Utrecht.

Sergey L.Frenkel holds M.S. degree both in Radio Communication and Applied Mathematics, and Ph.D degree in Computer Science. He is a senior researcher at the Institute of Informatics Problems Russian Academy of Sciences, and he is an associate professor in Moscow State Tech. University "MIREA". His research interests mainly include probabilistic modelling of digital/computer systems. He has written papers on a variety of topics in mathematical modeling, testability, performance evaluation, as well as one textbook.

REFINED TCP PERFORMANCE EVALUATION WITH SIMPLE MODELING

Sophie FORTIN-PARISI and Bruno SERICOLA

IRISA-INRIA

Campus universitaire de Beaulieu

35042 Rennes cedex, France

Email : {Sophie.Fortin, Bruno.Sericola}@irisa.fr,

Abstract: This paper presents analytical results of a TCP (Transmission Control Protocol) model based on a Markov chain, refining the previous works on performance evaluation of one bulk transfer TCP flow among exogeneous traffic. While most of these works are mainly focused on the mean throughput evaluation, our model allows with low cost, a study of many other performance measures and thus a more detailed study of the TCP behavior.

Keywords: Performance modeling, Markov chain, Transmission Control Protocol, congestion control.

1 INTRODUCTION

The *Transmission Control Protocol* TCP represents a large part of today's Internet transfers. It has been, for that reason, the subject of many studies, centered either on live Internet measurements (downstream), simulations or modeling (upstream). TCP principle is to make sure that all data are actually received by the endpoint. When lost, a *segment* – i.e. a TCP packet – is retransmitted. Based on a sliding window dynamic, new segments are released into the network each time an acknowledgment (*ACK*), a small packet sent by the receiver to confirm the arrival of a segment, arrives. The function of TCP is to modify the window size, that can be correlated to an instantaneous throughput, according to an algorithm defined in the RFC2001 ([Stevens, 2001]): an exponential increase (*slow start*) under a variable threshold, and then successive linear increases (*congestion avoidances*) separated by loss events that halve the window size.

A basic, but efficient, model presented in [Mathis et al, 1997] has shown that the mean throughput ρ of a TCP connection was in the order of $1/\sqrt{p}$, where p denotes the segment loss rate. Among earlier papers proposing a TCP model, many use a continuous-time and fluid approach ([Lakshman and Madhow, 1997], [Kumar, 1998], [Misra et al, 1999], [Altman et al, 1997], [Abouzeid et al, 1999] and [Altman et al, 1999]) and are usually and mainly interested in getting an analytical expression for the mean throughput of a single steady-state TCP connection. The case of multiple TCP connections is the subject of [Ait-Hellal et al, 1997], and [Brown, 2000] for instance, and an original modeling approach is provided in [Baccelli and Hong,

2000] by using the max-plus algebra.

Our paper is based on the reference works of [Padhye et al, 1998], [Padhye et al, 1999], and [Cardwell et al, 2000] which consider a discrete-time model and a discrete evolution of the window size. We present here the results of a discrete-time Markov chain model that we introduced in [Fortin and Sericola, 2001], and which aims to give analytical expressions not only for the mean throughput, but also for various performance measures, of a bulk transfer TCP-Reno flow among exogenous traffic (a flow may represent the transfer of a large data file as well as the global TCP traffic from one ftp server to another for instance).

The organization of this paper is as follows : after a presentation of the model in Section 2, we comment, in Section 3, our results for the mean throughput with a comparison to [Mathis et al, 1997] and [Padhye et al, 1998]. We then give in Section 4 other examples of performance measures which are the proportion of time during which the throughput is maximum, and the time-interval between two consecutive losses.

2 TCP MODELING

The choice of a discrete-time Markov chain modeling the congestion window evolution has been inspired from the pioneering work [Padhye et al, 1998], where the authors introduced the notion of *round* also used in [Padhye et al, 1999], [Cardwell et al, 2000] and [Fortin and Sericola, 2001]. A *round* is the period of time between the departure of the first segment of the current window and the arrival of its ACK. This defi-

dition is coherent when the dispatch duration of all the segments and all the ACKs held in a given window is negligible compared to the *round trip time* RTT. Note that the duration of a round is thus close to the round trip time.

2.1 Presentation Of The Markov Chain

We model the window behavior by a discrete-time Markov chain $X = (X_n)_{n \geq 1}$ with two components $X_n = (W_n^c, W_n^{th})$. W_n^c denotes, when positive, the n -th round congestion window size and the null value for W_n^c is used to represent the time-out period. W_n^{th} denotes the value of the slow start threshold during the n -th round. We denote by b the number of segments validated per ACK. Typically, b is equal to 1 or 2 (in the case of delayed ACKs). TCP-Reno congestion control mechanisms can be described as follows:

- *slow start (ss)* : increase of 1 segment per ACK, that is $W_{n+1}^c = W_n^c + \lceil W_n^c/b \rceil$, as long as $1 \leq W_n^c < W_n^{th}$ and no loss occurs,
- *congestion avoidance (ca)* : increase of $1/W^c$ segment per ACK, that is increase of 1 segment every b rounds, as long as $W_n^c \leq W_n^{th} \leq W_{\max}$ and no loss occurs (when W_n^c reaches the maximum receiver's buffer capacity W_{\max} , it remains constant),
- segment loss detection by three duplicate ACKs (*TD*) : after the first ACK indicating that segment number n is the next expected one, the reception of three successive ACKs indicating that it is still missing notifies the loss of segment number n . $W_{n+1}^c = \max(\lfloor W_n^c/2 \rfloor; 1)$, $W_{n+1}^{th} = \max(\lfloor W_n^c/2 \rfloor; 2)$, and then a new congestion avoidance phase initiates,
- segment loss detection by *time-out (TO)* : when its ACK has not arrived before a timer (T_0) expiry : $W_{n+1}^c = 0$ and $W_{n+1}^{th} = \max(\lfloor W_n^c/2 \rfloor; 2)$, then enter a new *time-out period*,
- *time-out period (to)* : just after a *TO* detection, the segment is retransmitted as long as no ACK for this segment arrives (the timer value doubling from T_0 to $2T_0$, $4T_0$, $8T_0$, ... until $64T_0$), and then a new slow start phase begins with $W_{n+1}^c = 1$.

An illustration of the window evolution is given in Figure 1.

Because the state space E of the Markov chain is such that

$$E \subseteq \{0, \dots, W_{\max}\} \times \{2, \dots, W_{\max}/2\},$$

its size is less than or equal to $(W_{\max} + 1)(W_{\max}/2 - 1)$. The number of states of E is thus less than or equal to 20000 for $W_{\max} < 200$, and less than or equal to 5000 for $W_{\max} < 100$. In both cases, the state space is quite small for the Markov chains computing methods.

2.2 Some Transition Probabilities

All transition probabilities of this Markov chain can be found in [Fortin and Sericola, 2001]. However, for a better understanding of this model, it is interesting to detail the two following phases. We denote by $P_{(i,j)(i',j')}$ the transition probability from state (i, j) to state (i', j') .

2.2.1 Time-out Period

The time-out period corresponds to the case where $W_n^c = 0$. In order to make the mean duration of a time-out period equal to RTT times the mean number of successive visits to the state $(0, j)$, we define the two following transitions from each state $(0, j)$,

- $P_{(0,j)(0,j)} = 1 - \frac{RTT}{E[T_{to}]}$: lost segment not yet ACKed,
- $P_{(0,j)(1,j)} = \frac{RTT}{E[T_{to}]}$: the ACK has just arrived,

where

$$E[T_{to}] = T_0 \frac{f(p)}{1-p} - RTT$$

is the mean duration of a time-out period (see [Fortin and Sericola, 2001]), and

$$f(p) = 1 + p + p^2 + 4p^3 + 8p^4 + 16p^5 + 32p^6.$$

2.2.2 Congestion Avoidance

In congestion avoidance the congestion window is increased by 1 every b rounds, thus for a completely accurate model, we would have to define our model using three components : W_n^c , W_n^{th} , and a counter R_n going from 1 to b (see [Padhye et al, 1999]), with :

- $W_{n+1}^c = W_n^c$ and $R_{n+1} = R_n + 1$ if $R_n < b$ and no loss occurs (case 1),
- $W_{n+1}^c = W_n^c + 1$ and $R_{n+1} = 1$ if $R_n = b$ and no loss occurs (case 2).

However, that would first of all significantly increase the size of the Markov chain and thus any computing time. Secondly, that would not change the measures of interest since the stationary distribution on the state space of the original Markov chain remains the same if we define the transition probabilities such that the mean sojourn time of the Markov chain in a state (i, j) , with $j \leq i < W_{\max}$, remains equal to b , that is, lasts b rounds. We thus have

- $P_{(i,j)(i,j)} = (1-p)^i \left(1 - \frac{1}{b}\right)$ (no loss, case 1),
- $P_{(i,j)(i+1,j)} = (1-p)^i \frac{1}{b}$ (no loss, case 2),

- $P_{(i,j)(0, \max(\lfloor i/2 \rfloor, 2))} = (1 - (1-p)^i) q_i$ (TO -type loss),
- $P_{(i,j)(\max(\lfloor i/2 \rfloor, 1), \max(\lfloor i/2 \rfloor, 2))} = (1 - (1-p)^i) q_i$ (TD -type loss),

where the probability that a loss (in a round of size i) is a TO -type loss is

$$q_i = \frac{(1 - (1-p)^{2b+1}) (1 + (1-p)^{2b+1} - (1-p)^i)}{1 - (1-p)^i}$$

if $i \geq 2b + 2$, and $q_i = 1$ otherwise (see [Padhye et al, 1998]). This formula is obtained by the study of partial rounds, called the *residual rounds*.

This notion is based on the assumption that, when a segment loss occurs, all the following segments in its round get also lost, because the congestion responsible of that loss has not yet disappeared when the last segment of the round arrives. In Figure 2, if the $(k+1)$ -th segment (and thus all the following ones) of the current window is lost, the k first segments will generate ACKs, and thus the congestion window will slide a little and release k new segments that form the residual round.

2.3 Stationary Distribution

Long term TCP transfers are supposed to reach a stationary regime. We will therefore focus on the cyclic stationary behavior of TCP (one *ss* phase, followed by successive *ca* phases until the next TO loss that causes a time-out period, and so on).

Note that, because of the exponential growth during slow start, the Markov chain does not reach all couples (i, j) for $0 \leq i \leq W_{\max}$ and $2 \leq j \leq \lfloor W_{\max}/2 \rfloor$. For instance, for $b = 1$ then the successive congestion window values in slow start are $1, 1 + \lceil 1/b \rceil = 2, 2 + \lceil 2/b \rceil = 4, 8, 16, 32, 64, \dots$, and for $b = 2$ they are $1, 1 + \lceil 1/b \rceil = 2, 2 + \lceil 2/b \rceil = 3, 5, 8, 12, 18, \dots$. Excluding the states (i, j) which are not reached by the Markov chain, we obtain an irreducible and aperiodic finite state Markov chain. Therefore, the stationary probability distribution, denoted by π , exists and satisfies $\pi P = \pi$, where P is the transition probabilities matrix.

2.4 Results

Several measures of interest as, for instance, the speed of convergence to stationary regime, the proportion of time spent in slow start, the mean time-interval between two consecutive losses, the mean number of segments sent and received (successfully transmitted) between two losses or two time-out periods, the proportion of time in which the maximum window size is reached, and of course the mean throughput, can be expressed as functions of p , RTT , T_0 and the stationary probabilities $\pi(i, j)$. Some of them have been explored in [Fortin and Sericola, 2001]. We consider, in the following sections, the evaluation of the throughput, the

mean time-interval between consecutive losses and the maximum window size.

3 THROUGHPUT COMPUTATION

3.1 Send Rate And Goodput

First of all, let us make an important distinction between the throughput in terms of number of segments sent per second which is called the *send rate* (the input rate) and denoted by ρ , and the throughput in terms of number of segments received by the endpoint which is called the *goodput* (the output rate) and denoted by ρ_0 .

The send rate is given by the following formula

$$\rho = \frac{E[d_{to}] + E[d_{cycle}] + N_{loss} E[d_{rr}]}{E[T_{to}] + E[T_{cycle}] + RTT(N_{loss} - 1)p_{rr}},$$

where :

- $E[d_{to}]$, $E[d_{cycle}]$ and $E[d_{rr}]$ denote the average number of segments sent during, respectively, each time-out period (*to*), each *cycle* (one *ss* and successive *ca* until the next TO -loss detection), and each residual round (*rr*),
- $E[T_{to}]$ and $E[T_{cycle}]$ denote the average duration of, respectively, each time-out period and each cycle,
- N_{loss} denotes the average number of losses per cycle,
- p_{rr} denotes the probability that a residual round is not empty, which means that at least one segment of the round that has experienced a loss, has been ACKed (the last residual round of a cycle, i.e. the one due to a TO -type loss, is not taken into account because it is considered as included in the following time-out period).

Similarly, the goodput is given by

$$\rho_0 = \frac{E[d_{cycle}^0] + N_{loss} E[d_{rr}^0]}{E[T_{to}] + E[T_{cycle}] + RTT(N_{loss} - 1)p_{rr}},$$

where $E[d_{cycle}^0]$ and $E[d_{rr}^0]$ represent the mean number of segments successfully transmitted, respectively, during a cycle and during a residual round.

The expressions of all these quantities are detailed in [Fortin and Sericola, 2001].

For illustration, the expressions of the mean number of segments, respectively sent and received, during a cycle (between two successive time-out periods) are given by :

$$E[d_{cycle}] = \frac{\sum_{i=1}^{W_{\max}} i \sum_{j=2}^{\lfloor W_{\max}/2 \rfloor} \pi(i, j)}{p_0 \sum_{j=2}^{\lfloor W_{\max}/2 \rfloor} \pi(0, j)},$$

and

$$E[d_{cycle}^0] = \frac{(1-p) \sum_{i=1}^{W_{\max}} (1-(1-p)^i) \sum_{j=2}^{\lfloor W_{\max}/2 \rfloor} \pi(i,j)}{p_0 \sum_{j=2}^{\lfloor W_{\max}/2 \rfloor} \pi(0,j)},$$

where $p_0 = RTT/E[T_{to}] = P_{(0,j)(1,j)}$ (which means that $p_0 \sum_{j=2}^{\lfloor W_{\max}/2 \rfloor} \pi(0,j)$ is the probability that a cycle starts with the slow start threshold equal to j).

3.2 Comparison To Reference Models

Figure 3 shows that the results of our model are very close to the reference models presented in [Mathis et al, 1997] and [Padhye et al, 1998].

However, our results are slightly lower than theirs. This is explained by the accuracy of our model which, for instance, includes slow start phases and window size limitation. This difference is more obvious for lower RTT values, as shown in Figure 4.

The goodput gives similar results.

3.3 Efficiency

We call efficiency, the ratio $e = \rho_0/\rho$ (output rate over input rate). This ratio represents the percentage of useful data among the transfer load. The remaining load constitutes the retransmission of lost segments. Figure 5 shows the efficiency e for different values of W_{\max} . It confirms that, the higher the throughput is allowed to be (large W_{\max}), the more the transfer suffers losses.

4 OTHER EXAMPLES OF PERFORMANCE MEASURES

As we said in Section 2.4, many performance measures can be done with this model. Here we choose to present, in a first Section, the proportion of time p_{\max} during which the congestion window size is maximum (the instantaneous send rate is W_{\max} segments per RTT), and in a second Section, the time-interval between two consecutive losses.

4.1 Maximum Window Size

Figure 6 shows the evolution of

$$p_{\max} = \sum_{j=2}^{\lfloor W_{\max}/2 \rfloor} \pi(W_{\max}, j),$$

for different values of W_{\max} . Although it is not surprising that p_{\max} is sensitive to W_{\max} , this figure shows that for high values of W_{\max} and low values of p , neglecting a maximum size for the congestion window would not have much impact on the results. This is absolutely wrong for lower values of W_{\max} and higher values of p , e.g. for $W_{\max} = 32$ and $p = 0.001$ we have $p_{\max} \simeq 33\%$.

This means that during one third of the time, the window size is equal to W_{\max} and is not growing anymore. Any model that does not consider a window limitation will thus significantly overestimate the connection throughput.

4.2 Time-interval Between Two Consecutive Losses

Figure 7 shows the mean time-interval between two consecutive losses in a cycle, denoted by $E[\Delta T_{loss}]$, and equal to the mean duration $E[T_{ca}]$ of a congestion avoidance phase. In [Fortin and Sericola, 2001], we proved that

$$E[T_{ca}] = \frac{\sum_{j=2}^{\lfloor W_{\max}/2 \rfloor} (X_j(\alpha_{2j} + \alpha_{2j+1}) + X_{w_{n_j+1}} \beta_j)}{\sum_{j=2}^{\lfloor W_{\max}/2 \rfloor} (\alpha_{2j} + \alpha_{2j+1} + \beta_j)},$$

where

- $\alpha_i = (1 - (1-p)^i) (1 - q_i) \sum_{j=2}^{\lfloor W_{\max}/2 \rfloor} \pi(i, j)$,
- $\beta_j = \pi(w_{n_j}, j)(1-p)^{w_{n_j}}$,
- $X_i = RTT(1-p)^{-\frac{bi(i-1)}{2}} \left(\sum_{w=i}^{W_{\max}-1} \lambda_w + \mu \right)$,
- $\lambda_w = (1-p)^{\frac{bw(w-1)}{2}} \frac{1 - (1-p)^{bw}}{1 - (1-p)^w}$,
- $\mu = \frac{(1-p)^{\frac{bW_{\max}(W_{\max}-1)}{2}}}{1 - (1-p)^{W_{\max}}}$.

When W_{\max} increases, the rounds are likely to reach bigger sizes, and therefore, the risk of a segment loss also increases. That is why the bigger the W_{\max} , the higher the loss frequency, and the lower the $E[\Delta T_{loss}]$.

5 CONCLUSION

This paper is based on a Markov model, and extends the well-known discrete model of [Padhye et al, 1998] which is a reference in modeling the TCP stationary behavior. We have shown that our results for the mean throughput are consistent with previous works led on the subject.

However, we believe that we got more various and accurate results than many other models, without using neither too complex mathematical theories, nor too heavy computation methods. The examples of performance measures that we developed in this paper only represent an sample of what our model can bring. What is more, its strength also lies in its easy adaptability to other additive increase and multiplicative decrease parameters than $1/b$ and $1/2$, and also to other functions of increase and decrease with relatively reasonable modifications. Such a generalization will be the object of further work.

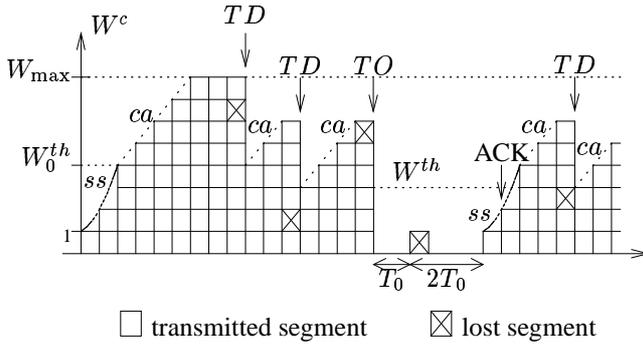


Figure 1: Example of congestion window evolution.

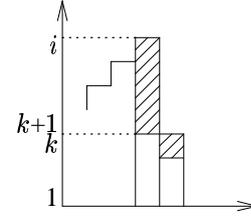


Figure 2: Residual round due to the ACKment of k segments.

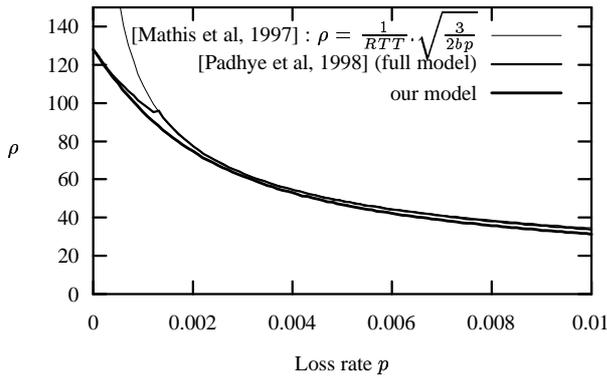


Figure 3: Send rate ρ vs previous models for $RTT = 0.250$ s ($W_{max} = 32$, $b = 2$, $T_0 = 0.500$ s)

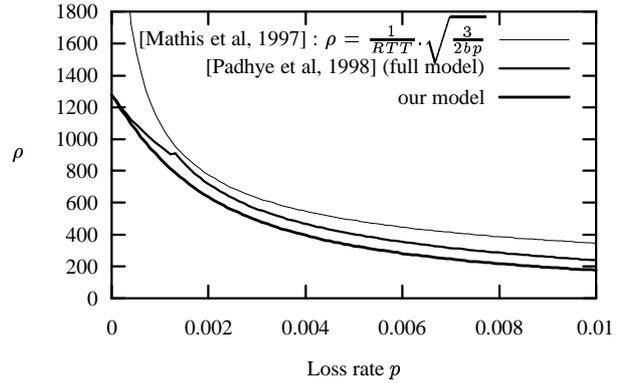


Figure 4: Send rate ρ vs previous models for $RTT = 0.025$ s ($W_{max} = 32$, $b = 2$, $T_0 = 0.500$ s)

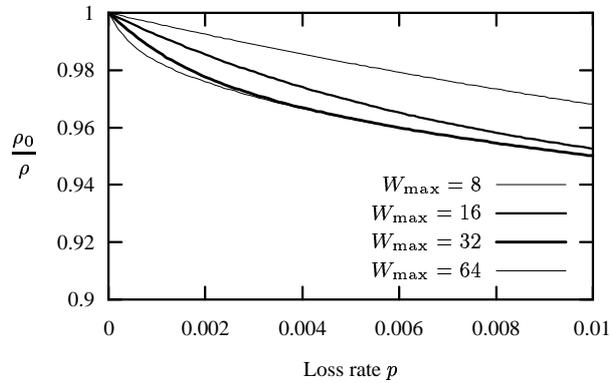


Figure 5: Efficiency $e = \rho_0 / \rho$ for different values of W_{max} ($b = 2$, $RTT = 0.250$ s, $T_0 = 0.500$ s)

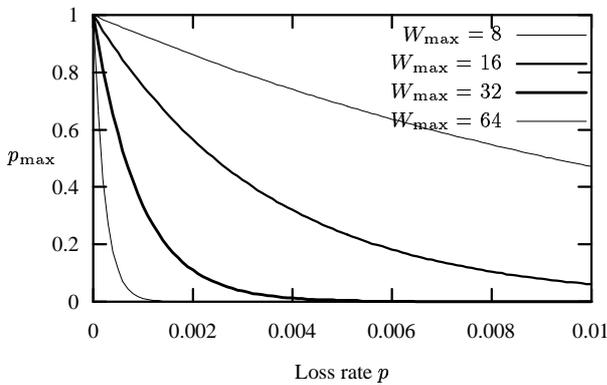


Figure 6: Evolution of p_{max} for different values of W_{max} ($b = 2$, $RTT = 0.250$ s, $T_0 = 0.500$ s)

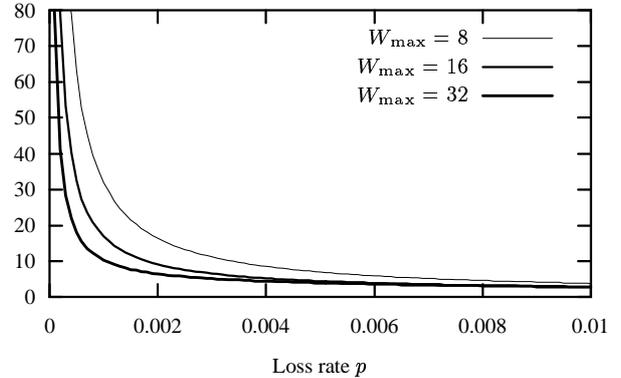


Figure 7: Mean loss interval $E[\Delta T_{loss}]$ in function of loss rate ($b = 2$, $RTT = 0.250$ s, $T_0 = 0.500$ s)

REFERENCES

Fortin S. and Sericola B. 2001, "A Markovian Model for the Stationary Behavior of TCP". *INRIA RR-4240*, <http://www.inria.fr/rrrt/rr-4240.html>.

Padhye J., Firoiu V., Towsley D. and Kurose J. 1998, "Modeling TCP Throughput : a simple model and its empirical validation". *In Proc. SIGCOMM'98* (Vancouver, Canada).

Padhye J., Firoiu V. and Towsley D. 1999, "A stochastic model of TCP Reno congestion avoidance and control". *University of Massachusetts 99-02*.

Cardwell N., Savage S. and Anderson T. 2000, "Modeling TCP latency". *In Proc. INFOCOM'00* (Tel-Aviv, Israel).

Stevens W. 1997, "TCP slow start, congestion avoidance, fast retransmit, and fast recovery algorithms". RFC 2001.

Lakshman T. V. and Madhow U. 1997, "The Performance of TCP/IP for Networks with High Bandwidth-Delay Products and Random Loss". *IEEE/ACM Transactions on Networking* 5(3).

Kumar A. 1998, "Comparative Performance Analysis of Versions of TCP in a Local Networks with a Lossy Link". *IEEE/ACM Transactions on Networking* 6(4).

Baccelli F. and Hong D. 2000, "TCP is Max-Plus Linear". *INRIA RR-3986*.

Brown P. 2000, "Resource sharing of TCP connections with different round trip times". *In Proc. INFOCOM'00* (Tel-Aviv, Israel).

Altman E., Bolot J., Nain P., Elouadghiri D., Erramdani M., Brown P. and Collange D. 1997, "Performance Modeling of TCP/IP in Wide-Area Network". *INRIA RR-3142*.

Altman E., Avrachenkov K. and Barakat C. 1999, "TCP in presence of bursty losses". *INRIA RR-3142*.

Ait-Hellal O., Altman E., Elouadghiri D., Erramdani M. and Mikou N. 1997, "Performance of TCP/IP : the case of two Controlled Sources". *In Proc. ICC'97* (Cannes, France).

Misra V., Gong W.-B. and Towsley D. 1999, "Stochastic Differential Equation Modeling and Analysis of TCP-Window Size Behavior". *In Proc. Performance'99* (Istanbul, Turkey).

Abouzeid A. A., Roy S. and Azizoglu M. 2000, "Stochastic Modeling of TCP over Lossy Links". *In Proc. INFOCOM'00* (Tel-Aviv, Israel).

Mathis M., Semke J., Mahdavi J. and Ott T. 1997, "The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm". *Computer Communications Review* 27(3).

BIOGRAPHY



Sophie FORTIN-PARISI is teaching applied mathematics in the Telecommunications and Networks department of the Institute of Technology (IUT) of Valence (France) since 1998, and is preparing a Ph.D. supervised by Bruno SERICOLA. Her main research activity is in

Internet flow control performance evaluation with stochastic models.



Bruno SERICOLA received the Ph.D. degree in computer science from the University of Rennes I in 1988. He has been with INRIA (Institut National de Recherche en Informatique et Automatique, a public research French laboratory) since 1989. His main research activity

is in computer and communication systems performance evaluation, dependability and performability analysis of fault-tolerant architectures and applied stochastic processes.

A GENERALIZED MARKOVIAN QUEUE TO MODEL AN OPTICAL PACKET SWITCHING MULTIPLEXER

RAM CHAKKA

Department of Computer Science

Norfolk State University, USA

TIEN VAN DO, ZSOLT PÁNDI

Department of Telecommunications

Budapest University of Technology and Economics, Hungary

Abstract: Packet and burst switching have been proposed for optical networks because they can better accommodate bursty traffic generated by IP applications. In optical packet switching networks the payload and the header of the same packet are conveyed in the same channel, while burst switching networks allow the separate transportation of the payload and the header of the same burst. In this paper we consider an optical packet switching node that assigns arriving packets to channels in a link with c available data channels (wavelengths) and a buffer of $L - c$ size. The paper applies the novel MM $\sum_{k=1}^K CPP_k/GE/c/L$ G-queue to model optical packet switching nodes. It is worth emphasizing that our method can be applied to model burst switching nodes as well. Moreover, we show that a model previously presented in the literature is only the special case of our model. Numerical results quantitatively demonstrate that the characteristics (e.g.: burstiness) of the offered traffic have a significant impact on the performance of optical nodes.

Keywords: optical packet switching, optical burst switching, MM $\sum_{k=1}^K CPP_k/GE/c/L$ G-queue, G-networks

1 INTRODUCTION

To efficiently accommodate bursty IP data traffic two technical solutions (packet and burst switching) are being proposed for networks based on optical technology. The final aim is to have networks that switch packets of constant or variable length while the payload data stays in the optical domain. In burst switching networks payload data and its control data (header) are transported in different channels, while packet switching networks convey payload data and its header in the same channel [El-Bawab and Shin, 2002, Yao et al., 2002].

In this paper we develop a new model for optical nodes operating in either optical packet switching or burst switching networks. To evaluate the performance of optical nodes a decomposition approach is used. Namely, the performance of an optical node is determined if we can evaluate the performance of multiplexers before the transmission links. That is, we consider an optical packet (or burst) switching multiplexer that assigns arriving packets (or bursts) to c available data channels (wavelengths) and has a buffer for $L - c$ packets (or bursts). Therefore, we propose the use of the MM $\sum_{k=1}^K CPP_k/GE/c/L$ G-queue to model nodes in both kinds of networks (burst and packet switching), which queue has been proposed recently in [Chakka et al., 2003]. This is a homogeneous multi-server queue with c servers, GE service times and with K independent customer arrival streams, each of which is a CPP, i.e. a Poisson point process with batch arrivals of geometrically distributed batch size.

The use of the MM $\sum_{k=1}^K CPP_k$ process to model packet or burst arrival process is motivated by the following reason. Recent studies have shown that the traffic in today's telecommunications systems often exhibits burstiness – i.e. batches of transmission units (e.g. packets) arrive together – and correlation among interarrival times. As a consequence different mod-

els have been proposed. These models include the compound Poisson process (CPP) in which the inter-arrival times are assumed to have generalized exponential (GE) probability distribution [Kouvatsos, 1994], the Markov modulated Poisson process (MMPP) and self-similar traffic models such as Fractional Brownian Motion (FBM) [Mandelbrot and Ness, 1968, Norros, 1994]. A CPP traffic model often gives a good representation of burstiness of the traffic from one or more sources, e.g. [Bhabuta and Harrison, 1997, Fretwell and Kouvatsos, 1999], but not of the auto-correlations observed in real traffic. Conversely, the MMPP models can capture auto-correlation but not burstiness, e.g. [Fretwell and Kouvatsos, 1997, Meier-Hellstern, 1989]. The self-similar models such as FBM can account for both auto-correlation and burstiness, but they are analytically intractable in a queueing context. Often, the traffic to a node is the superposition of traffic from a number of sources complicating the system analysis further. The MM $\sum_{k=1}^K CPP_k$ captures the burstiness and correlation of the traffic, and its parameter K can be used to model various traffic passing optical nodes from different sources in a flexible manner. Moreover, the Markov modulated $\sum_{k=1}^K CPP_k/GE/c/L$ G-queue is mathematically tractable with efficient analytical solution for the steady state probabilities with the use of mathematically oriented transformations [Chakka et al., 2003]. To obtain the steady state probabilities and thus the performance measures either the spectral expansion method [Chakka, 1995] or Naoumov's method [Naoumov et al., 1997] extended for QBD processes, or the matrix-geometric solution method [Neuts, 1995] can be used.

Related to the performance analysis aspect, Turner has proposed a birth-death process to analyze a multiplexer in optical burst switched networks [Turner, 1999]. However, Turner's model has some limitations like the as-

sumption of exponential burst arrival process, exponential service times and constant burst size. It can be shown and numerically demonstrated that Turner's model is the special case of our model. Moreover, our model overcomes the limitations of Turner's model as regards the arrival process.

The rest of the paper is organized as follows. The proposed model is described in Section 2. Some numerical results are then presented in Section 3. The paper concludes in Section 4.

2 MODEL DESCRIPTION

Since we consider a multiplexer before a transmission link with c available data channels (wavelengths) and a buffer for $L - c$ packets (or bursts), a queueing model for a multiplexer has c servers and L queueing capacity¹ for packets (or bursts). In what follows we outline the important characteristics of the proposed model.

2.1 The Arrival Process

The arrival and service processes are modulated by the same continuous time, irreducible Markov phase process with N states. Let Q be the generator matrix of this process, given by

$$Q = \begin{bmatrix} -q_1 & q_{1,2} & \dots & q_{1,N} \\ q_{2,1} & -q_2 & \dots & q_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ q_{N,1} & q_{N,2} & \dots & -q_N \end{bmatrix},$$

where $q_{i,k}$ ($i \neq k$) is the instantaneous transition rate from phase i to phase k , and

$$q_i = \sum_{j=1}^N q_{i,j}, \quad q_{i,i} = 0 \quad (i = 1, \dots, N)$$

Let $\mathbf{r} = (r_1, r_2, \dots, r_N)$ be the vector of equilibrium probabilities of the modulating phases. Then, \mathbf{r} is uniquely determined by the equations:

$$\mathbf{r}Q = 0 \quad ; \quad \mathbf{r}\mathbf{e}_N = 1.$$

where \mathbf{e}_N stands for the column vector with N elements, each of which is unity.

The arrival process (MM $\sum_{k=1}^K CPP_k$) is the superposition of K independent CPP arrival streams of customers², in a Markov modulated environment. The customers of different arrival streams are not distinguishable. The parameters of the GE inter-arrival time distribution of the k^{th} ($1 \leq k \leq K$) customer arrival stream in phase i are $(\sigma_{i,k}, \theta_{i,k})$. Thus, all the K arrival point-processes are Poisson, with batches arriving at each

point having geometric size distribution. Specifically, the probability that a batch is of size s is $(1 - \theta_{i,k})\theta_{i,k}^{s-1}$, in phase i , for the k^{th} stream of customers.

Let $\sigma_{i,\cdot}, \bar{\sigma}_{i,\cdot}$ be the average arrival rate of customer batches and customers in phase i respectively. Let $\sigma, \bar{\sigma}$ be the overall average arrival rate of batches and customers respectively. Then,

$$\begin{aligned} \sigma_{i,\cdot} &= \sum_{k=1}^K \sigma_{i,k} & ; & & \bar{\sigma}_{i,\cdot} &= \sum_{k=1}^K \frac{\sigma_{i,k}}{(1 - \theta_{i,k})} & (1) \\ \sigma &= \sum_{i=1}^N \sigma_{i,\cdot} r_i & ; & & \bar{\sigma} &= \sum_{i=1}^N \bar{\sigma}_{i,\cdot} r_i \end{aligned}$$

Because of the superposition of many CPP's, the overall arrivals in phase i can be considered as bulk-Poisson ($M^{[x]}$) with arrival rate $\sigma_{i,\cdot}$ and with a batch size distribution $\{\pi_{l/i}\}$ (the probability of batch size being l given that the phase is i) that is more general than mere geometric. The probability that this batch size is l is given by,

$$\pi_{l/i} = \sum_{k=1}^K \frac{\sigma_{i,k}}{\sigma_{i,\cdot}} (1 - \theta_{i,k}) \theta_{i,k}^{l-1} \quad (2)$$

$$\sum_{l=1}^{\infty} \pi_{l/i} = 1.0 \quad (3)$$

The overall batch size distribution is then given by,

$$\pi_{l/\cdot} = \sum_{i=1}^N r_i \pi_{l/i} \quad (4)$$

Define $\pi_{i,l}$ as the probability that a given batch arrival is during phase i and is of size l , then $\pi_{i,l} = r_i \pi_{l/i}$.

2.2 The GE Multi-server

Each data channel will be modelled as a server. Therefore there are c homogeneous servers in parallel, each with GE-distributed service times with parameters (μ_i, ϕ_i) in phase i . The service discipline is FCFS and each server serves at most one customer at any given time. The operation of the GE server is similar to that described for the CPP arrival processes above. L denotes the queueing capacity in all phases, including the packets in service, if any. L can be finite or infinite. When the number of packets is j and the arriving batch size of customers is greater than $L - j$ (assuming a finite L), we assume that only $L - j$ customers are taken in and the rest are rejected.

However, the batch size associated with a service completion is bounded by one more than the number of customers waiting to commence service at the departure instant. For queues of length $c \leq j < L + 1$ (including any packets in service), the maximum batch size at a departure instant is $j - c + 1$, only one server being

¹Including the packets (or bursts) in service.

²A customer denotes either a packet or a burst

able to complete a service period at any one instant under the assumption of exponentially distributed batch-service times. Thus, the probability that a departing batch has size s is $(1 - \phi_i)\phi_i^{s-1}$ for $1 \leq s \leq j - c$ and ϕ_i^{j-c} for $s = j - c + 1$. In particular, when $j = c$, the departing batch has size 1 with probability one, and this is also the case for all $1 \leq j \leq c$ since each packet is already engaged by a server and there are then no packets waiting to commence service.

It is assumed that the first packet in a batch arriving at an instant when the queue length is less than c (so that at least one server is free) *never* skips service, i.e. always has an exponentially distributed service time. However, even without this assumption the methodology described in this paper is still applicable.

2.3 Negative Customers

The parameters of the GE inter-arrival time distribution of negative customers are (ρ_i, δ_i) in phase i . That is, the inter-arrival time probability distribution function is $1 - (1 - \delta_i)e^{-\rho_i t}$ for the negative customers in phase i . Thus, the negative customer arrival *point*-process is Poisson, with batches arriving at each point having geometric size distribution.

A negative customer removes a positive customer in the queue, according to a specified *killing discipline*. When a batch of negative customers of size l ($1 \leq l < j - c$) arrives, l positive customers are removed from the end of the queue leaving the remaining $j - l$ positive customers in the system. If $l \geq j - c \geq 1$, then $j - c$ positive customers are removed, leaving none waiting to commence service (queue length equals to c). If $j \leq c$, the negative arrivals have no effect.

$\bar{\rho}_i$, the average arrival rate of negative customers in phase i and $\bar{\rho}$, the overall average arrival rate of negative customers are given by,

$$\bar{\rho}_i = \frac{\rho_i}{1 - \delta_i} \quad ; \quad \bar{\rho} = \sum_{i=1}^N r_i \bar{\rho}_i \quad (5)$$

Negative customers remove (positive) customers in the queue and have been used to model random neural networks, task termination in speculative parallelism, faulty components in manufacturing systems and server breakdowns [Fourneau et al., 1996, Fourneau and Hernandez, 1993]. The name G-queue has been adopted for queues with negative customers in acknowledgement of Gelenbe who first introduced them. This queueing model can account for burstiness and correlation, but in addition the negative customers, with an appropriate killing discipline, can represent additional behaviours such as breakdowns, killing signals, cell losses and load balancing. We show in Section 3 how negative customers can be used to model packet losses.

2.4 Condition for Stability

When L is finite, the system is ergodic since the representing Markov process is irreducible. Otherwise, i.e. when $L = \infty$, the overall average departure rate increases with the queue length, and its maximum (the overall average departure rate when the queue length tends to ∞) can be determined as,

$$\bar{\mu} = c \sum_{i=1}^N \frac{r_i \mu_i}{1 - \phi_i}. \quad (6)$$

Hence, we conjecture that the necessary and sufficient condition for the existence of steady state probabilities is

$$\bar{\sigma} < \bar{\rho} + \bar{\mu}. \quad (7)$$

2.5 The Steady State Balance Equations

The state of the system at any time t can be specified completely by two integer-valued random variables, $I(t)$ and $J(t)$. $I(t)$ varies from 1 to N , representing the phase of the modulating Markov chain, and $0 \leq J(t) < L + 1$ represents the number of positive customers in the system at time t , including any in service. The system is now modelled by a continuous time discrete state Markov process, \bar{Y} (Y if L is infinite), on a rectangular lattice strip. Let $I(t)$, the phase, vary in the horizontal direction and $J(t)$, the queue length or *level*, in the vertical direction. We denote the steady state probabilities by $\{p_{i,j}\}$, where $p_{i,j} = \lim_{t \rightarrow \infty} \text{Prob}(I(t) = i, J(t) = j)$, and let $\mathbf{v}_j = (p_{1,j}, \dots, p_{N,j})$.

The process \bar{Y} evolves due to the following instantaneous transition rates:

- (a) $q_{i,k}$ – purely lateral transition rate – from state (i, j) to state (k, j) , for all $j \geq 0$ and $1 \leq i, k \leq N$ ($i \neq k$), caused by a phase transition in the Markov chain governing the arrival phase process;
- (b) $B_{i,j,j+s}$ – s -step upward transition rate – from state (i, j) to state $(i, j + s)$, for all phases i , caused by a new batch arrival of size s customers. For a given j , s can be seen as bounded when L is finite and unbounded when L is infinite;
- (c) $C_{i,j,j-s}$ – s -step downward transition rate – from state (i, j) to state $(i, j - s)$, ($j - s \geq c + 1$) for all phases i , caused by a batch service completion of size s , or a batch arrival of negative customers of size s ;
- (d) $C_{i,c+s,c}$ – s -step downward transition rate – from state $(i, c + s)$ to state (i, c) , for all phases i , caused by a batch arrival of negative customers of size $\geq s$ or a batch service completion of size s ($1 \leq s \leq L - c$);

- (e) $C_{i,c-1+s,c-1}$ - s -step downward transition rate, from state $(i, c-1+s)$ to state $(i, c-1)$, for all phases i , caused by a batch departure of size s ($1 \leq s \leq L-c+1$);
- (f) $C_{i,j+1,j}$ - 1-step downward transition rate, from state $(i, j+1)$ to state (i, j) , ($c \geq 2$; $0 \leq j \leq c-2$), for all phases i , caused by a single departure.

Define,

$$\begin{aligned}
B_{j-s,j} &= \text{Diag} [B_{1,j-s,j}, B_{2,j-s,j}, \dots, B_{N,j-s,j}] \\
&\quad (j-s < j \leq L); \\
B_s &= B_{j-s,j} \quad (j < L) \\
&= \text{Diag} \left[\dots, \sum_{k=1}^K \sigma_{i,k} (1 - \theta_{i,k}) \theta_{i,k}^{s-1}, \dots \right]; \\
\Sigma_k &= \text{Diag} [\sigma_{1,k}, \sigma_{2,k}, \dots, \sigma_{N,k}] \\
&\quad (k = 1, 2, \dots, K); \\
\Theta_k &= \text{Diag} [\theta_{1,k}, \theta_{2,k}, \dots, \theta_{N,k}] \\
&\quad (k = 1, 2, \dots, K); \\
\Sigma &= \sum_{k=1}^K \Sigma_k; \\
R &= \text{Diag} [\rho_1, \rho_2, \dots, \rho_N]; \\
\Delta &= \text{Diag} [\delta_1, \delta_2, \dots, \delta_N]; \\
M &= \text{Diag} [\mu_1, \mu_2, \dots, \mu_N]; \\
\Phi &= \text{Diag} [\phi_1, \phi_2, \dots, \phi_N]; \\
C_j &= jM \quad (0 \leq j \leq c); \\
&= cM = C \quad (j \geq c); \\
C_{j+s,j} &= \text{Diag} [C_{1,j+s,j}, C_{2,j+s,j}, \dots, C_{N,j+s,j}]; \\
E &= \text{Diag} (\mathbf{e}'_N).
\end{aligned}$$

Then, we get,

$$\begin{aligned}
B_s &= \sum_{k=1}^K \Theta_k^{s-1} (E - \Theta_k) \Sigma_k; \\
B_1 &= B = \sum_{k=1}^K (E - \Theta_k) \Sigma_k; \\
B_{L-s,L} &= \sum_{k=1}^K \Theta_k^{s-1} \Sigma_k; \\
C_{j+s,j} &= C(E - \Phi) \Phi^{s-1} + R(E - \Delta) \Delta^{s-1} \\
&\quad (c+1 \leq j \leq L-1; s = 1, 2, \dots, L-j); \\
&= C(E - \Phi) \Phi^{s-1} + R \Delta^{s-1} \\
&\quad (j = c; s = 1, 2, \dots, L-c); \\
&= C \Phi^{s-1} \\
&\quad (j = c-1; s = 1, 2, \dots, L-c+1); \\
&= 0 \quad (c \geq 2; 0 \leq j \leq c-2; s \geq 2); \\
&= C_{j+1} \quad (c \geq 2; 0 \leq j \leq c-2; s = 1).
\end{aligned}$$

The steady state balance equations are,

- (1) For the L^{th} row or level:

$$\sum_{s=1}^L \mathbf{v}_{L-s} B_{L-s,L} + \mathbf{v}_L [Q - C - R] = 0; \quad (8)$$

- (2) For the j^{th} row or level:

$$\begin{aligned}
&\sum_{s=1}^j \mathbf{v}_{j-s} B_s + \mathbf{v}_j [Q - \Sigma - C_j - R I_{j>c}] + \\
&\sum_{s=1}^{L-j} \mathbf{v}_{j+s} C_{j+s,j} = 0 \quad (0 \leq j \leq L-1); \quad (9)
\end{aligned}$$

- (3) Normalization

$$\sum_{j=0}^L \mathbf{v}_j \mathbf{e}_N = 1. \quad (10)$$

where, $I_{j>c} = 1$ if $j > c$ else 0, and \mathbf{e}_N is a column vector of size N with all ones.

Each equation ((8, 9, 10)) has *all* the unknown vectors \mathbf{v}_j 's. If L is unbounded, then each of these are infinite number of equations in infinite number of unknowns, \mathbf{v}_j 's, and each equation is infinitely long containing all the infinite number of unknowns. Also, the coefficient matrices of \mathbf{v}_j are j -dependent. It may be noted that there has been neither a solution nor a solution methodology to solve these equations. In [Chakka et al., 2003], a novel methodology is developed to solve these equations *exactly and efficiently*. First these complicated equations are *transformed* to a computable form by using certain mathematically oriented transformations. The resulting transformed equations are of the QBD-M type (QBD with simultaneous-multiple-bounded births and simultaneous-multiple-bounded deaths) and hence can be solved by one of the several available methods, viz. the spectral expansion method, Bini-Meini's method or the matrix-geometric method with folding or block size enlargement [Haverkort and A.Ost, 1997].

2.6 Performance Measures

Some performance measures can be derived as follows:

- Packet loss probability

$$\sum_{j=0}^L \sum_{l=L-j+1}^{\infty} \mathbf{v}_j (\pi_{1,l}, \dots, \pi_{N,l})' \frac{l - (L-j)}{l} \quad (11)$$

- Average departure rate of positive customers

$$\bar{\nu} = \sum_{s=1}^{L-c+1} s \nu_s \quad (12)$$

where

$$\nu_n = \sum_{i=1}^N \sum_{j=c+n}^L p_{i,j} (1 - \phi_i) \phi_i^{n-1} c \mu_i + \sum_{i=1}^N p_{i,c+n-1} \phi_i^{n-1} c \mu_i \quad (n = 2, \dots, L - c + 1) \quad (13)$$

$$\text{and } \nu_1 = \sum_{i=1}^N \sum_{j=1}^c p_{i,j} j \mu_i + \sum_{i=1}^N \sum_{j=c+1}^L p_{i,j} (1 - \phi_i) c \mu_i \quad (14)$$

3 NUMERICAL RESULTS

Three numerical results are presented. First, we show that Turner's model is the special case of our model. Next, we present the impact of bursty traffic on the performance of the system. Note that in the first two cases, no negative customers are allowed in the system. Then, we show how the throughput of connections can be determined through the presence of negative customers.

3.1 Turner's Model is the Special Case of our Model

In this section we demonstrate that Turner's model for burst switching is the special case of our model by letting $K = 1$, $N = 1$, $[q_{i,j}] = [0]$, $\theta_{1,1} = 0$, $\phi_1 = 0$, $\mu_1 = 1$. It is easy to prove that the traffic load is determined by $\sigma_{1,1}$.

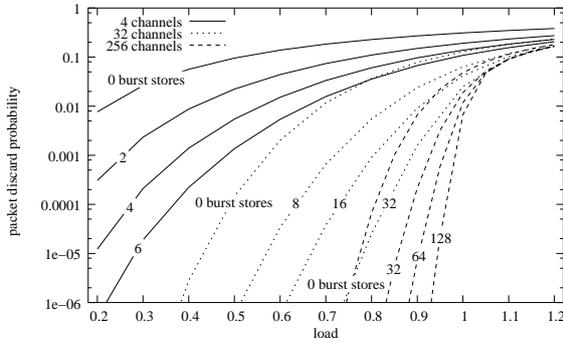


Figure 1: Packet loss probability vs load and c

Figure 1 is exactly the same as Figure 2 in [Turner, 1999], except that the data was produced by our model with the parameter settings mentioned earlier. In order to demonstrate the equivalence, the results were calculated and compared to 20 significant digits using both models for a subset of the parameter set displayed on Figure 1. The calculations were executed on a Sun Ultra 60 Workstation, which had a machine epsilon³ $\epsilon = 1.9 * 10^{-34}$. Table 1 summarizes the outcome. It is clear that the differences between the results produced by the two models are $O(\epsilon)$.

³The machine epsilon is the smallest floating point number that bounds the roundoff in individual floating point operations.

Table 1: Numerical comparison of Turner's model and the MM $\sum_{k=1}^K CPP_k/GE/c/L$ model for $c = 32$

load	number of identical digits					exponent of numerical value				
	b	0	8	16	24	32	0	8	16	24
0.2	20	16	9	4	0	-13	-18	-24	-30	-35
0.3	20	20	16	13	9	-9	-13	-17	-21	-25
0.4	20	20	20	18	15	-6	-9	-12	-16	-19
0.5	20	20	20	20	20	-4	-7	-9	-12	-14
\vdots						\vdots				
1.2	20	20	20	20	20	-1	-1	-1	-1	-1

3.2 Impact of Bursty Traffic

In this section we show the impact of the burstiness of the offered traffic on the performance of the multiplexer.

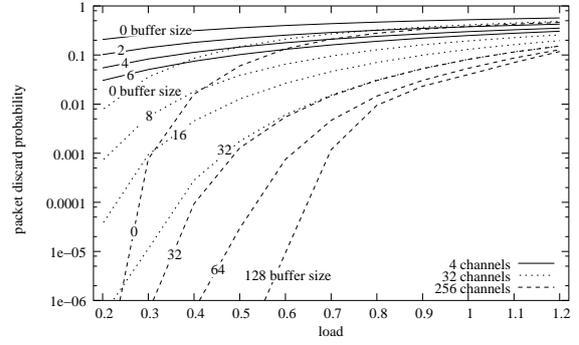


Figure 2: Packet loss probability vs load and c

Figure 2 plots the packet discard probability for this numerical example where batch arrivals are allowed. It is clearly observed that batch arrivals have a significant impact on the performance of the system and batch arrivals can be better handled by increasing the buffer space (at the expense of some queueing delay) than by increasing the number of channels. The performance of 256 channels with no buffer is worse than that of 32 channels with a buffer for 8 packets in our example for relative load values above 0.4.

3.3 Impact of the Connection Loss on the Connection Throughput

In this section we present an approximation to calculate the performance parameter (throughput) of a connection based in the presented queueing model. We also illustrate, then, the impact of a packet loss on the performance of a connection. The considered problem here is the approximation of the throughput of two communicating peers in optical networks. A preliminary approximation can be proposed as follows. The throughput of two communicating peers can be approximated with the queueing model of a single node incorporating the packet loss phenomena along the path. It is showed based on measurements in [Yajnik et al., 1999] that packet loss can be modelled as a 2-state Markov

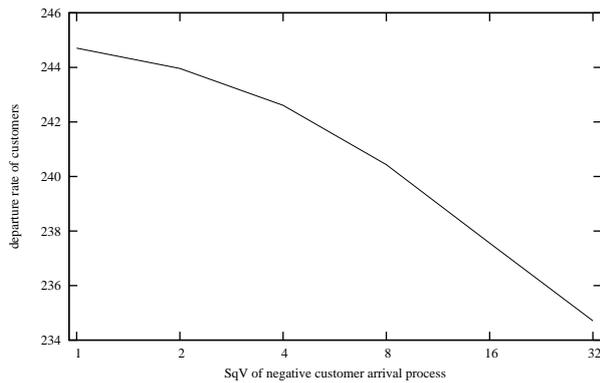


Figure 3: Effect of the negative customer arrival process

chain model. Therefore, the MM $\sum_{k=1}^K CPP_k / GE/c/L$ G-queue can be applied in this case, where negative customers model the loss along the path, and the departure rate of positive customers is the performance measure related to the throughput of a connection.

Figure 3 illustrates the dependency of the customer departure rate on the parameter controlling the packet loss process (modelled by negative customers). It can be observed that the correlation of the packet losses has a significant impact on the performance of the system.

4 CONCLUSIONS

We have applied a new queueing model for the performance analysis of optical packet switching nodes, which model overcomes some of the limitations of the previous work. Moreover, it is shown that Turner's model is the special case of our model. Numerical results quantitatively demonstrate that the characteristics (e.g.: burstiness) of the offered traffic have a significant impact on the performance of optical nodes. In addition the proposed model is able to handle large or unbounded batch sizes, both in arrivals and services, with great computational efficiency and hence may have definite advantages over BMAP based models.

REFERENCES

[Bhabuta and Harrison, 1997] Bhabuta, M. and Harrison, P. (1997). Analysis of ATM Traffic on the London MAN. In *Proc. 4th Int. Conf. on Performance Modelling and Evaluation of ATM Networks*, Ilkely. Chapman and Hall.

[Chakka, 1995] Chakka, R. (1995). *Performance and Reliability Modelling of Computing Systems Using Spectral Expansion*. PhD thesis, University of Newcastle upon Tyne (Newcastle upon Tyne).

[Chakka et al., 2003] Chakka, R., Do, T. V., and Pandi, Z. (2003). The MM $\sum_{k=1}^K CPP_k / GE/c/L$ G-Queue: Steady Solution, Applications and Extensions. Technical report, Norfolk University and Budapest University of Technology and Economics. <http://cctic03.hit.bme.hu/tr/sigmaj.pdf>.

[El-Bawab and Shin, 2002] El-Bawab, T. S. and Shin, J.-D. (2002). Optical Packet Switching in Core Networks: Between Vision and Reality. *IEEE Communications Magazine*, pages 60–65.

[Fourneau et al., 1996] Fourneau, J., Gelenbe, E., and Suros, R. (1996). G-networks with Multiple Classes of Positive and Negative Customers. *Theoretical Computer Science*, 155:141–156.

[Fourneau and Hernandez, 1993] Fourneau, J. and Hernandez, M. (1993). Modelling defective parts in a flow system using g-networks. In *Second International Workshop on Performability Modelling of Computer and Communication Systems*, Le Mont Saint-Michel.

[Fretwell and Kouvasos, 1997] Fretwell, R. and Kouvasos, D. (1997). Correlated Traffic Modelling and Batch Renewal Markov Modulated Processes. In *Proc. 4th IFIP Workshop on Performance Modelling and Evaluation of ATM Networks*, pages 20–44, Ilkely. Chapman and Hall.

[Fretwell and Kouvasos, 1999] Fretwell, R. and Kouvasos, D. (1999). ATM Traffic Burst Lengths Are Geometrically Bounded. In *Proceedings of the 7th IFIP Workshop on Performance Modelling and Evaluation of ATM & IP Networks*, Antwerp, Belgium. Chapman and Hall.

[Haverkort and A.Ost, 1997] Haverkort, B. and A.Ost (1997). Steady State Analyses of Infinite Stochastic Petri Nets: A Comparison between the Spectral Expansion and the Matrix Geometric Methods. In *Proceedings of the 7th International Workshop on Petri Nets and Performance Models*, pages 335–346, Saint Malo, France.

[Kouvasos, 1994] Kouvasos, D. (1994). Entropy Maximisation and Queueing Network Models. *Annals of Operations Research*, 48:63–126.

[Mandelbrot and Ness, 1968] Mandelbrot, B. and Ness, J. (1968). Fractional brownian motions, fractional noises and applications. *SIAM Review*, 10:422–437.

[Meier-Hellstern, 1989] Meier-Hellstern, K. (1989). The Analysis of a Queue Arising in Overflow Models. *IEEE Transactions on Communications*, 37:367–372.

[Naoumov et al., 1997] Naoumov, V., Krieger, U., and Wagner, D. (1997). Analysis of a Multi-server Delay-loss System with a General Markovian Arrival Process. In Chakravathy, S. and Alfa, A., editors, *Matrix-analytical methods in Stochastic models*, pages 43–66. Marcel Dekker.

[Neuts, 1995] Neuts, M. (1995). *Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach*. Dover Publications.

[Norros, 1994] Norros, I. (1994). A Storage Model with Self-similar Input. *Queueing Systems and their Applications*, 16:387–396.

[Turner, 1999] Turner, J. S. (1999). Terabit Burst Switching. *Journal of High Speed Networks*, 8:3–16.

[Yajnik et al., 1999] Yajnik, M., Moon, S., Kurose, J., and Towsley, D. (1999). Measurement and Modeling of the Temporal Dependence in Packet Loss. In *INFOCOM'99*, New York.

[Yao et al., 2002] Yao, S., Xue, F., Mukherjee, B., Yoo, S. J. B., and Dixit, S. (2002). Electrical Ingress Buffering and Traffic Aggregation for Optical Packet Switching and Their Effect on TCP-Level Performance in Optical Mesh Networks. *IEEE Communications Magazine*, pages 66–72.

Ram Chakka obtained a Ph.D degree from University of Newcastle upon Tyne in 1995. He is an associate professor at Department of Computer Science, Norfolk State University.



Tien Van Do obtained a Ph.D degree from the Budapest University of Technology and Economics in 1996. He is an associate professor at Department of Telecommunications, Budapest University of Technology and Economics.



Zsolt Pándi obtained a M.Sc degree from the Budapest University of Technology and Economics in 2001. He is a Ph.D student at Department of Telecommunications, Budapest University of Technology and Economics.



COMPARATIVE PERFORMANCE EVALUATION OF E-COMMERCE TECHNOLOGIES: A TPC-W-BASED BENCHMARKING TOOL

YUSSUF N. ABU SHAABAN* and JANE HILLSTON

ICSA, School of Informatics, University of Edinburgh.

Abstract E-commerce systems are an important new application area in which maintaining good performance under scaling workloads is crucial to business success. The TPC-W benchmark is a benchmark designed to exercise a web server and associated transaction processing system in representative e-commerce scenarios. Whilst the benchmark specifies the architecture of the system, and the form of the interactions between users, web server and underlying database, it does not stipulate the supporting technology used to communicate between the web server and the database. In this paper we describe a tool which has been developed to allow comparative benchmarking of different technologies with minimal re-implementation. The tool implements the TPC-W specification in a modular fashion which allows different application servers to be inserted and experimented upon. The use of the tool in the comparative benchmarking of Java Servlets and PHP is demonstrated

Keywords Software Tools, Software Performance, Performance Modelling, Measurements Techniques, Workload Modelling and Characterization.

1 INTRODUCTION

In recent years e-commerce systems have become one of the most noticeable manifestations of the Internet phenomenon. Although there are no widely accepted definitions, such systems are generally deemed to provide facilities for commercial transactions to take place between remote participants. In particular, three categories of e-commerce systems are identified [3]: *Electronic Markets*, *Electronic Data Interchange (EDI)* and *Internet Commerce*. For the remainder of this paper we will treat “*e-commerce*” as synonymous with *Internet commerce*.

In many e-commerce systems, timely behaviour is crucial in order to maintain the site owner’s competitive edge: poor performance can quite literally translate into lost revenue [1]. However, little systematic work has been done on analysing the performance of such systems and their supporting technologies. In this paper we describe a tool which aims to provide a framework in which different e-commerce programming technologies can be easily benchmarked against the standard TPC-W benchmark [11].

The TPC-W benchmark has been developed by the Transaction Processing Performance Council (TPC) as a response to the rise of e-commerce systems. It specifies the behaviour of an on-line bookstore, including many of the elements commonly found in e-commerce applications: a web-site supported by a web serving component which can present both static and dynamic web pages; a relational database which is accessed from the web server to provide transaction processing and decision support. Moreover the benchmark also spec-

ifies emulated remote browsers for different classes of customers, providing the workload on the system.

Our tool implements the TPC-W benchmark, in Java, as a framework for the investigation of application server technologies. The application server sits between the web server and the database and provides the business logic as well as the interface between the manipulation and the presentation of the data. Whilst the benchmark places constraints on the application server it does not stipulate how it should be implemented. Available technologies for this module of the system include CGI, ASP, Java Servlets, JSP and PHP. Furthermore, the tool incorporates support for experiment design, allowing the user to replicate runs of the benchmark and compare the results obtained with different workload mixes. A detailed description of the tool design can be found in [8].

The remainder of the paper is structured as follows. In Section 2 we review the role of the application server within an e-commerce system and the different technologies currently available. In Section 3 we describe the TPC-W benchmark and in the following section we show how this is captured within the design of our tool. Section 5 describes the implementation of the tool. Results of using the tool in evaluating the scalability of Java Servlets and PHP are given in Section 6. Finally in Section 7 we give some conclusions.

2 APPLICATION SERVERS

The application server is a vital component of an e-commerce system: it is the software component that

*Corresponding author, DIRC, supported by EPSRC studentship 00317428

contains the business logic of the system. It is responsible for receiving HTTP requests via the web server from clients and executing the business functions associated with each of them. This can involve interacting with database servers and/or transaction servers. The application server also has the responsibility of dynamically building web pages, formatting database query results in HTML, to be sent back to clients.

Today, many programming environments exist for implementing application servers, fulfilling the need for dynamic web-site generation and allowing the connection between web front-ends and databases. These include the Common Gateway Interface (CGI) [4], Microsoft's Active Server Pages (ASP) [5], Personal Home Pages (PHP) [10], Java Servlets [7] and Java Server Pages (JSP) [6]. Currently in the tool we offer implementations of Java Servlets and PHP, and we give a brief description of these below:

Java Servlets Java Servlets [7] are server side Java code that runs on a server to answer client HTTP requests. Servlets make use of the Java standard extension classes in the packages `java.servlet` and `javax.servlet.http`. Since servlets are written in the highly portable Java language and follow a standard framework, they provide a means to create sophisticated server extensions in a server and OS independent way. When a servlet program is called for the first time, it is loaded into memory. After the request is processed, the servlet remains in memory and will not be unloaded from memory until the server is shutted down. Servlets offer excellent connectivity with many databases through the use of Java's JDBC package. The output of HTML in a servlet environment is an issue, as the servlet is required to output all HTML internally. This requires complicated output statements to handle the output of the entire HTML content, as well as the code for the rest of the application.

Personal Home Pages Personal Home Pages (PHP) [10] is a server-side, cross-platform, HTML embedded scripting language. It was developed late in 1994 by Rasmus Lerdorf to keep track of visitors to his on-line resume. Since then, it has undergone several changes with two versions released, PHP3 and PHP4. Currently, PHP is shipped with a number of commercial products such as Stronghold web server and Red-Hat Linux. PHP provides a programming approach similar to VBScript of ASP [5], but its broad support for databases gives it an edge over VBScript. Its code can be embedded directly into a HTML page and executes on the server. PHP modules are lightweight and speedy and have no process creation overhead.

3 TPC-W BENCHMARK

The TPC-W benchmark [11] has been developed by the Transaction Processing Performance Council (TPC), a consortium of system and database vendors. Historically, TPC has specified standard benchmarks for evaluating the performance of both transaction processing and decision support database systems. One of its

latest benchmarks is TPC-W, an e-commerce-specific benchmark. It specifies the behaviour of an on-line bookstore, including the three main components of an e-commerce application: remote browsing, web server and database server. TPC-W's remote browsing specifications are described next. This is followed by an overview of the web and database server components. Finally, TPC-W's performance metrics and measurement intervals specifications are discussed.

Remote Browser Emulator A main component of the TPC-W benchmark is the Remote Browser Emulator (RBE), which is a specification for a set of Emulated Browsers (EBs). EBs simulate the activities of concurrent web browsing e-commerce users, each autonomously traversing the bookstore web pages by making requests to a web server. Each EB can represent one of three classes of users: a customer, a new user or a site administrator. As described in the next paragraph, TPC-W defines 14 web interactions which can be requested by an EB. During its lifetime, an EB requests a sequence of these web interactions moving from one interaction to the next, in the same way that a web browsing user navigates a site clicking one hypertext link after another. TPC-W specifies the next possible navigation options that can be requested by an EB on completion of each of the web interactions defined in the benchmark. Threshold integer values between 1 and 9999 are specified for each navigation option. To select its next request, the EB generates a random number, from a uniform distribution between 1 and 9999. It then selects the navigation option for which the threshold is equal to, or most immediately greater than the random number. The EB spends a random period of time (*Think Time*) sleeping between subsequent web interactions. This emulates the user's think time and is generated from an exponential distribution specified in TPC-W. User-specific information must be maintained in an EB, possibly including session tracking details and customer identification.

Web and Database Servers The TPC-W benchmark defines 14 web interactions to be supported by a web server component which are: Home, Shopping Cart, Customer Registration, Buy Request, Buy Confirm, Order Inquiry, Order Display, Search Request, Search Result, New Product, Best Seller, Product Detail, Admin Request and Admin Confirm. These interactions vary in the amount of server-side processing they need. Some require dynamically generated HTML pages and one or more connections to a database. Others are lightweight, requiring only web serving of static HTML pages and images. For each web interaction, TPC-W specifies its input requirements, processing definition, response page definition and EB navigation options which are the set of web interactions that can be selected by the EB on completion of the interaction. Session tracking is vital to any e-commerce application. Some method is needed to retain information such as shopping carts from one HTTP request to another. TPC-W suggests two techniques for session tracking which are URL-rewriting and cookies [2].

The TPC-W benchmark defines the exact schema used for an online bookstore database consisting of eight tables: *item*, *customer*, *address*, *order*, *order line*, *credit card transaction*, *author* and *country*. A *scale factor* is also defined, that is the size of the *item* table. The size of the database depends on the number of EBs that will be used as a workload and the *scale factor*. TPC-W specifies database table sizes as follows:

- Item: Scale Factor.
- Customer: 2880 * Number of EBs rows.
- Country: 92 rows.
- Address: 2 * Number of customers rows.
- Orders: 0.9 * Number of customers rows.
- Order_line: 3 * Number of orders rows.
- Author: 0.25 * Number of items rows.
- cc_xacts: 1 * Number of orders rows.

Performance Metrics and Measurement Intervals TPC-W defines one primary performance metric which is throughput, measured as the number of completed web interactions per second (WIPS). Three distinct measurement intervals are specified by TPC-W: shopping, browsing and ordering. They are distinguished by the ratio of browsing-related web pages visited to ordering-related web pages visited during the measurement interval. The shopping interval is intended to reflect a shopping scenario, in which 80% of the pages visited are related to browsing and 20% are related to ordering. In a browsing interval, ordering pages visited go down to 5% whereas in an ordering interval the ratio of browsing and ordering is even.

4 DESIGN OF THE TOOL

Our tool allows comparative benchmarking of e-commerce programming technologies. In designing the tool, a number of high-level design objectives were emphasised. These are discussed below. In the following we present an overview of the design. A detailed design description can be found in [8]

- It was felt important that the tool should provide experiment design features allowing the user to specify a number of experiments. For each experiment, factor level combinations such as size of workload, size of store and measurement interval type can be specified. The number of replications for each experiment can also be specified.
- Another key objective was to provide performance metrics to assist e-commerce technology performance evaluators. Metrics such as response time frequency distributions of web interactions, overall response time and overall throughput are analysed and presented graphically.
- Ensuring that the analysis of experimental results have minimal overhead was an objective from early stages in the design, thus preserving the realistic nature of the simulation. This resulted in a local data collection strategy in the design. Experimental data results are maintained locally by different parts of the system. Results are gathered at the end of each experiment run from different parts of the system for analysis.

- Realistic e-commerce modelling was an important design criterion. This was ensured by adopting TPC-W as an e-commerce model. Not only different e-commerce site components are represented faithfully; the design also includes realistic workload generation capabilities based on TPC-W's e-commerce access patterns.
- Developing a flexible, extendable tool was a major concern. This led to a modular design where incorporating a new programming technology for benchmarking just involves adding a new module to the tool. In addition, modular design allows for the easy extension of the tool to benchmark other e-commerce components such as security protocols, session tracking techniques, etc.

Figure 1 illustrates the main modules of the tool. The RBE module is responsible for generating and maintaining the workload on the Web and Database Servers. Its design is based on the TPC-W benchmark as described in Section 3. The Web Server contains the application server which represents the application code for implementing the 14 web interactions specified in the TPC-W benchmark (see Section 3). The Database Server represents the persistent storage component of the TPC-W e-commerce model. The final main module in the design is the Control Unit which provides experimental design features, data gathering and analysis. It is also responsible for controlling the setup and maintenance of experiments on various parts of the tool. The Control Unit module is described next, this is followed by a description of RBE and finally the Web and Database Servers are described.

Control Unit The Control Unit is the central component of the benchmark. In addition to providing a GUI to the user, it is responsible for setting up and maintaining the running of experiments. It is also responsible for gathering experiment data results from different parts of the system for analysis and presentation. Figure 2 illustrates the main sub-modules of the Control Unit. The ControllerGUI provides a GUI allowing the user to input experimental design details. The Experimental Design component is responsible for holding the experimental plan of the user. It informs the Control Unit of the experiment design details needed at each stage of the simulation. As the user is allowed to specify a number of experiments which are run sequentially, the Experimental Design component consists of a set of Experiment objects, each holding the design details for one experiment including: Number of replications, workload size (number of EBs to instantiated), measurement interval type and a scale factor contributing to the determination of the bookstore database size (see Section 3). Experimental data gathering and analysis is the responsibility of the Result Analysis Unit. It consists of a set of Experiment Results objects holding the results for each experiment executed in the simulation. Each Experiment Results object is linked with an Experiment object and contains a set of Run objects holding the results of each experiment run. During an experiment run, experimental data is recorded locally by different parts of the

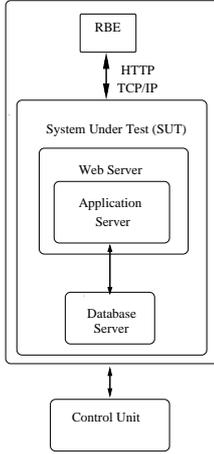


Figure 1: Tool Overall Design

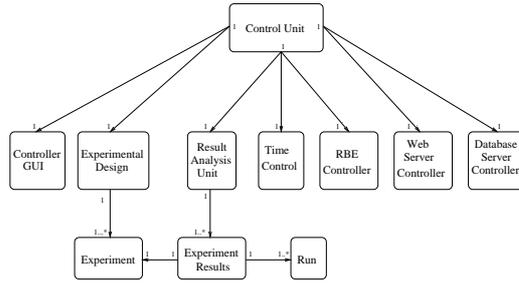


Figure 2: Control Unit

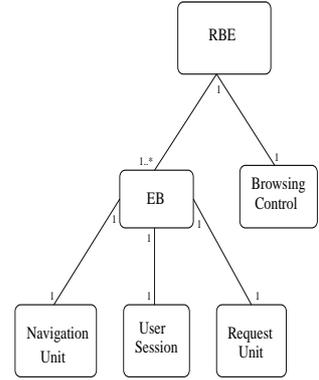


Figure 3: RBE

system minimising data recording overhead. On completion of the run, the Result Analysis Unit receives a request from the Control Unit to gather the experimental data which are then stored in a Run object. Data analysis is done after the completion of each experiment. Performance metrics provided include overall average throughput, overall average response time and individual interactions average response time (see Section 6). Time Control is the central timing component of the tool. It distributes timing information to all parts of the system. The Control Unit also includes three controller components, RBE Controller, Web Server Controller and Database Server Controller for controlling the RBE, Web Server and Database Server respectively. Instructed by the Control Unit, these components configure and control the running of experiments on the parts of the tool they are responsible for. Controlling an extension to the tool requires only adding a new controller to the Control Unit.

RBE RBE is the component responsible for driving the tool workload. As shown in Figure 3, it includes a set of EBs which emulate web browsing e-commerce users requesting web interactions from a web server as specified in the TPC-W benchmark (see Section 3). Its Navigation Unit emulates the user’s navigational behaviour. The decision on which navigation option to select next is based on the current web interaction just completed and its EB navigation option thresholds, specified by TPC-W (see Section 3). The Navigation Unit generates a random Think Time period which is spent sleeping between web interaction requests. The EB User Session sub-module is designed to emulate the maintenance of e-commerce user-specific information in a web browser including session tracking details and customer identification. Cookies was the method adopted for session tracking. The sending and receiving of HTML content via HTTP and TCP/IP is emulated by the Request Unit sub-module. It is responsible for forming HTTP requests for the different web interactions specified in TPC-W. It is also responsible for maintaining HTTP/TCP/IP connections with the web server and associating these connections with User Ses-

sions. In addition, the Request Unit log data statistics about web interactions requested and completed successfully including the starting and finishing time of a web interaction, its type and how many times it has been requested in an experiment run.

EBs are created and maintained by the Browsing Control component. It receives instructions from the Control Unit (via its RBE Controller) on the measurement interval type and the number of EBs required for each experiment. When creating an EB, Browsing Control provides it with the type of user it presents (customer, new user, or site administrator) which is selected randomly. The measurement interval type is also provided to the EB. Browsing Control also informs EBs of the start/end of a measurement interval and/or an experiment run (as ordered by the Control Unit). Data recorded by the Request Unit is collected by Browsing Control to be passed to the Control Unit.

Web and Database Servers The Web and Database Servers represent the system tested by the workload. The Web Server is responsible for serving HTTP requests for the 14 dynamic and static web interactions specified in TPC-W (see Section 3). It contains the application code for the dynamic web interactions, implemented by the programming technologies currently benchmarked by the tool. The HTML code for the static web interaction and images for different interactions also reside in the web server.

The Database Server contains the *bookstore* database with a schema following exactly the one specified in TPC-W (see Section 3). The Database Server Controller of the Control Unit populates/de-populates the database according to TPC-W specifications.

5 IMPLEMENTATION

In this section we discuss the tool implementation issues. First, the decision to implement the tool in the Java language is discussed. The use of Apache as the web server of the tool is then described and the choice of MySQL to manage the bookstore database is discussed. This is followed by a description of the support

provided for e-commerce programming technologies.

Java (SDK 1.4.1) is used to implement the tool. Being a pure object-oriented programming language, Java ensured the realisation of the tool's modular design. The implementation in a portable programming language as Java resulted in a platform-independent tool which can be deployed on machines of different architectures across a network. Java's `java.net` package, with its URL and socket classes, provided a neat implementation of the interactions between the EBs and the Web Server components (see Section 4). The reliable `System.currentTimeMillis()` method, which is part of Java's `java.lang` package is used to get timestamps from various parts of the system needed to produce performance metrics. `System.currentTimeMillis()` is quick with almost no overhead, thus enforcing the realistic nature of the simulation.

A web server is needed to serve HTTP requests and host the application server (see Section 4). The Apache web server (Version 1.3) is used. It is a freeware web server and is the choice of the majority of active site developers. A survey by Netcraft [9] on October 2001 concluded that 61% of the active sites they monitor use Apache. The way Apache is designed was another reason for choosing it. It is built around an API which allows third-party programmers to add new server functionality. Everything in Apache is implemented as one or several modules, using the same extension API available to third parties. Thus, extending the tool to provide support for a new e-commerce programming technology requires implementing a new Apache module and incorporating it using Apache's API.

A Database Management System (DBMS) is needed to host the bookstore database (see Section 4). It was decided to use MySQL (Version 3.23). MySQL is a popular DBMS and works well with Apache for different e-commerce programming technologies.

Currently, the tool provides application server implementations in two e-commerce programming technologies; Java Servlets [7] (using `mod_jserv` apache module) and PHP [10] (using `PHP` apache module). Providing benchmarking support for another e-commerce programming technology requires only finding/implementing a supporting Apache module and implementing the application code for that technology.

6 RESULTS

To demonstrate the capabilities of the tool, experiments were designed and implemented to compare the scalability of the Java Servlets and PHP application server implementations. The effect of varying the size of the workload on overall average throughput, overall average response time and individual web interaction average response time was examined.

Experiment Design The technologies involved in the experiments performed were Java Servlets and PHP. The size of the workload was varied by doubling the number of EBs at each stage with a minimum value

of 1 EB and a maximum value of 128 EBs. The scale of the database used was 1000. The total time interval of each experiment run was 400 sec. The tool allows for a *Rump up* period of $1/4 * \text{total time interval}$ (100 sec in this case) for EBs initialisation. The measurement interval (during which measurement is recorded) takes the remaining $3/4$ of the total time period (300 sec in this case). The measurement interval type used was Shopping reflecting a shopping scenario (see Section 3). Finally, each run was replicated 3 times. In all experiment runs, the workload generator RBE, the web server, the application server and the database server were running on a GenuineIntel Pentium III, 1 GHz, 265MB Linux Dell Machine.

Workload Effect on Throughput Figure 4 summarise the results obtained from examining the average throughput when the number of EBs is varied between 1 and 128. The average throughput is considered as the average number of web interactions completed successfully per second during the measurement interval. Figure 4 illustrates two points. Firstly, it can be seen that both Java Servlets and PHP scale well up to a workload size of 64 EBs before which the average throughput starts to degrade. Secondly, one can argue that Java Servlets scales slightly better as it degrades at a slower rate than PHP.

Workload Effect on Response Time Results obtained from varying the size of the workload and recording average response time are shown in Figure 5. The response time of a web interaction is considered as the time elapsed from the last byte received by the EB to compute a web interaction until the first byte sent by the EB to request the next interaction. Average response time is the total response time of all web interactions completed successfully divided by the number of these interactions. It can be seen from Figure 5 that the average response time starts to increase almost exponentially when the size of the workload increases above 64 EBs. Again, Java Servlets scale better with a slower rate of average response time increase. From Figure 4 and 5, one could argue that optimal performance is achieved at the 64 EBs workload, with maximum throughput and good response time

Workload Effect on Individual Web Interactions Response Time Figure 6 shows the average response time for the Shopping Cart interaction as an example of a dynamic web interaction. The average response time of the static interaction Search Request is illustrated in Figure 7. The average response time of an individual interaction is computed by totaling the response time of all occurrences of that interaction and dividing that by the number of times the interaction was requested and completed successfully. Again, it can be seen that the performance degrade point is when the workload size exceeds 64 EBs with Java Servlets degrading at a slower rate in both the static and dynamic interaction examples. It can also be noticed that the Shopping Cart Interaction (a dynamically built HTML page) average response time degrade faster than the Search Request Interaction (a static HTML page).

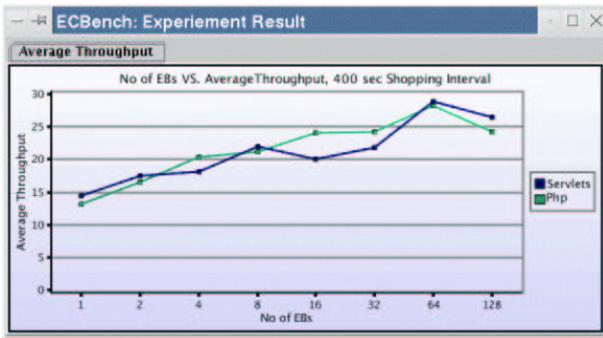


Figure 4: Overall Average Throughput

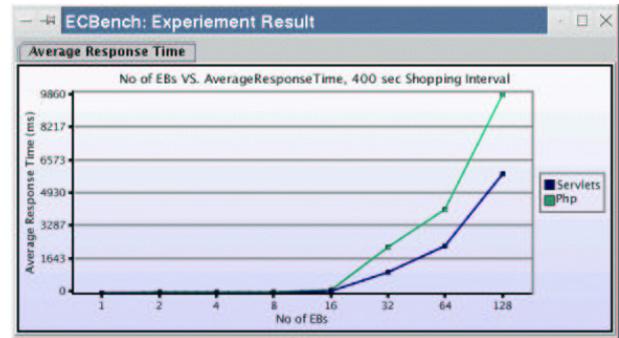


Figure 5: Overall Average Response Time

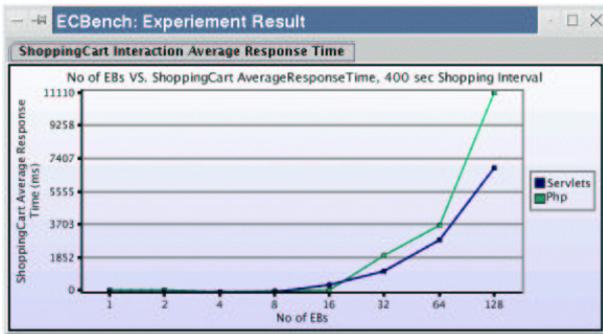


Figure 6: Shopping Cart Avg. Response Time

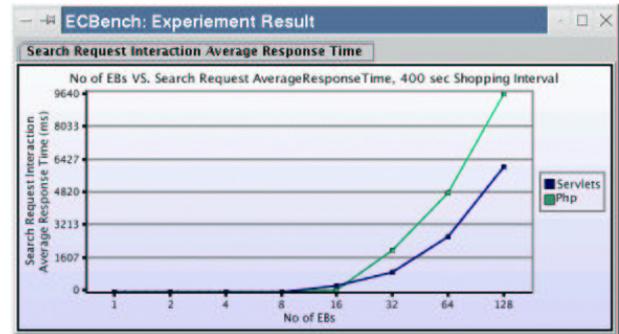


Figure 7: Search Request Avg. Response Time

7 CONCLUSIONS

A tool to allow the comparative benchmarking of e-commerce programming technologies has been developed. It is based on the TPC-W benchmark, ensuring the realistic modelling of e-commerce components. It provides a set of experiment support features allowing the user to design a set of experiments to be run sequentially by the tool. For each experiment, a number of replications can also be specified. A set of e-commerce relevant performance metrics are analysed and presented with minimal overhead on the simulation. This was achieved by adopting a local experimental data recording strategy during experiment runs. Data is gathered and analysed on completion.

The modular design of the tool widens the scope of further work. New modules can be incorporated to benchmark other programming languages/technologies. The RBE module can be enhanced to provide robots workload [12] in addition to remote browsers. The tool can also be extended with small modifications to allow for the benchmarking of other e-commerce components such as session tracking mechanisms, security and payment protocols.

References

- [1] Menasce D. A. and Almeida V. A. *Scaling for E-Business: Technologies, Models, Performance, and Capacity Planning*. Prentice Hall PTR, 2000.
- [2] Ince D. *Developing Distributed and E-commerce Applications*. Addison Wesley, 2002.
- [3] Whiteley D. *e-Commerce: Strategy, Technologies and Applications*. McGraw Hill, 2000.

- [4] Kruse M. <http://mkruse.netexpress.net/info/cgi>. Technical report.
- [5] Microsoft. <http://msdn.microsoft.com/library/psdk/iisref/aspguide.htm>. Technical report, Microsoft.
- [6] Sun Microsystems. <http://java.sun.com/products/jsp/whitepaper.html>. Technical report, Sun Microsystems.
- [7] Sun Microsystems. <http://java.sun.com/products/servlet/whitepaper.html>. Technical report, Sun Microsystems.
- [8] Abu Shaaban Y. N. and Hillston J. A TPC-W-based tool for benchmarking e-commerce programming technologies. In *Proc. 18th UK Performance Engineering Workshop, Glasgow, July 10-11, 2002*.
- [9] Netcraft. <http://www.netcraft.com/survey>. Technical report, Netcraft.
- [10] Bakken S. S., Aulbach A., Schmid E., Winstead J., Wilson L. T., Lerdorf R., Zmievski A., and Ahto J. <http://www.php.net/manual.en>. Technical report, The PHP Group.
- [11] TPC. <http://www.tpc.org/tpcw>. Technical report, Transaction Processing Performance Council.
- [12] Almeida V., Menasce D., Riedi R., Peligrinelli, Fonseca R., and Meira W. Analyzing robot behavior in e-business sites. In *Proc. 2001 ACM Sigmetrics Conference, June 16-20, 2001*.



Yussuf N. Abu Shaaban is a Ph.D. student in ICISA, School of Informatics, University of Edinburgh. He is part of the EPSRC-funded *Dependability IRC*. His research interest is performance evaluation of e-commerce systems.

Performance Modelling of Differentiated Services in 3G Mobile Networks

Irfan Awan and Khalid Al-Begain
Mobile Computing and Communications Research Group
Department of Computing, University of Bradford
BD7 1DP, Bradford, UK
{I.Awan, K.Begain}@bradford.ac.uk

Abstract

One of the main features of the third Generation (3G) mobile networks is their capability to provide different classes of services; especially multimedia and real-time services in addition to the traditional telephony and data services. These new services, however, will require higher Quality of Service (QoS) constraints on the network mainly regarding delay, delay variation and packet loss. Additionally, the overall traffic profile in both the air interface and inside the network will be rather different than used to be in today's mobile networks. Therefore, providing QoS for the new services will require more than what a call admission control algorithm can achieve at the border of the network, but also continuous buffer control in both the wireless and the fixed part of the network to ensure that higher priority traffic is treated in proper way. This paper proposes and analytically evaluates a buffer management scheme that is based on multi-level priority and Complete Buffer Sharing (CBS) policy for all buffers at the border and inside the wireless network. The analytical model is based on the G/G/1/N censored queue with single server and R ($R \geq 2$) priority classes under the Head of Line (HOL) service rule for the CBS scheme. The traffic is modelled using the Generalised Exponential distribution. The paper presents an analytical solution based on the approximation using the Maximum Entropy (ME) principle. The numerical results show the capability of the buffer management scheme to provide higher QoS for the higher priority service classes.

Keywords

3G mobile networks, performance evaluation, maximum entropy (ME) principle, queueing network model (QNM), generalised exponential (GE) distribution, head-of-line (HOL) discipline, complete buffer sharing (CBS) rule.

1 Introduction

The Third Generation (3G) mobile networks are under deployment in many regions in the world. In Europe, the Universal Mobile Telecommunications System (UMTS) has been implemented almost by every major mobile network operator offering new mostly multimedia based services. At the moment, the volume of data traffic in mobile networks is still moderate but it is obvious that as more and more customers join the new opportunities offered by 3G, the volume and the nature of the traffic at both the border and inside the mobile network will be very different. Furthermore, many of the new services will require more strict quality of service (QoS) guarantees than the simple data transmission, mainly regarding packet loss, delay and delay variation. Therefore, it is important to implement suitable algorithms to provide prioritization between services and to guarantee preferential treatment to the packets belonging

to the higher priority services. This leads to a traffic with Differentiated Services (DS).

Considering the UMTS architecture [1], two levels of QoS assuring algorithms can be identified. First, at the air interface (UTRAN), efficient call admission control algorithms (CACA) can be implemented [2] to limit the volume of the traffic entering the network. However, CACA are not enough to prevent congestions and delay inside the mobile or the fixed parts of the network. Additionally, due to the hierarchical architecture of UMTS, multiple traffic stream from different connections belonging to different classes of services will aggregate at the Gateway Servicing GPRS Node (GSGN) [1] which serves as a gateway towards the public data network (the Internet). In this node, there is a need to implement a buffer management scheme that provides differentiated service.

Finite buffer queues with service and space priorities

are of great importance towards effective congestion control and quality of service (QoS) protection in high speed telecommunication networks.

Many queueing systems with priorities have been explored by Cohen [3] and various applications of the analytical results of priority queues to data communication systems are surveyed by de Moraes [4]. A stable infinite capacity G/G/1 queue with a single server and priority classes under either Preemptive-Resume (PR) or Head-of-the-Line (HOL) scheduling disciplines has been analysed in [5], by applying the method of entropy maximisation (MEM). MEM has also been used in [6] to study a stable single class G/G/1/N censored queue with a single server and First-Come-First-Served (FCFS) scheduling discipline.

A stable G/G/1/N censored queue with a single server, finite capacity and priority classes under complete buffer sharing (CBS) scheme is an important building block in the performance of communication networks. The analysis of such queue is very difficult to tackle using the classical queueing theory. To the authors' knowledge, no exact or approximate closed-form solutions for a stable G/G/1/N censored queue with service priorities have appeared in the literature.

This paper presents further advances of maximum entropy (ME) towards the approximate analysis of a stable G/G/1/N censored queue with a single server, and, R ($R \geq 2$) priority classes under HOL service rule for CBS scheme.

The paper is organised as follows: The ME solution for a stable G/G/1/N censored queue with service priorities is characterised in Section 2. Marginal and aggregate performance distributions are presented in Section 3. Numerical validation results against simulation, involving Generalised Exponential (GE) interarrival and service time distributions, are included in Section 4. Section 5 includes conclusions and remarks for future work.

Remarks

- *The GE Distribution*

The GE distribution is an interevent-time distribution of the form

$$F(t) = P(W \leq t) = 1 - \tau e^{-\sigma t}, \quad t \geq 0, \quad (1)$$

$$\tau = 2/(C^2 + 1), \quad (2)$$

$$\sigma = \tau\nu, \quad (3)$$

where W is a mixed-time random variable (rv) of the interevent-time, whilst $(1/\nu, C^2)$ are the mean and squared coefficient of variation (SCV) of rv W . The GE distribution is versatile, possessing pseudo-memoryless properties which makes the solution of

many GE-type queueing systems and networks analytically tractable [7].

2 ME Analysis of GE/GE/1/N Priority Queue

Consider a single server finite capacity GE/GE/1/N queue at equilibrium with R ($R > 1$) distinct priority classes of jobs (indexed from 1 to R in descending order of priority) such that

- the total buffer capacity is N for a CBS scheme.
- the interarrival and service times per class are distributed according to a GE distribution under HOL service rule in conjunction with CBS buffer management scheme.

Notation

For each class i ($i = 1, 2, \dots, R$), let λ_i be the mean arrival rate, $C_{a_i}^2$ be the interarrival time SCV, μ_i be the mean service rate and $C_{s_i}^2$ be the service time SCV.

Focusing on a stable GE/GE/1/N queue, let at any given time

n_i ($0 \leq n_i \leq N$) be the number of class i ($i = 1, 2, \dots, R$) jobs in the queue (waiting and/or receiving service)

$\mathbf{S} = (n_1, n_2, \dots, n_R, \omega)$ be a joint queue state, where ω ($1 \leq \omega \leq R$) denotes the class of the current job in service and $\sum_{i=1}^R n_i \leq N$ (n.b., for an idle queue $\mathbf{S} \equiv \mathbf{0}$ with $\omega = 0$)

\mathbf{Q} be the set of all feasible states \mathbf{S}

$\mathbf{n} = (n_1, n_2, \dots, n_R)$ be an aggregate joint queue state (n.b., $\mathbf{0} = (0, \dots, 0)$)

$\mathbf{\Omega}$ be the set of all feasible states \mathbf{n}

Remarks

- The arrival process for each class i ($i = 1, 2, \dots, R$) is assumed to be censored i.e., a job of class i ($i = 1, 2, \dots, R$) will be lost if on arrival finds N jobs at the queue.

2.1 Prior Information

For each state \mathbf{S} , $\mathbf{S} \in \mathbf{Q}$, and class i ($i = 1, 2, \dots, R$) the following auxiliary functions are defined:

$$\begin{aligned} n_i(\mathbf{S}) &= \text{the number of class } i \text{ jobs present in state } \mathbf{S}, \\ s_i(\mathbf{S}) &= \begin{cases} 1, & \text{if } \omega = i, \\ 0, & \text{otherwise.} \end{cases} \\ h_i(\mathbf{S}) &= \begin{cases} 1, & \text{if } n_i(\mathbf{S}) > 0, \\ 0, & \text{otherwise.} \end{cases} \\ f_i(\mathbf{S}) &= \begin{cases} 1, & \text{if } \sum_{i=1}^R n_i(\mathbf{S}) = N \text{ \& } \omega = i, \\ 0, & \text{otherwise.} \end{cases} \end{aligned}$$

Suppose all that is known about the state probabilities $\{P(\mathbf{S})\}$ is that they satisfy the

- Normalisation constraint

$$\sum_{\mathbf{S} \in \mathbf{Q}} P(\mathbf{S}) = 1, \quad (4)$$

and that the following marginal mean value constraints per class i exist:

- Server utilisation, U_i ($0 < U_i < 1$),

$$\sum_{\mathbf{S} \in \mathbf{Q}} s_i(\mathbf{S}) P(\mathbf{S}) = U_i, \quad i = 1, 2, \dots, R; \quad (5)$$

- Busy state probability per class, θ_i ($0 < \theta_i < 1$),

$$\sum_{\mathbf{S} \in \mathbf{Q}} h_i(\mathbf{S}) P(\mathbf{S}) = \theta_i, \quad i = 1, 2, \dots, R; \quad (6)$$

- Mean queue length, L_i ($U_i \leq L_i < N$),

$$\sum_{\mathbf{S} \in \mathbf{Q}} n_i(\mathbf{S}) P(\mathbf{S}) = L_i, \quad i = 1, 2, \dots, R; \quad (7)$$

- Full buffer state probability, ϕ_i ($0 < \phi_i < 1$),

$$\sum_{\mathbf{S} \in \mathbf{Q}} f_i(\mathbf{S}) P(\mathbf{S}) = \phi_i, \quad i = 1, 2, \dots, R; \quad (8)$$

satisfying the flow balance equations, namely

$$\lambda_i(1 - \pi_i) = \mu_i U_i, \quad i = 1, 2, \dots, R; \quad (9)$$

where π_i is the blocking probability that an arriving job of class i finds the queue full.

The choice of mean value constraints (4) - (8) is based on the type of constraints used for the ME analysis of a stable multiple class queues with or without priorities (c.f., [7, 8, 9]). Note that if additional constraints are used, it is no longer feasible to capture a computationally efficient ME solution in closed-form. As a consequence, this will have adverse implications towards the creation of a cost-effective queue-by-queue decomposition algorithm for arbitrary queueing network models (QNMs). Conversely, the removal of one or more constraints from the set (4) - (8) will result into a ME solution of reduced accuracy.

2.2 A Universal Maximum Entropy Solution

A universal form of the state probability distribution $\{P(\mathbf{S}), \mathbf{S} \in \mathbf{Q}\}$ can be characterised by maximising the entropy functional

$$H(P) = - \sum_{\mathbf{S}} P(\mathbf{S}) \log P(\mathbf{S}), \quad (10)$$

subject to constraints (4) - (8). By employing Lagrange's method of undetermined multipliers, the ME solution is expressed by

$$P(\mathbf{S}) = \frac{1}{Z} \prod_{i=1}^R g_i^{s_i(\mathbf{S})} \xi_i^{h_i(\mathbf{S})} x_i^{n_i(\mathbf{S})} y_i^{f_i(\mathbf{S})}, \quad \forall \mathbf{S} \in \mathbf{Q}; \quad (11)$$

where Z , the normalising constant, is clearly given by

$$Z = \sum_{\mathbf{S} \in \mathbf{Q}} \left(\prod_{i=1}^R g_i^{s_i(\mathbf{S})} \xi_i^{h_i(\mathbf{S})} x_i^{n_i(\mathbf{S})} y_i^{f_i(\mathbf{S})} \right), \quad (12)$$

and $\{g_i, \xi_i, x_i, y_i, i = 1, 2, \dots, R\}$ are the Lagrangian coefficients corresponding to constraints (5) - (8), respectively.

Remarks

Although constraints (5) - (8) are not known priori, nevertheless it is assumed that these constraints exist. This information, therefore, has been incorporated into the ME formalism (4) - (10) in order to characterise the form of the joint state probability (11). An efficient computational implementation of the ME solution (11), however, requires the prior estimation of the Lagrangian coefficients. This can be achieved by making GE-type buffer size invariance assumptions with regard to Lagrangian coefficients $\{g_i, \xi_i, x_i, i = 1, 2, \dots, R\}$ together with asymptotic connections to an infinite capacity GE/GE/1 queue (c.f., [7]).

Aggregating (11) over all feasible states $\mathbf{S} \in \mathbf{Q}$, and after some manipulation, the joint aggregate ME queue length distribution $\{P(\mathbf{n}), \mathbf{n} \in \Omega\}$ is given by:

$$P(\mathbf{0}) = \frac{1}{Z}. \quad (13)$$

$$\begin{aligned} P(\mathbf{n}) &= \frac{1}{Z} \left(\prod_{i=1}^R x_i^{n_i} \xi_i^{h_i(\mathbf{n})} \right) \left(\sum_{j=1 \wedge n_j > 0}^R g_j y_j^{f_j(\mathbf{n})} \right), \\ &\forall \mathbf{n} \in \Omega - \{\mathbf{0}\}; \end{aligned} \quad (14)$$

where $h_i(\mathbf{n}) = 1$, if $n_i > 0$, or 0 otherwise and $f_i(\mathbf{n}) = 1$, if $\sum_{j=1}^R n_j = N$, or 0, otherwise, for $i = 1, 2, \dots, R$.

2.3 Recursive Relationships

Taking advantage of the ME product-form solution (13)-(14) and applying the generating function approach [10], recursive expressions for marginal utilisations $\{U_i, i = 1, \dots, R\}$, aggregate state probabilities $\{P(n), n = 0, \dots, N\}$, marginal state probabilities $\{P_i(n_i), n_i = 0, 1, \dots, N\}$ and marginal mean queue lengths $\{L_i, i = 1, \dots, R\}$ can be obtained.

2.3.1 Marginal Utilisations

It can be observed that the marginal utilisations $\{U_i, i = 1, \dots, R\}$ are clearly defined by $U_i = \sum_{\mathbf{S} \in \mathbf{Q}} s_i(\mathbf{S})P(\mathbf{S})$ and after some manipulation, they take the following universal form, namely

$$U_i = \frac{1}{Z} g_i \xi_i x_i \left(\sum_{v=1}^N y_i^{\gamma(v)} C^{(i)}(v-1) \right), \quad i = 1, 2, \dots, R; \quad (15)$$

where Z is the normalising constant and can be derived from the above equation (15) as follows:

$$Z = 1 + \sum_{i=1}^R g_i \xi_i x_i \left(\sum_{v=1}^N y_i^{\gamma(v)} C^{(i)}(v-1) \right). \quad (16)$$

where $\gamma(v) = 1$, if $v = N$ or 0 , otherwise. $C^{(i)}(v)$ can be calculated recursively using the following expressions:

$$C^{(i)}(v) = (1 - \xi_i) x_i C^{(i)}(v-1) + C(v) - x_i^N C(v-N) + x_i^{N+1} \xi_i C^{(i)}(v-N-1), \quad (17)$$

for $v = 1, 2, \dots, N-1$; $i = 1, 2, \dots, R$ with initial conditions

$$C^{(i)}(v) = \begin{cases} 0, & v < 0, \\ 1, & v = 0, \end{cases}$$

where $C(v) = C_R(v)$ and

$$C_r(v) = x_r C_r(v-1) + C_{r-1}(v) - (1 - \xi_r) x_r C_{r-1}(v-1) - \xi_r x_r^{N+1} C_{r-1}(v-N-1), \quad (18)$$

$r = 1, 2, \dots, R$ with initial conditions

$$C_r(v) = \begin{cases} 0, & v < 0, \\ 1, & v = 0, \\ \xi_1 x_1^v, & v > 0. \end{cases}$$

2.3.2 Marginal State Probabilities

Aggregating ME solution $\{P(\mathbf{S}), \mathbf{S} \in \mathbf{Q}\}$ and defining an appropriate z-transform [10], after some manipulation, the following recursive expressions for the marginal probabilities can be obtained:

$$P_i(0) = \frac{1}{Z} \left(1 + \sum_{j=1 \wedge j \neq i}^R g_j \xi_j x_j \sum_{v=0}^{N-1} C_i^{(j)}(v) y_j^{\delta(v)} \right), \quad (19)$$

$$P_i(n) = \frac{1}{Z} \xi_i x_i^n \left(\sum_{j=1}^R g_j E_j \sum_{k=1 \wedge k \neq i}^R \sum_{v=0}^{N-n-F} C_i^{(j)}(v) y_j^{\delta(v)} \right), \quad (20)$$

where $E_j = \xi_j x_j$ if $j \neq i$ or 1 ow, $F = 1$ if $j \neq i$ or 0 ow.

The coefficients $\{C_i^{(j)}(v), v = 0, 1, \dots, N-1, (i, j) \in [1, R]\}$ can be determined by the following recursive formulae:

$$C_i^{(j)}(v) = \begin{cases} C^{(j)}(v) - x_i C^{(j)}(v-1) + (1 - \xi_i) x_i C_i^{(j)}(v-1) + \xi_i x_i^{N+1} C_i^{(j)}(v-N-1), & i \neq j \\ C^{(j)}(v) - x_i C^{(j)}(v-1) + x_i^N C_i^{(j)}(v-N), & i = j \end{cases} \quad (21)$$

with initial condition $C_i^{(j)}(v) = 0$ if $v < 0$, or 1 , if $v = 0$, where $C^{(j)}(v)$ is determined by (17).

2.4 The Blocking Probability

A universal form for the marginal blocking probabilities $\{\pi_i, i = 1, 2, \dots, R\}$ of a stable multiple class GE/GE/1/N queue can be approximately established, based on GE-type probabilistic arguments, by the following expression:

$$\pi_i = \frac{1}{Z} \left(\sum_{v=0}^N \delta_i(v) (1 - \sigma_i)^{N-v} P(v) \right), \quad (22)$$

where $\delta_i(v) = \frac{r_i}{r_i(1-\sigma_i) + \sigma_i}$, $\sigma_i = 2/(1 + C_{a_i}^2)$ and $r_i = 2/(1 + C_{s_i}^2)$ and $P(v)$ are the aggregate probabilities.

2.5 The Lagrangian Coefficients

It is assumed, as in earlier works (c.f., [7, 8, 9]), that the Lagrangian coefficients $\{g_i, \xi_i, x_i, i = 1, \dots, R\}$ of the ME solution (11) for GE-type queues and networks are largely invariant to the buffer threshold size N_i ($i = 1, 2, \dots, R$). These coefficients can be, therefore, approximated via closed form asymptotic queueing theoretic expressions based on the ME solution of the corresponding infinite capacity GE/GE/1 queue at equilibrium (c.f., [5]). Using the flow balance condition (9) and the closed-form expressions for the normalising constant, Z , the aggregate probabilities $\{P(n), n = 0, 1, \dots, N_1\}$ and the blocking probabilities $\{\pi_i, i = 1, \dots, R\}$, the Lagrangian coefficients $\{y_i, i = 1, 2, \dots, R\}$ can be recursively determined (c.f., [11]):

3 Numerical Results

This section presents typical numerical experiments in order to illustrate the credibility of the proposed ME solution against simulation. Moreover, it demonstrates the applicability of ME results as simple but cost-effective performance evaluation tools for assessing the effect of external multiple class GPRS traffic at the GE/GE/1/N/HOL queue.

The numerical study focuses on two data packet classes representing typical Internet applications, namely, 12.5 KBytes (class 1, e.g., email) and 62.5 KBytes (class 2, e.g., web browsing) and , respectively. The parameterization also involves mean arrival rates and SCV of inter-arrival and service times. It is assumed that the GPRS partition consists of one frequency providing total capacity of 171.2 Kbps. Without loss of generality, the evaluation study focuses on marginal performance metrics of utilisation, mean response time and mean queue length per class. Numerical tests are carried out to verify the relative accuracy of the ME algorithm against simulation at 95 % confidence intervals based on the Queueing Network Analysis Package (QNAP-2) [12], using the same assumptions and input parameterization as the ones used for the analytic ME solution (c.f., Figs. 1-3). It can be observed that the ME results are very comparable to those obtained via simulation. Moreover, it can be seen that the interarrival-time SCV has an inimical effect, as expected, on the mean response time per class (c.f., Fig. 3). Results shows that high priority calls face less mean response times as compared to the low priority calls.

Moreover, relative comparisons to assess the impact at varying degrees of interarrival time SCVs and buffer size, N , at the GE/GE/1/N/HOL queue upon ME generated mean queue lengths are presented in Figs. 4-5, respectively. It can be seen that the analytically established mean queue lengths deteriorate rapidly with increasing external interarrival-time SCVs (or, equivalently, average batch sizes) beyond a specific critical value of the buffer size which corresponds to the same mean queue length for two different SCV values. It is interesting to note, however, that for smaller buffer sizes in relation to the critical buffer size and increasing mean batch sizes, the mean queue length steadily improves with increasing values of the corresponding SCVs. This ‘buffer size anomaly’ can be attributed to the fact that, for a given arrival rate, the mean batch size of arriving bulks increases whilst the interarrival time between batches increases as the interarrival time SCV increases, resulting in a greater proportion of arrivals being blocked (lost) and, thus, a lower mean effective arrival rate; this influence has much greater impact on smaller buffer sizes.

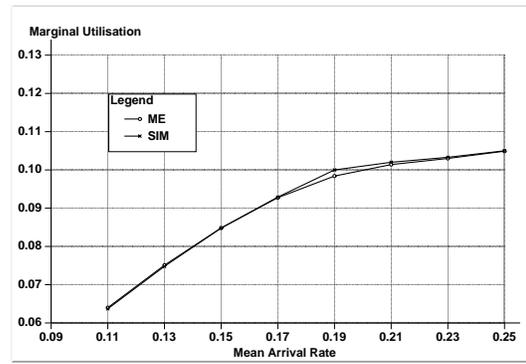


Figure 1: Marginal Utilisations for Class 1 Calls

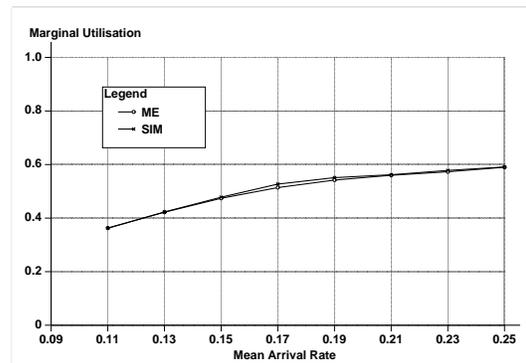


Figure 2: Marginal Utilisations for Class 2 Calls

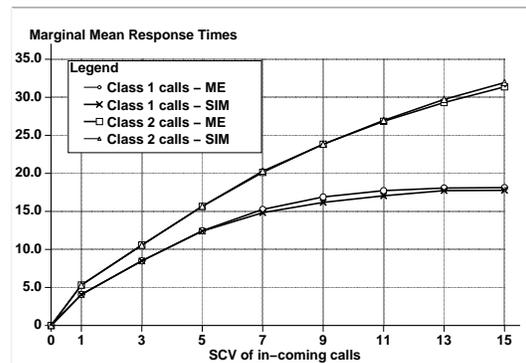


Figure 3: Effect of varying degrees of SCV on Mean Response Time

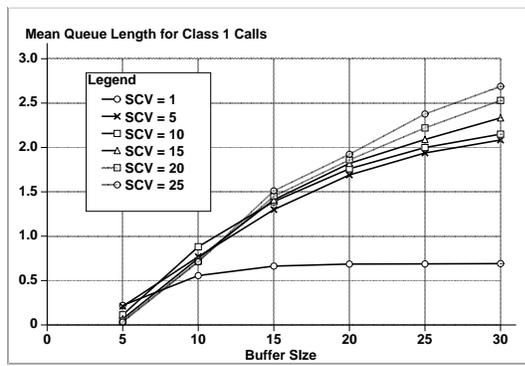


Figure 4: Effect of varying degrees of SCV on MQLs of Class 1 at different buffer sizes

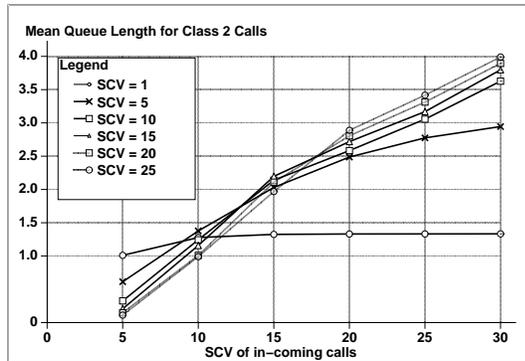


Figure 5: Effect of varying degrees of SCV on MQLs of Class 2 at different buffer sizes

4 Conclusions

This paper presents an analytical model to evaluate performance of the wireless network where different applications are given preferential treatment in order to provide quality of service. In this context, a G/G/1/N censored queue with single server and R ($R \geq 2$) priority classes under the HOL service rule for the CBS scheme has been analysed using ME principle. Closed form expressions for marginal utilizations, state probabilities and blocking probabilities are presented. Typical numerical experiments show the capability of the buffer management scheme to provide higher QoS for the higher priority service classes.

References

[1] Bernhard H. Walke *Mobile Radio Networks, Networking and Protocols*, . WILEY.
 [2] K. Al-Begain and A. Zreikat ,”Interference Based CAC for Up-link Traffic in UMT S

Networks”, in *Proceedings of World Wireless Congress, 2002*, 28-31 June 2002, pp. 298 - 303, San Francisco, USA.

[3] Cohen, J. W. The Single Server Queue. Revised edition, *North-Holland Publishing Company*, Amsterdam (First edition: 1969).
 [4] de Moraes, L. F. M. Priority Scheduling in Multiaccess Communication, Stochastic Analysis of Computer and Communication Systems, H. Takagi (ed.), *Elsevier Science Publishers* (North-Holland), Amsterdam, pp. 699-732, 1990.
 [5] D.D. Kouvatsos and N.M. Tabet-Aouel, A Maximum Entropy Priority Approximation for a Stable G/G/1 Queue, *Acta Informatica* 27, (1989), pp. 247-286.
 [6] D.D. Kouvatsos, Maximum Entropy and the G/G/1/N Queue, *Acta Informatica*, Vol. 23, (1986), pp. 545-565.
 [7] D.D.Kouvatsos, Entropy Maximisation and Queueing Network Models, *Annals of Operation Research* 48, (1994), pp. 63-126.
 [8] D.D.Kouvatsos and I.U.Awan, MEM for Arbitrary Closed Queueing Networks with RS-Blocking and Multiple Job Classes, *Annals of Operations Research* 79, (1998), pp. 231-269.
 [9] D.D.Kouvatsos and Xenios N.P., MEM for Arbitrary Queueing Networks with Multiple General servers and repetitive-Service Blocking, *Performance Evaluation*, Vol. 10, (1989), pp. 169-195.
 [10] A.C.Williams and R.A.Bhandiwad, A Generating Function Approach to Queueing Network Analysis of Multiprogrammed Computers, *Networks* 6, (1976), pp. 1-22.
 [11] I.U. Awan and D.D.Kouvatos, Maximum Entropy Analysis of Arbitrary Queueing Network Models with Service priorities, *Research Report RS-07-01*, Performance Modelling and Engineering Research Group, Department of Computing, University of Bradford, August, (2001).
 [12] M. Veran and D. Potier, QNAP-2: A Portable Environment for Queueing Network Modelling Techniques and Tools for Performance Analysis, D.Potier (ed.), North Holland, pp. 25-63, 1985.

BIOLOGY & MEDICINE

FORMING OF CONTROLLED LIVING MICROENVIRONMENTS

ALEXANDER A. AMELKIN

Moscow State University of Food Production

11 Volokolamskoye shosse, Moscow 125080, Russia

E-mail: aaamelkin@mtu-net.ru; URL: <http://www.angelfire.com/pe/daptap99/amelkin.html>

Abstract

The purpose of the present work is to work out an approach for the development of software and the choice of hardware structures when designing subsystems for automatic control of technological processes realized in living objects containing limited space (microenvironment). The subsystems for automatic control of the microenvironment (SACME) under development use the Devices for Air Prophylactic Treatment, Aeroionization, and Purification (DAPTAP) as execution units for increasing the level of safety and quality of agricultural raw material and foodstuffs, for reducing the losses of agricultural produce during storage and cultivation, as well as for intensifying the processes of activation of agricultural produce and industrial microorganisms. A set of interconnected SACMEs works within the framework of a general microenvironmental system (MES). In this research, the population of baker's yeast is chosen as a basic object of control under the industrial fed-batch cultivation in a bubbling bioreactor. This project is an example of a minimum cost automation approach. The microenvironment optimal control problem for baker's yeast cultivation is reduced from a profit maximum to the maximization of overall yield by the reason that the material flow-oriented specific cost correlates closely with the reciprocal value of the overall yield. Implementation of the project partially solves a local sustainability problem and supports a balance of microeconomical, microecological and microsocioal systems within a technological subsystem realized in a microenvironment maintaining an optimal value of economical criterion (e.g. minimum material, flow-oriented specific cost) and ensuring: (a) economical growth (profit increase, raw material saving); (b) high security, safety and quality of agricultural raw material during storage process and of food produce during a technological process; elimination of the contact of gaseous harmful substances with a subproduct during various technological stages; (c) improvement of labor conditions for industrial personnel from an ecological point of view (positive effect of air aeroionization and purification on human organism promoting strengthened health and an increase in life duration, pulverulent and gaseous chemical and biological impurity removal). An alternative aspect of a controlled living microenvironment forming is considered.

Keywords: Aeroionizer; agricultural raw materials; agriculture; air purifier; baker's yeast; barley; environmental engineering; feed and aeration rates; feedback control system; food processing; material flow-oriented specific cost; mathematical model; microenvironment; overall yield

Introduction

One of the most important tasks of the food and processing branches of the agroindustrial complex is the development and introduction of progressive technological processes, equipment and control systems providing an increase in quality and biological value of the foodstuffs. The significant part of such technological processes is realized

in a limited space (microenvironment) containing living objects (for example, cultivation of baker's yeast, storage and transportation of fruits, vegetables, barley and other kinds of agricultural raw material, malting, activation of yeast before fermentation, green sprouting of potatoes before planting, suppression of activity of mould, vermin (insects, acarina), and putrefactive microflora during foodstuff storage, etc.). The abovementioned targets could be partially achieved by MES project introduction at food and agricultural enterprises.

Attention must be paid to following an essential feature of this project concerning a sustainable development problem treatment on a local level. By applying SACMEs at food manufacturing enterprises operating with living objects (such as breweries, bakeries, biotechnological productions, etc.), for example, a constructive compromise could be achieved in a simultaneous solution of three problems with no contradictions arising: economical growth (productivity increase, raw material saving); ensuring security, safety and quality of agricultural raw material during storage process and of food produce during production technological process; improvement of labor conditions for industrial personnel from an ecological point of view.

In brewing the MES project introduction will result, for instance, in:

- brewer's barley, rice, maize, hops, and malt storage period increase and better preservation (losses decrease, quality increase) during storing;
- putrefactive microflora and mould elimination;
- air purification in storehouse and other industrial departments of dust, pathogenous microorganisms, harmful gaseous impurities;
- brewer's barley germination (box or drum malting) stimulation (But 1977);
- brewer's yeast fermentation intensification;
- elimination of gaseous harmful substance contacts with raw material and beer during various technological stages;
- positive effect of air aeroionization and purifying on industrial personnel promoting strengthened health and life duration increase.

The SACME systems could be fulfilled in a stationary performance (for large industrial areas and big volumes of produce to be processed) and in portable or transport modifications for the case of small processing capacities, petty warehouse premises, and also for installation on transport facilities destined for raw material (potatoes, fruit, barley) and ready products (malt, hop, yeast, bread) for long-distance deliveries.

In this work, the process of industrial baker's yeast cultivation as a basic object for SACME development is chosen with an investigation of possibility of result application for other objects, such as, for instance, the potato storehouse.

The use of DAPTAP through the effects of the various aeroion concentrations on intracellular respiration is offered as an execution device for the purpose of living object control. The mechanisms of aeroionized media influence on living systems are considered in other works (Chizhevsky 1999, Lifshitz 1990, Muzychenko 1991, Temnov et al. 2000).

As the other basic controls in baker's yeast cultivation in a bioreactor, the flow of aerating air can be used as a source of oxygen and the flow of molasses solution as a source of sugars. As the basic measurable and controlled parameters of living systems, the specific metabolic heat generation rate of a living object and the rate of the metabolic by-product formation (ethanol in yeast production and ethylene during agricultural produce storage) can be chosen.

The solution of the control problems was realized with the use of a complex mathematical model, describing a living system on mitochondrial and cellular/population levels and on the level of interaction of the population with the microenvironment.

In the present paper, some results of previous research are used (Amelkin et al. 2003, Amelkin et al. 2001a-2001c, Amelkin et al. 2000a-2000b, Amelkin and Amelkin 1997, Amelkin and Amelkin 1996).

1 Process Modelling

The mathematical model of a bioreactor as an example of a controlled microenvironment containing a living system (a population of yeast), consists of *three subsystems* - a model of intracellular respiration (a model of mitochondrial respiratory chain), intermediate model describing intercoupling of cellular and population levels, and a model of a bioreactor (microenvironment). Hereafter, these three modelling levels will be indicated as Model 1, Model 2, and Model 3, correspondingly.

Below, these three levels of microenvironment modelling are considered in detail.

2.1 A model of intracellular respiration (Model 1)

Two types of respiration exist for a living, aerobic organism - external respiration and cellular respiration. Cellular (mitochondrial) respiration is the process of oxidizing food molecules (like carbohydrates) to carbon dioxide and water. Biochemical oxidation is catalyzed by intracellular (intermitochondrial) enzymes and is the mechanism for obtaining energy from fuels (food molecules). The energy released is stored in the form of ATP for use by all the energy-consuming processes of the

cell. Actually 95% of the ATP is produced in the mitochondria. That is why mitochondria are often called 'the cell's power station'. The main and the most complicated part of the total mitochondrial respiration process is the respiratory chain or respiration system, which is based on the inner mitochondrial membrane. Here, most of the ATP is generated due to the proton gradient that is developed across this inner membrane.

The respiratory chain of the mitochondrion (**Fig.1**) consists of three large enzyme complexes built into the inner membrane, which serve as electron carriers (Alberts et al. 1989, Skulachev 1994, Skulachev 1989):

I. *NADH dehydrogenase complex* includes Fe-S centers as well as FMN bound with NADH dehydrogenase, and CoQ.

II. *Electron transport complex*, which is presented by cytochromes b-c₁-c, i.e. iron containing proteins transferring electrons from NADH dehydrogenase complex to cytochrome oxidase complex.

III. *Cytochrome oxidase complex* contains two cytochromes a-a₃ and two copper atoms. It is the site at the end of the mitochondrial respiratory chain. This site is the terminal accumulator of electrons carrying them directly to oxygen.

Beside electron carriers, the respiratory chain also contains several ATP synthase complexes.

The respiration system of mitochondria can be regarded as a biochemical generator with hydrogen electrode (enzyme complex I with potential ϕ_1) and oxygen electrode 2 (enzyme complex III with potential ϕ_3) assuming that intermediate redox pairs of respiratory chain play a regulatory role for redox processes in the mitochondrion. A similar assumption is already known (Volkenshtein 1988).

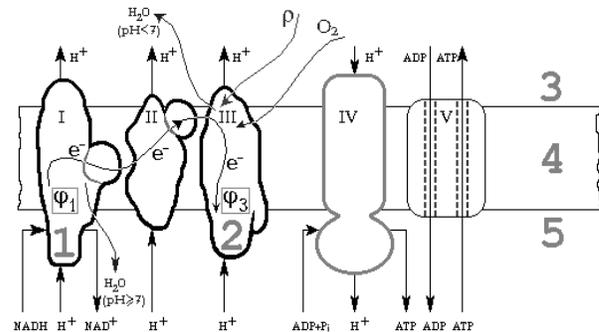


Fig.1: Respiration system of mitochondria as a biochemical generator with hydrogen electrode 1 and oxygen electrode 2 (enzyme complexes: I - *NADH dehydrogenase complex*; II - *Electron transport complex*; III - *Electron transport complex*; IV - *ATP synthase complex*; V - *Adenine nucleotide transporter*). Enzyme complexes I-V are built into the the inner membrane 4, separating matrix 5 from intermembrane space 3.

The mathematical description (Model 1) of the respiratory chain is a system of dynamic equations and kinetic expressions describing the electrochemical and biological processes of respiration occurring in a living organism on a cellular (mitochondrial) level. The Model 1 describes mechanisms of pH oscillation, proton and electron transport, oxidative phosphorylation, dismutation of

superoxide radicals, and superoxide dismutase (SOD) activation.

The pH oscillations phenomenon takes place in matrix and, in case of pH oscillation, center shift into extremely alkaline or acid zone metabolism is retarded (this can occur under definite environmental parameter variation conditions). The above mentioned oscillation processes during respiration and a level of SOD activity can be controlled by the rate of income of superoxide radicals from the execution device DAPTAP to mitochondrial matrix.

2.2 An Intermediate Model (Model 2) Describing Intercoupling of Cellular and Population Levels

With the aim of the cellular and population mathematical models coupling it is necessary to build the intermediate Model 2 describing relationships between the main parameters for different levels (Amelkin et al. 2000b, Wolf et al. 2000).

Such intermediate model linking population and cellular levels is destined to describe links between concentrations of the key components in cultural liquid (or in ambient medium) and flows to mitochondrion and its respiratory chain:

- flows of protons and of NADH to the respiratory chain of mitochondrion are determined by the Krebs cycle action and is linked with sugar concentration in the cultural liquid value;
- flow of molecular oxygen to the respiratory chain of mitochondrion is linked with dissolved oxygen concentration in the cultural liquid;
- flow of superoxide radicals is determined by superoxide radical concentration in cultural liquid, which in its turn is linked with negative aeroions concentration in aeration air;
- flows of ADP, of inorganic phosphate and of Ca^{2+} and Na^+ cations are controlled by pumping processes and other factors.

Model 2 will describe such parts of cell metabolism as Glycolysis, Acetyl-CoA Pathway, Krebs Cycle, as well as transport, dynamic and quantitative links of these parts with population level, on the one hand, and mitochondria inner membrane level (electron and ion transport, and oxidative phosphorylation systems) on the other hand.

2.3 Population Model (Model 3)

The population model is a combination of mass, volumetric, gaseous, and heat balances of a bioreactor. In this investigation a concrete example of the population model published in work is used (Amelkin et al. 1995). Model 3 is a system of differential material, gaseous and heat balances equations. The structural and parametric identification of Model 3 was fulfilled by the authors of this work on the basis of the industrial and experimental data obtained during cultivation of various strains of baker's yeast (Amelkin 1991, Amelkin et al. 1995, Castrillo and Ugalde 1994, Gaponov 1984, Okada et al.

1981, Peringer et al. 1974, Shkidchenko et al. 1983, Sonnleitner and Käppli 1986, Wöhrer and Röhr 1981).

3 Optimal Control Problem Solution for the Cellular Level

The optimal control problem is reduced to maximization of energy evolution function of mitochondrion expressed in ATP synthesis by influence on respiratory chain of aeroion flow determined by the input voltage of the DAPTAP aeroion generator.

The optimal control problem treatment was fulfilled within a class of stepwise constant functions on a qualitative level with the use of OptiMod software (Amelkin 1992, Amelkin et al. 2000a) on the basis of Model 1. The results for the potatoes storage/ greensprouting case is depicted on **Figs. 2-5**.

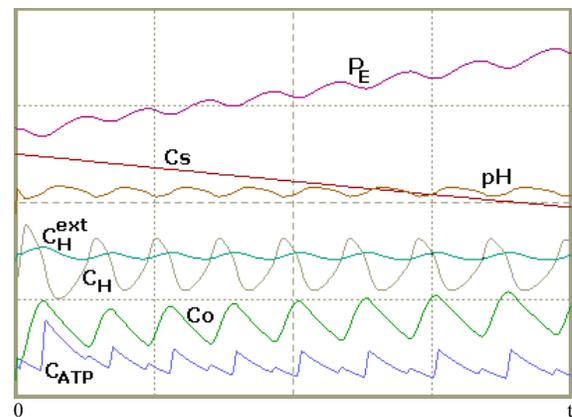


Fig.2: A case of no control.

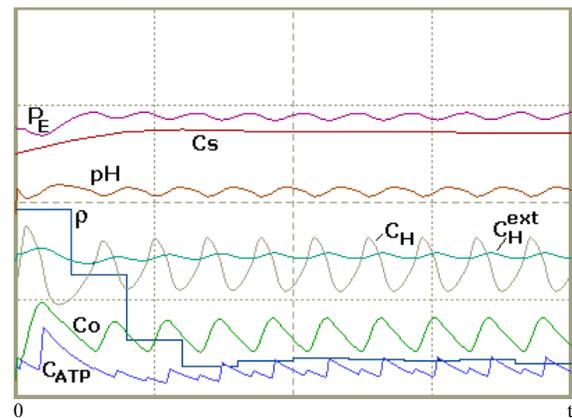


Fig.3: Potatoes greensprouting.

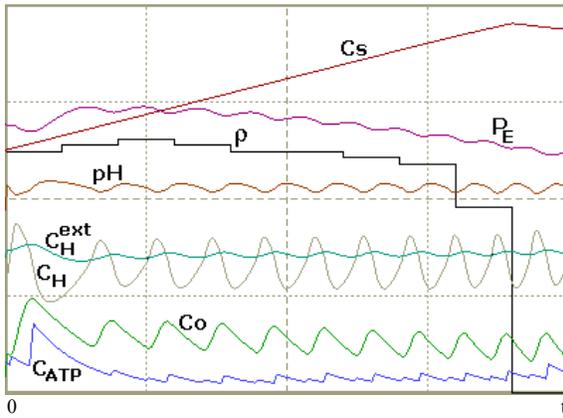


Fig.4: Potatoes storage control by ethylene parameter.

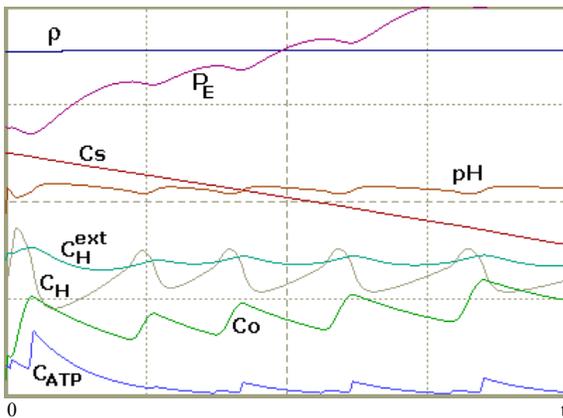


Fig.5: Overshoot case.

4 Microenvironment Optimal Control Problem Solution for the Yeast Cultivation Process

The lion's share of expenditures in baker's yeast production falls on the technological process of fermentation. That is why the fermentation process is usually given principal consideration in systems of automatic and automatized control of production as a whole. On the other hand, this process is the most complicated from the point of view of control.

Material costs make up the largest cost component in most production companies, at an average of up to 60% of total costs. In baker's yeast production the greatest portion of material costs is formed by molasses, salts and other feed component costs.

Taking into account the relatively high price of the main raw material (sugar beet molasses) and its high share in the process profit the microenvironment optimal control problem for baker's yeast industrial fed-batch cultivation in a bubbling bioreactor is reduced from a profit maximum to a maximization of the overall yield by the reason that the reciprocal value of overall yield correlates closely with the material flow-oriented specific cost.

The microenvironment optimal control problem for baker's yeast industrial fed-batch cultivation treatment was fulfilled with the use of OptiMod software as well

(Amelkin et al. 2000a) on the basis of Model 3 within a class of stepwise constant functions.

5 Development of Algorithm of Microenvironment Automatic Control

The analysis of optimal control problem solution results has shown that the optimal controls can be approximated for a significant length of time by exponential dependences.

The maximum of the biomass instantaneous yield corresponds to the maximum specific metabolic heat generation rate which allows one to use this parameter as a main parameter of feedback control during development of algorithms of automatic control. The rate of ethanol formation could be chosen as an additional feedback parameter which could be used in a control algorithm. The combination of these two parameters gives an opportunity of unambiguous recognition of the type and degree of technological process unfavorable variation.

At the present moment, the computer simulation of the elaborated algorithm of automatic control is being accomplished including measuring errors and parameters of drift simulating, and applying methods of exponential filtering of measured parameters and information validity monitoring.

6 Functional and Parametric Scheme of SACME

The functional and parametric scheme of subsystems for automatic control of the microenvironment (SACME) is developed. The SACME must function in close intercoupling with other MES control subsystems. In a potato storehouse, for example, SACME should interact with the subsystem of air conditioning (Brook 1999, Muzychenko 1991), in brewing - with air treatment and temperature control subsystems (Lobanov et al. 2000), and in yeast production - with control subsystems of cultural liquid temperature, pH, heat exchanger, etc. The SACME under design for baker's yeast cultivation includes sensors of state and perturbation parameters (temperature of liquid flows to and out of heat exchanger, cultural liquid temperature, flow of cooling water, temperature of feed flow, ethanol concentration), execution devices (the control valves in lines of aeration air and feed, aeroionizing device - DAPTAP), as well as control algorithm, realized by software means of a central computer or microprocessor controller. In the case of a potato storehouse, the periphery content is to be changed accordingly: for example, the concentration of ethylene but not ethanol is to be used as a measurable metabolic by-product of living objects. For the purpose of SACME simulation it is planned to use the complex mathematical model (Model 1 + Model 2 + Model 3) with a further demand of special experiments series setting to finish the structural and parametric identification of models.

7 Execution Device for Air Aeroionizing

The in-flow aeroionizing of aeration air in baker's yeast production could be realized at early stages of the process only during seed and intermediate culture production when air flow values are relatively small. Air aeroionizing is to be realized with the help of stationary or portable modifications of DAPTAP.

Stationary modifications of DAPTAP can be installed in the input air flow entering the bioreactor or malt-house, and portable modifications (**Fig.6**) are to be used for microenvironment control in boxes for seed culture growing as well as in storehouses, transport, hothouses, etc.

The essential feature of the constructed devices is that they ionize preliminarily ozonated and purified air during three-stage filtration, evolve almost no ozone into the environment and permit one to get air free of dust and gaseous chemicals, and of biological impurities, with an efficiency close to 100% and low power consumption. The modular construction of DAPTAP permits one to create stationary and portable devices of various modifications, capacities and configurations. The device is inexpensive and simple to operate. The microprocessor control unit will allow one to realize various optimal control modes depending upon aim of treatment and type of object. The working model of DAPTAP was tested at a potato-storehouse: the losses of potatoes were reduced by 30% (Amelkin et al. 2000a). Similar results were obtained during the fruits and vegetables storage process with the use of a device for aeroionization treatment of agricultural raw materials (Muzychenko 1991). The manner of air treatment and devices for its realization are protected by a valid patent (Amelkin et al. 1998).

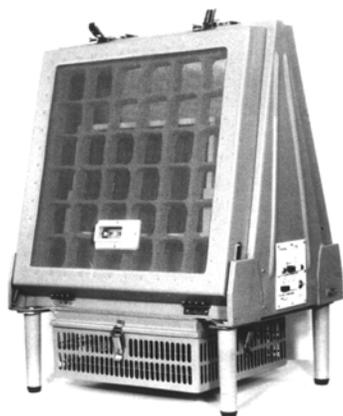


Fig.6: A pilot model of DAPTAP portable modification.

8 Sensors

In SACME there are various sensors are to be used for ethanol, ethylene, ozone, and biomass concentration detection by various methods: selective UV absorption, electrochemical fuel cell method, conductance-measuring method, etc.

9 The Domestic and Medical Application

An alternative application of the execution device for air purification and aeroionizing DAPTAP is its use in domestic conditions and for medical purposes. A group of pediatricians of Kazakhstan (Skuchalina et al. 2001) recently reported about a constant percentage growth of children sick with allergic bronchial asthma and atopic dermatitis throughout the world. The authors pointed to the responsibility of the microenvironment for the development of allergic diseases at child age. The main microecological factor is multicomponent composition of a domestic dust containing up to 900 ticks, 520000 fungus cells and 26770000 microflora cells per 1 gram of dust. Taking this into account, a controlled living microenvironment forming with DAPTAP use will be health-giving in regard to domestic and medical application.

10 Conclusions

1. The approach to SACME designing is developed: (a) the three-level mathematical model of microenvironment development; (b) the control problem's formulation and solution; (c) the optimal solution analysis and construction of algorithms of automatic control; (d) the functional and parametric scheme of SACME construction; and (e) choice of software/hardware means for SACME realization.

2. The use of DAPTAP device as an execution device for microenvironment aeroionizing is offered. The effect of DAPTAP on a living organism (human, mammal, gallinaceae, vegetable, fruit, cereal, plant, fungus, bacterial, vegetable, insect, etc.) can be indirectly monitored by measuring the different integral feedback parameters (the maximum specific metabolic heat generation rate, the rate of ethanol formation, etc.).

3. During further development of the present work it is proposed: (a) to set the series of special experiments; (b) to finish structural and parametric identification of complex mathematical model; (c) to accomplish SACME simulation with the use of the complex mathematical model; and (d) to choose the software/hardware means for SACME realization for various control objects.

4. The SACMEs under development can be applied within the MES framework in any areas where the living objects placed into a limited microenvironment are used.

5. The SACMEs serial production organization, including scientific laboratory establishment and the conductance of experimental investigations will need 500,000 USD of the total investments with 30 months repayment and 20% interest.

Nomenclature

Variables:

C_{ATP} - concentration of ATP, M;

C_S - concentration of active superoxide dismutase, M;

C_O - concentration of superoxide radicals, M;

C_H - protons concentration in matrix, M;

C_H^{ext} - protons concentration in intermembrane space, M;
 φ_1, φ_3 - electrochemical potentials of I and III complexes as hydrogen and oxygen electrodes of biochemical generator of electric current, respectively, V;
 $\Delta\varphi$ - potentials difference ($= \varphi_3 - \varphi_1$), V;
 P_E - by-product (ethanol for yeast case or ethylene for vegetables storage case) concentration in cell, M;
 P_i - inorganic phosphate concentration, M;
 pH - value of pH of matrix;
 pH^{ext} - value of pH of intermembrane space between the inner and outer membranes;
 $\rho(t)$ - income of superoxide radicals from the device to mitochondrial matrix (control function), M/h;
 t - time, h;

References

- Alberts B, Bray D, Lewis J, Raff M, Roberts K, Watson JD (1989): Molecular biology of the cell. Garland Publishing, New York - London
- Amelkin AA (1992): OPTIMOD Software: Mathematical Modelling, Computer Simulation, and Optimisation of a Biotechnological Process. - In: Food Engineering in a Computer Climate. A Three Day Symposium Organised by the Institution of Chemical Engineers' Food & Drink Subject Group on Behalf of the EFCE Food Working Party, Held at St. John's College, Cambridge, 30 March - 1 April 1992. Edited by: Wolf Hamm, Publisher: Hemisphere Publishing Corporation, Pages: 467-468
- Amelkin AA (1991): A mathematical model of baker's yeast cultivation process. *Izvestiya Vuzov. Food Technology (Russia)* 4-6: 93-97
- Amelkin AA, Amelkin AK (1997): Respiration modelling and control. - In: Proceedings of the Third IFAC Symposium 'Modelling and Control in Biomedical Systems (including Biological Systems)' (University of Warwick, 23-26 March 1997), The Institute of Measurement and Control, London, the UK, Session 8: 1-6
- Amelkin AA, Amelkin AK (1996): Mathematical modelling and control of aerobic organisms respiration. *Biotechnologiya (Russia)* 9 (Sep.): 45-50
- Amelkin AA, Blagoveschenskaya MM, Amelkin AK (2001a): The Microenvironmental Systems Project. - In: Proceedings of the 6th IFAC Symposium on Cost Oriented Automation (Low Cost Automation 2001 - LCA 2001) (Berlin, October 8-9, 2001). - Institut für berufliche Bildung, Zentrum Mensch-Maschine Systeme, Technische Universität Berlin, Berlin, Germany, 192-197
- Amelkin AA, Blagoveschenskaya MM, Amelkin AK (2001b): The subsystems of automatic control of living objects elaboration for food industry and agriculture illustrated with baker's yeast cultivation, lysine biosynthesis and agricultural produce storage, processing, and presowing activation examples. - Moscow State University of Applied Biotechnology, The 4th International Scientific and Technical Conference 'Food. Ecology. Man.' Materials. Education Ministry of Russia, MGUAB, Moscow, Russia, 240-250
- Amelkin AA, Blagoveschenskaya MM, Amelkin AK (2001c): The subsystems of automatic control of living objects elaboration for food industry and agriculture. - Moscow State University of Food Production 70th Anniversary Collected Papers Edition. Education Ministry of Russia, MGUPP, Moscow, Russia, 440-452
- Amelkin A, Blagoveschenskaya M, Amelkin AK, Amelkina A, Pletnyova A (2000a): Microenvironment control systems in agriculture, biotechnology, and food industry. - In: Proceedings of the 2nd International Euro Environment Conference on Industry and Environmental Performance, CD-ROM (Ålborg, 18 - 20 October 2000), Ålborg Congress and Culture Centre, Ålborg, Denmark
- Amelkin A, Blagoveschenskaya M, Amelkin AK, Amelkina A, Pletnyova A (2000b): Microenvironmental systems: modelling, control, applications. - In: Proceedings of the 14th Forum for Applied Biotechnology (Brugge, Belgium, 27 - 28 September 2000), Proceedings part I, GOM West-Vlaanderen, Med. Fac. Landbouww. Univ. Gent, 65/3a, Session 1 'Environmental Biotechnology', Posters: 163-166
- Amelkin AA, Blagoveschenskaya MM, Lobanov YuK, Amelkin AK (2003): Minimum Specific Cost Control of Technological Processes Realized in a Living Objects Containing Microenvironment. *ESPR - Environ. Sci. & Pollut. Res.* 10 (1): 44-48
- Amelkin AA, Kulikov AV, Pechkovsky AG (1995): The ways of automatic control of baker's yeast fermentation. - In: Proceedings of the IMACS/IFAC First International Symposium 'Mathematical modelling and simulation in agriculture and bio-industries' (M²SABI'95) (Brussels, 9-12 May 1995), Universite Libre de Bruxelles, Brussels, Belgium
- Amelkin AK, Smirnov V, Mikhajlov V (1998): A device for air sanitary treatment. - Patent of Russia No. 1623346, Patentee: Amelkin AA. Published in Bulletin of Inventions of Russia No. 15, Class F 24 F 3/16
- Brook RC (1999): Potato storage experiments: Two decades of progress in technology and management. *Michigan Potato Research Report* 31: 154-162
- But AI (1977): Application of electron-ion technique in food industry. Moscow, Food Industry: 88 pp.
- Castrillo JL, Ugalde UO (1994): A general model of yeast energy metabolism in aerobic chemostat culture. *Yeast* 10: 185-197
- Chizhevsky AL (1999): Air ions and life. Discussions with Tsiolkovsky. Moscow, Russia
- Gaponov KP, Chugasova VA, Pozdnyakova VM (1984): Oxygen in fermentation processes. Moscow, ONTITelmicrobioprom, Russia
- Lifshitz MN (1990): Aeroionification. Moscow, Russia
- Lobanov YuV, Garmash JuV, Terletsy MJu (2000): Experience of application of FIX SCADA package of Intellution Company at 'Ochakovo' Moscow Brewing & Bottling Integrated Plant. *Industrial Automated Control Systems and Controllers (Russia)* 9
- Muzychenko VA (1991) Electroaeroionization treatment of fruits and vegetables during storage. Abstract of thesis. Kiev, Ukraine
- Okada W, Fukuda H, Morikawa H (1981): Kinetic expressions of ethanol production rate and ethanol consumption rate in baker's yeast cultivation. *J. Ferment. Technol.* 59 (2): 103-109
- Peringer P, Blachere H, Corrieu G, Lane AG (1974): A generalized mathematical model for the growth kinetics of *Saccharomyces cerevisiae* with experimental determination of parameters. *Biotechnol. Bioeng.* 16 (4): 431-454
- Shkidchenko AN, Orlova VS, Termkhitarova NG, Rylkin SS (1983): The features of diauxic growth of *Saccharomyces cerevisiae*. *Microbiological Journal (Russia)* 45 (6): 30-35
- Skuchalina LN, Starosvetova EN, Sejtgaliev GM (2001): An Importance of Microenvironment for Development of Allergic Diseases at Child Age. - In: Proceedings of the 1st (5th) Congress of Pediatricians of Kazakhstan (Astana, 1 - 3 October 2001), Republican Society of Pediatricians of Kazakhstan, Scientific Center of Mother and Child Health Care, Kazakh State Medical University, Kazakhstan
- Skulachev VP (1994): Chemiosmotic concept of the membrane bioenergetics: what is already clear and what is still waiting for elucidation? *J. Bioen. Biomembr.* 26: 589-598
- Skulachev VP (1989): Biological membranes energetics. Moscow, Russia
- Sonnleitner B, Käppli O (1986): Growth of *Saccharomyces cerevisiae* is controlled by its limited respiratory capacity:

- formulation and verification of a hypothesis. *Biotechnol. Bioeng.* **28** (6): 927-937
- Temnov AV, Stavrovskaya IG, Sirota TV, Kondrashova MN (2000): Self-organization of associations of mitochondria and the effect of negative air ions. *Biophysics (Russia)* **45** (1): 83-88
- Volkenshtein MV (1988): *Biophysics*. Moscow, Russia
- Wöhler W, Röhr M (1981): Respiratory aspects of baker's yeast metabolism in aerobic fed-batch cultures. *Biotechnol. Bioeng.* **23** (3): 567-581
- Wolf J, Passarge J, Somsen OJG, Snoep JL, Heinrich R, and Westerhoff HV (2000): Transduction of Intracellular and Intercellular Dynamics in Yeast Glycolytic Oscillations. *Biophysical Journal*, **78** (3): 1145-1153

BIOMECHANICAL SIMULATION OF HUMAN LIFTING

COLOBERT B. MULTON F., CRETUAL A. and DELAMARCHE P.

Laboratoire de Physiologie et Biomécanique de l'Exercice Musculaire
Address : av. Charles Tillon, Université Rennes 2, UFRAPS, 35044 Rennes, France
Tèl : (+33) 02 99 14 17 78- Fax : (+33) 02 9914 17 74
Mail : briac.colobert@uhb.fr

Abstract: The aim of this paper is to simulate several levels of lifting strategies from parameters depending on the subject's centre of mass movements. Usually, symmetrical lifting strategies were categorized in two major solutions (Chaffin and Andersson, 1991): the squat lift that mainly involves a knee flexion and the back lift that mainly involves hip flexions. In the literature two main indexes (Zhang et al. 2000) were introduced to evaluate the natural selected strategy. These indexes either did not take all the articulations into account or did not consider the evolution of posture depending on time. We propose a new index based on kinematic simulation obtained through a blending of the two extreme strategies. This work was based on the motion blending technique introduced in computer animation (Witkin and Popovic, 1995). To parameterise this simulation method, five squat lifts, five back lifts and five freestyle lifts were performed by one subject. A para sagittal model with five body segments was used to describe the posture of the subject. We captured the angular trajectories of the different lifts to abstract the natural lift movement as a blending of the two extreme strategies. To this end, a blending coefficient, considered as the strategy index, was introduced to minimize the set of control parameters of such a model. Indeed, instead of specifying a blending coefficient to each joint separately, we introduced a unique blending coefficient based on the displacement of the centre of mass. This choice enabled us to use only a force plate system to generate the inputs of our model. Hence, the angular trajectories could be simulated only thanks to the displacement of the centre of mass and to the blending coefficients identified in this paper. Our results showed a constant pattern of the angular trajectories for each joint and each strategy. The resulting blending coefficient remained constant for each joint during the movement. However depending on the joint, different values of blending coefficients were computed. For the ankle, we found that back lift was very attractive (the same behaviour whatever the strategy used) whereas for the knee and the shoulder different behaviours were found. On the opposite, the hip and the elbow trajectories were not influenced by the strategy. This work has potential applications in computer animation and in clinical biomechanics. To conclude, this approach could be applied to all kinds of movements involving a compromise between two extreme strategies, such as lifting.

Keywords: biomedical simulation, human lifting, motion blending

1. INTRODUCTION

Lifting is a daily activity and its modelling is of interest for several areas. Several authors developed biomechanical models to evaluate and predict the effect of the weight of the load, position of load, body posture, or the strength capability of the human body on the way people lift weights. (Chaffin and Andersson, 1991; Hsiang et al. 1999).

Computer animation modelled human motion in order to generate realistic synthetic motions. These models were created to avoid motion capture and heavy manual motion editing, more expensive and difficult to reemploy. Most biomechanists (Hsiang et al. 1999, Chaffin and Andersson, 1991)

referred to the principle of two pure lift strategies. In case of back lift, leg remains in extension and only the hip joint, the spine and the upper limbs are used. Squat lift uses a flexion at the knee that decreases spinal constraints. Two authors presented indexes to quantify the strategy employed. Burgess-Limerick and Abernethy, (1997) proposed to quantify lifting strategy by the ratio between the knee flexion and the sum of ankle, hip and lumbar vertebral flexion. Unfortunately this index was only based on two postures: the standing posture and the one that occurred at the beginning of the lift, when the weight was held. Another index (Zhang et al., 2000) was based on leg and back velocity during

the lift. Nevertheless this model did not include the arms that contributed to the lift. Another problem was that the index was supposed time-invariant which is not true.

Kinematic simulation was widely used in computer animation (Multon et al. 1999). Especially, Frame Space Interpolation (Guo and Roberg, 1996) was introduced to blend four different angular trajectories by using interpolation and time-warping. Motion warping (Witkin and Popovic 1995) was also used to modify a reference motion in order to generate new behaviours. Nevertheless, these techniques have never been validated in comparison to real movements and, consequently, were not used in clinical applications.

2. MODEL

In our study, we used a 5-link para-sagittal lifting model (Chaffin and Anderson, 1991) currently used by ergonomists. Similar to Witkin and Popovic (1995), we blended two sequences of joint angles to create new joint trajectories. The blend was a straightforward weighted sum (considered as a time-dependant interpolation) of the two motion curves:

$$(1) \quad \theta_i = \alpha_i(t) * \theta_{M1} + (1 - \alpha_i(t)) * \theta_{M2}$$

where θ_i , θ_{M1} , and θ_{M2} were respectively, the interpolated motion, motion one (referred to as a back lift strategy) and motion two (referred to as a squat lift strategy) and $\alpha_i(t)$ was a normalized weight function depending on time.

It was possible to compute blending coefficients $\alpha_i(t)$ at each time of the trial. In the literature, the strategy evaluated by considering the initial posture may yield to large errors (Zhang et al., 2000). Hence, the strategy could be better identified in the middle-part of the movements while the initial and final posture may be identical. To avoid this problem, only the middle part of the trial was considered in our method to identify blending coefficients.

The coefficients $\alpha_i(t)$ were computed for each angular trajectory (ankle, knee, hip, shoulder, elbow) and for each time step:

$$(2) \quad \alpha_i(t) = \frac{\theta_i(t) - \theta_{M2}(t)}{\theta_{M1}(t) - \theta_{M2}(t)}$$

The same kind of calculation $\alpha_G(t)$ was carried-out for the centre of mass movements because it reflects the global posture of the subject:

$$(3) \quad \overrightarrow{OG^{M1}} = \alpha_G \left(\frac{\sum m_i \overrightarrow{OG^{M1}_i}}{\sum m_i} \right) + (1 - \alpha_G) \left(\frac{\sum m_i \overrightarrow{OG^{M2}_i}}{\sum m_i} \right)$$

where

$$\overrightarrow{\alpha_G} = \begin{cases} \alpha_{Gx} \\ \alpha_{Gy} \\ \alpha_{Gz} \end{cases} \text{ and } \alpha_{Gc} = \frac{OG_c^{M1} - OG_c^{M2}}{OG_c^{M1} - OG_c^{M2}},$$

, with c in $\{x, y, z\}$

Where \overrightarrow{OG} was the centre of mass position, $\overrightarrow{OG_i}$ was centre of mass position of the i^{th} segment. m_i was the mass of the i^{th} segment, α_G and α_i were obtained the same way, by computing the coefficient that linked the natural posture, the pure back-lift posture and the pure squat-lift one.

The two pure strategies and each natural lift motion engendered different centre of mass displacements that can be modelled according to equation 3. Our model was designed to be capable of simulating new lifting movements according to pre-recorded pure back lift and pure squat lift trajectories. The movements of the centre of mass depend on those of the body segments. The inverse kinetics problem that links the centre of mass position and those of the body segments engendered infinity of solutions because of the redundancy of the kinematic chain. As a first approximation we proposed to use a linear relationship between these two values, for each time step. To this end, we proposed to normalize the $\alpha_i(t)$ values by $\alpha_G(t)$:

$$(4) \quad \alpha_{iG}(t) = \frac{\alpha_i(t)}{\alpha_G(t)}$$

Using equation (3), it comes:

$$(5) \quad \overrightarrow{OG^{M1}} = \left(\frac{\sum m_i \frac{\alpha_i}{\alpha_{iG}} \overrightarrow{OG^{M1}_i}}{\sum m_i} \right) + \left(\frac{\sum m_i (1 - \frac{\alpha_i}{\alpha_{iG}}) \overrightarrow{OG^{M2}_i}}{\sum m_i} \right)$$

where each variable is time-dependent.

Given a α_G and a set of predefined $\{\alpha_{iG}\}$, at each time, it was then possible to design a new motion by applying equation 1. To conclude with this part, the overall system could be depicted as in figure 1

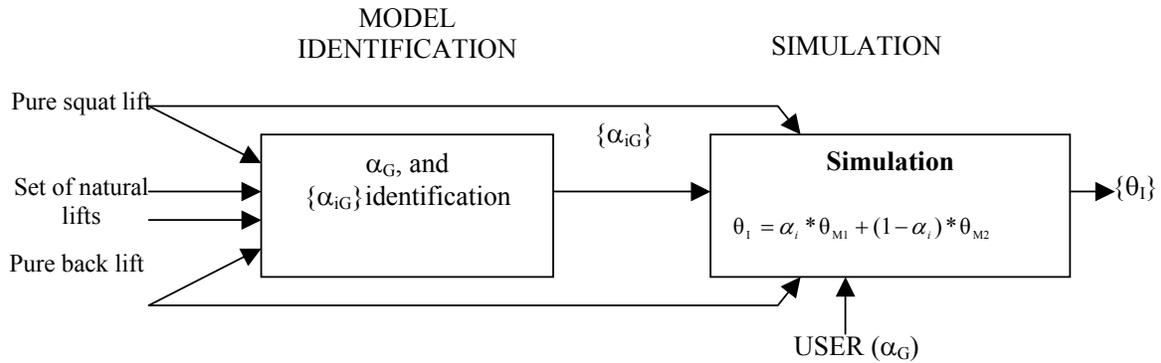


Figure 1: overview of the identification/simulation process.

3. PROTOCOL

One subject was instructed to perform back-lifts, squat-lifts and several free-lifts. Five trials of each style were performed. The subject was instructed to avoid a twist of the trunk during the lift. Every lift was started and finished from/to a stationary imposed posture. Markers were attached to the anatomical landmarks closed to joint centers and along the spine (figure 1). Joint displacements were collected with a motion capture system: VICON (370 Oxford Metrics) cadenced at 60 Hz. The joint trajectories were smoothed with cubic splines.

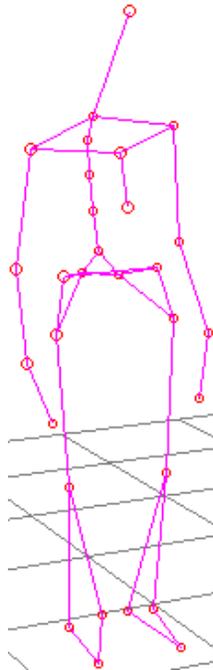


Figure 2 : position of the markers on the subject's body.

4.RESULTS

Given a lifting task with predetermined starting and ending positions, the five angular trajectories of an individual lifter are usually very consistent, with a distinctive pattern. This kind of observation was reported in the literature (Zhang et al., 2000; Hsiang et al, 1999). Standard deviation from the mean trajectory was around 3-5 degrees for all joints. This stability of the trajectory was observed for each evaluated strategy (see figure 3 for the squat-strategy). The smoothed and averaged angular trajectories for the squat-strategy and the back strategy trials are presented in figure 4. The resulting blending coefficients for all the trials are depicted in table 1.

Three kinds of behaviours were observed. Two articulations (knee and shoulder) behaved with a smooth transition between back strategy and lift strategy. Two articulations were not affected by the strategy and presented an identical shape in all cases (free, squat and back strategy). Finally an articulation (ankle) seems to be much more attracted by a strategy (squat lift) than the other (back lift).

Simulations were computed for six different values of α_G : 0, 0.25, 0.50, 0.75, and 1. Thanks to the imposed α_G , α_i for each joint were computed depending on the pre-recorded α_{iG} . The resulting movements are presented in figure 5 for α_G ranging from 0 (top of figure 5) to 1 (bottom of figure 5).

Table 1 : α_i , α_{iG} , α_G for the selected joints

Joint	α_i		α_{iG}
	mean	s.d.	Mean
Ankle	0.97	0.05	1.90
Knee	0.66	0.03	1.29
Hip	1.00	0.03	1.96
Shoulder	0.70	0.07	1.37
Elbow	0.0	0.2	0
Centre of mass (α_G)	0.51	0.1	

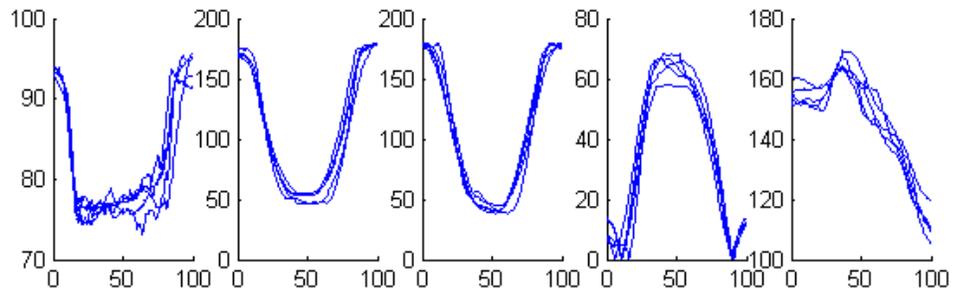


Figure 3 : angular trajectories (in degrees) of the five selected joints for the squat-strategy, ankle, knee, hip, shoulder and elbow, depending on time expressed as a percentage of the total duration of the movement.

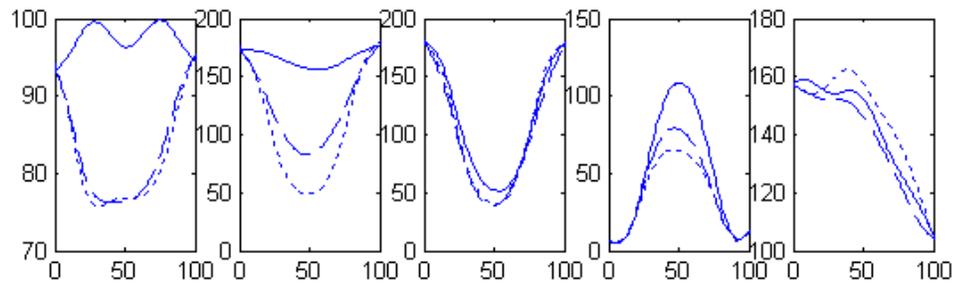


Figure 4 : angular trajectories (in degrees) of the five selected joints (ankle, knee, hip, shoulder and elbow) depending on time (% of the movement duration), for the three lifting condition; back lift in solid line, squat lift in dotted line and a free lift in dashed line.

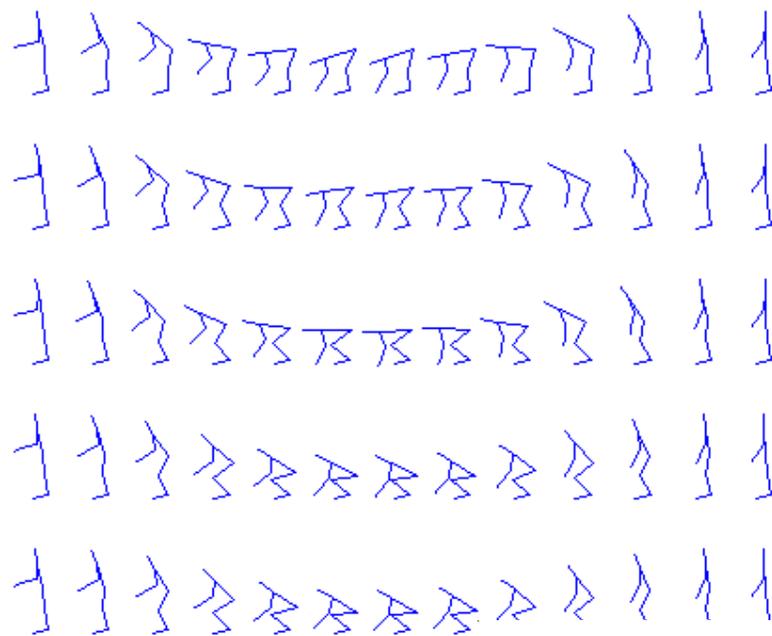


Figure 5 : simulated movements for α_G ranging from 0, to



5. DISCUSSION

This paper described a new approach to simulate a kinematical model of lifting movements thanks to a biomechanical analysis of lifting. As a result to this analysis, a new lift index was introduced. This input parameter of our model was the centre of mass blending coefficient between a squat and a back strategy. Such a blending coefficient is interesting as it relies to a physical meaning: the movements of the centre of mass. In addition to the index, a set of parameters were identified. These parameters deal with the link between the centre of mass position and the joints configuration. A table contained all these data to parameterize the simulation model. This method and its results can be applied in computer animation or in clinical biomechanics. For this last application field, using a costly motion capture system is not always possible. The alternative is generally to use force-plates that provides with other kinds of information (ground reaction forces, momentum and position of the centre of pressure). Consequently, it is necessary to define methods to use force-plates in order to indirectly access kinematic parameters. Force plate enables to evaluate the centre of mass displacements. Hence, our system enables to evaluate lifting strategy by only focusing on the centre of mass movements and using identified parameters.

To conclude, this approach could be applied to all kinds of movements involving a compromise between two extreme strategies, such as lifting.

REFERENCES

- Burgess-Limerick, R., Abernethy, B., 1997. Toward a quantitative definition of manual lifting postures. *Human Factors* 39, 141:148.
- Chaffin, D.B., Andersson, G.B.J., 1991. Occupational Biomechanical Models. In: *Occupational Biomechanics*. Wiley, New York
- Guo S. and Roberg J., 1996. A high-level mechanism for human locomotion based on parametric frame space interpolation. *Computer Animation and Simulation'96*, 95-107
- Hsiang S.M., Chang C., McGorry R.W. 1999. Development of a set of equations describing joint trajectories during para-sagittal lifting. *Journal of Biomechanics* 32, 871:876
- Witkin, A., Popovic, Z. 1995. Motion Warping SIGGRAPH 95, Los Angeles, August 6–11 *Computer Graphics Proceedings, Annual Conference Series*.
- Zhang, X., Nussbaum, M.A., Chaffin D.B. 2000. Back lift versus squat lift: an index and visualization of dynamic lifting strategies. *Journal of Biomechanics* 33, 777:782

KINEMATIC SIMULATION OF HANDBALL THROWING

FRADET L., KULPA R., BIDEAU B., MULTON F. and DELAMARCHE P.

Laboratoire de Physiologie et Biomécanique de l'Exercice Musculaire
Address : av. Charles Tillon, Université Rennes 2, UFRAPS, 35044 Rennes, France
Tel : (+33) 02 99 14 17 78- Fax : (+33) 02 9914 17 74
Mail : laetitia.fradet@uhb.fr

Abstract: Kinematic simulation of sport movements can be considered as an investigation tool for sport scientists. Nevertheless, kinematic simulation needs the specification of joint trajectories. Those trajectories can be modelled by control points and intermediate values can be computed with splines. So, a preliminary biomechanical analysis is required to model sport movements and especially to obtain the required control points. In addition to these control points, one has to define how the motion changes according to the situation, to consider these changes as inputs of the model. In handball throwing, one has to consider trajectories for all the joints and a set of operators that can adapt these control points to the situation: direction of the throw, ball speed, actions of opponents... The proposal of this study is to establish a model of handball throwing that could be adaptable to a maximum number of parameters, such as time of ball release, wrist position at ball release and throw type. The comparison of original joint trajectories obtained by motion capture with those obtained with such a model is encouraging. Moreover, the modification of an original movement produced trajectories that are closed to those obtained on real subjects placed in a similar situation. So, according to our results, this method looks promising to propose handball throwing simulations to sport scientists, even if only kinematics is considered.

Keywords: kinematic simulation, sport modelling, biomechanics, handball throwing

1. INTRODUCTION

Simulation is a good way to improve the technique of sport movements. Indeed computer simulation makes possible to validate (or not) investigations on human motion understanding. An hypothetical rule can be modelled in a computer module and tested in order to ensure that it produces coherent motions. Moreover, simulation gives the possibility to modify the movement in a larger way than real experiments do.

Several methods have been proposed in computer simulation in order to model human motion [Multon et al, 1999]. We can subdivide these methods in three main families. First, kinematic models consist in defining a mathematical expression to represent trajectories as a function of time [Zeltzer, 1982]. These models require to embed biomechanical knowledge on the studied motion, such as the phase duration [Bruderlin et al, 1996] or the trajectory of the ankle [Boulic et al, 1990] in human locomotion. Additional geometric constraints are added to ensure realistic adaptations to the skeleton of the subject and to the environment [Boulic et al, 1991].

Second, dynamics are used to ensure that the resulting motions verify the mechanical laws expressed in the Newton or Lagrangian formalism [Arnaldi et al, 1989]. The main problem of such a method is to design controllers to drive the motion equations. Several controllers are based on biomechanical knowledge on part of the motion. For example, maximum extensions and flexions angles of selected articulations (such as the knees and the hips) are used as objective functions to proportional derivative controllers [Hodgins, 1995]. Other controllers, such as constraint-based controllers [Multon et al, 1998] or those obtained through optimisation [van de Panne, 1994] are also tested. The main problems of these techniques still rely on the design of non-intuitive controller gains.

Third, motion capture and motion modification have been widely used by computing a new motion in the neighbourhood of the original one [Witkin and Popovic, 1995]. Additional constraints, such as spacetime constraints [Cohen, 1992; Gleicher and Litwinowicz, 1998] can also be added to make one part of the skeleton reach a target at a specific time. Another technique is to design coefficients with no

dimension to abstract motion parameters and, then, to simulate new motions by scaling these coefficients [Li, 2002].

Our goal is to design a model that reacts as a real handball thrower does in a similar situation. Simulating a complete human skeleton with dynamics is quite impossible for such complex movements because of the controller gains design. Moreover, modifying a captured or an average motion generally produces realistic behaviours only in the closed neighbourhood of the original motion. As a large number of motion strategies can occur in handball throwing, this method seems difficult to apply.

Hence, we propose to carry-out a biomechanical experiment to identify control points that seem fundamentals for every captured throws. As a second step, a kinematic model is designed to enable computer simulation of handball throws while respecting the fundamentals identified in the biomechanical experiment.

2. DEFINITION OF THE MODEL

2.1. Representation of the model

We choose to model all the Cartesian trajectories of selected articulations involved in a human model composed of 30 degrees of freedom.

The human body is composed with rigid bodies connected with joints (either pivots or ball-and-socket joints).

We choose to model the Cartesian position of selected points: the root placed at the middle of the pelvis which trajectory is described according to a fixed Cartesian reference frame. The sternum and the two shoulders are designed relatively to the root, both two elbows and two wrists relatively to their respective shoulder, both two hips relatively to the root, both two knees and two ankles relatively to their respective hip and, finally, both two toes relatively to their respective ankle.

These trajectories expressed in the Cartesian reference frame instead of in the joint angular representation enable us to control parts of the skeleton. Indeed, motion parameters are generally specified in the Cartesian reference frame: position of the wrist at ball release, initial velocity vector of the ball, direction of the throw, height of the elbow... Modifying these trajectories is then more intuitive than tuning angles to release the ball at the required position and speed.

Moreover, we describe the motion of a member extremity (such as the wrist) relatively to its proximal origin (such as the shoulder) to make motion modification easier and more intuitive. For

example, the modification of the wrist position at release is easier to modify relatively to its respective shoulder than relatively to the root, especially when intermediate articulations (such as the trunk flexion) also change. Each trajectory is also normalized according to the member or the kinematic sub-chain it belongs to. For example, the trajectory of the elbow in the shoulder reference frame is normalized according to the arm length. As a consequence, it enables to scale the motion to a new skeleton (with a different size).

2.2 Specificity of the elbows, knees and sternum

The aim of this model is to be adaptable to a large set of parameters. The modification of the wrist trajectory must induce a modification of the elbow trajectory (idem for the foot). For this reason, the elbows, the knees and the sternum trajectories are obtained by using analytical inverse kinematics with a constraint to be as closer as possible to the trajectory given by the model.

2.3. Mathematical modelling of the trajectories

We use cubic splines to approximate each trajectory [Watt and Watt, 1992]. The cubic splines are designed to fit the captured trajectories with an imposed maximum error. Hence, control points are added until the error between the resulting and the captured trajectory gone under this imposed threshold.

So to construct the splines, we need to specify the corresponding control points. In that case, the control points are represented by three parameters: the time, the joint coordinate on the concerned axe and finally, the derivative of this coordinate which gives the tangent of the curve at this specific time. To know these control points, we need to perform a dedicated biomechanical analysis of handball throwing that is done thanks to motion capture.

3. ANALYSIS OF HANDBALL THROWING

3.1. Experiment

Twelve male handball players took part of this study. These subjects play in the French Second League. The players completed informed consent, physical information and history on their handball practice.

Each subject, following warming up, threw at maximum velocity into a handball goal. They performed:

- 4 throws with the two feet on the ground with the last foot strike on the right foot,
- 4 throws with the two feet on the ground with the last foot strike on the left foot,

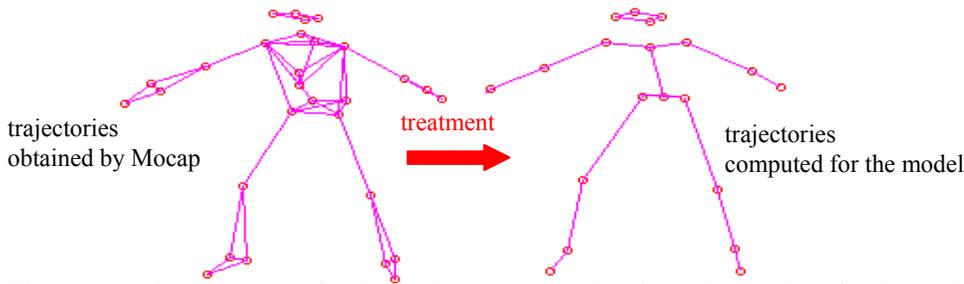


Figure 1: marker placement for the motion capture and trajectories required for the model.

- 4 jump in throws with the last foot strike on the right foot,
- 4 jump in throws with the last foot strike on the left foot.

For each throw, three-dimensional full body kinematic data are obtained at 60 Hz using an automatic opto-electronic motion capture system (Vicon, Oxford Metrics, England). Seven cameras are placed in a 9 m-radius circle with the centre being the throwing zone. Reflective, 20 mm diameter spherical markers are attached to each body segment as depicted in the left part of figure 1.

When occlusions occurred, the missing markers are calculated using the method developed by Ménardais et al [Ménardais et al, 2002]. The joint centres and the required trajectories are then computed using a method similar to that developed by Oxford Metrics in the Vicon software [Vicon, 2003].

The 60 Hz kinematic data are independently filtered using a Butterworth second order low-pass filter with a 10 Hz cut-off frequency.

3.2. Categories of movements for the two joint groups: “upper body” and “lower body”

The analysis described above allows us to specify the movement of each joint. We have detected that the joints can be subdivided into two groups: the “upper body” and the “lower body”. The “upper body” is made up of the sternum, the elbow of the throwing arm, both shoulders and the wrist of the throwing arm. The “lower body” is composed of the remainder of the joints.

With the knowledge of handball game, for each joint group, it is also possible to distinguish different main categories of movements. For the “lower body”, we consider if a jump occurs or not. For a jump, we also distinguish the motions for which the left or the right foot is used to jump. On the other hand, when the two feet are in contact with the ground, we distinguish if either the right or the left foot is in front of the body. Hence, four different main categories are identified.

For the “upper body”, four different main throws are also identified. The criterion used to differentiate the categories is the position of the wrist relatively to the shoulder at ball release. These

throws are the “external throw”, the “internal throw”, the “middle throw” and the “middle and high throw” as depicted in figure 2.

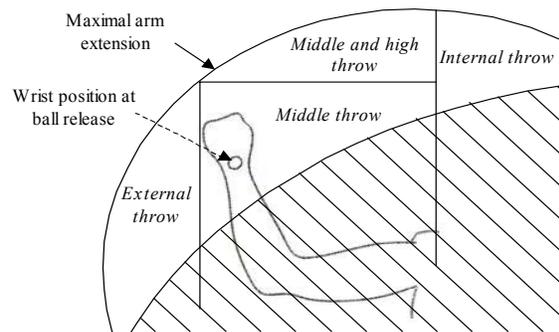


Figure 2: definition of the category of throws for the “upper body”

3.3. Time-decomposition of the throws

The whole throwing motion is subdivided into successive phases according to time events. For the “lower body”, two phases are considered:

- the “previous phase” begins at the last foot strike but one and finishes at the beginning of the last step.
- the “last phase” represents the last step or the aerial phase in case of throw with jump. This phase finishes with the foot strike.

For the “upper body”, three phases are considered:

- the “arm cocking phase” starts with the increasing of distance between each wrist and ends when the shoulder begins its forward movement.
- the “phase of the throw” ends at ball release.
- the “phase of deceleration” corresponds to the end of the motion.

Each phase is normalized by its duration in order to allow future adjustments imposed by a user, as an input of the simulation system. These phases are defined to have lots of possible movements. It is then possible to adjust a special movement only during one phase as the “arm cocking phase” without modify the other phases. In addition to the previous advantages, subdividing the movement into successive phases enables us to

easily consider all the possible modifications locally to each phase.

4. MODELLING OF HANDBALL THROWING

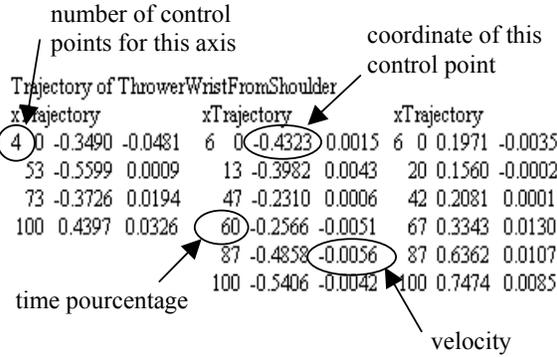


Figure 3 : Example of control points calculated for the wrist during the “throw phase” of the “middle and high” throw.

For each phase, each group of joint and each category of movements, we have made biomechanical analysis on the movements performed by our subjects. We have computed control points for these throws for all the trajectories considered for our model.

Wrist	Middle	Middle and high	External	Internal
X axis	4	4	4	4
Y axis	6	6	6	6
Z axis	6	6	6	6

Table 1: Number of control points for the wrist trajectory relatively to the shoulder during the “throw phase” for different categories of throws.

As in the biomechanical literature [Feltner and Dapena, 1986], each joint trajectory follows a similar shape, even if each thrower has a

X axis	Y axis	Z axis
$X_{1f} = X_{1i}$	$Y_{1f} = Y_{1i}$	$Z_{1f} = Z_{1i}$
$X_{2f} = X_{2i} + 1/3 * \Delta X$	$Y_{2f} = Y_{2i} + 1/4 * \Delta Y$	$Z_{2f} = Z_{2i} + 1/4 * \Delta Z$
$X_{3f} = X_{3i} + 2/3 * \Delta X$	$Y_{3f} = Y_{3i} + 1/2 * \Delta Y$	$Z_{3f} = Z_{3i} + 1/2 * \Delta Z$
$X_{4f} = X_{4i} + \Delta X$	$Y_{4f} = Y_{4i} + 3/4 * \Delta Y$	$Z_{4f} = Z_{4i} + 3/4 * \Delta Z$
	$Y_{5f} = Y_{5i} + \Delta Y$	$Z_{5f} = Z_{5i} + \Delta Z$
	$Y_{6f} = Y_{6i} + \Delta Y$	$Z_{6f} = Z_{6i} + \Delta Z$

Table 2: Modification of the control points linked to the wrist trajectory where (X_{ji}, Y_{ji}, Z_{ji}) are the jth control point of the initial average trajectory, (X_{jf}, Y_{jf}, Z_{jf}) are the jth control point at ball release and (ΔX, ΔY, ΔZ) are the vector coordinates that linked the desired wrist position at ball release and the original one.

personified movement. So it is yet possible to define common control points between the throwers. Consequently, we define an average trajectory that is a compromise of all the players’ styles. This average motion is also represented by the control points that are identified in all the measurements. Figure 3 gives the wrist control points for the throw phase of the “middle and high” throw. The number of control points required to specify the wrist trajectory during the “throw phase” are noted in table 1.

5. MOTION MODIFICATION

5.1. Trajectory modification

According to all the measurements, for each trajectory, a set of operators is identified. These operators are designed to enable the user to change high-level parameters, such as the position of the wrist at ball release and to calculate the corresponding modifications to apply to the control points.

Let us consider now the example of a change in the wrist position at ball release. For each category of throw, we have studied how the wrist trajectory varies according to the wrist position at ball release. Table 2 gives the changes of all the control points of the wrist trajectory depending on the final wrist position at ball release.

To this end, we modify the control points with the same method described for the wrist. So we analyse how the control points of this angular trajectory change according to the final orientation of the trunk.

As our model is based on the Cartesian trajectory of each point relatively to a father articulation, a two-steps process is proposed. First, all the Cartesian adaptations are performed to compute the new motion without taking the lateral flexion of the trunk into account. Next, the lateral flexion is applied to the trunk.

5.2. Time modification

As specified above, each phase of the throw is considered separately with its own duration. This duration is normalised by the total duration of the whole motion. However, as the “upper body” and

the “lower body” are dissociated, the specified duration for each throw is supposed to respect the synchronisation of the two parts. The analysis of the throw gives the mean duration of each phase and the synchronisation between them. Nevertheless, it is possible for the user to change these parameters.

The initial positions of a phase are set by the modified final positions. So, we are sure to have a continuous movement even if we modify a parameter during a phase.

6. RESULTS-DISCUSSION

6.1. Validity of the model

The model is embedded in a visualisation platform in which a user can specify high-level parameters through an interface dedicated to the handball application. This application enables us to visualise resulting motions given by the model.

This model needs to be validated in order to be used by sport scientists and coaches to test and improve knowledge on handball throwing. First, we compare the trajectories obtained by motion capture with the trajectories calculated by our model in similar situations. Figure 4 depicts that the modelled trajectories are very close to those obtained with motion capture.

The mathematical functions obtained by the movement analysis give positive results. The trajectories modified thanks to these mathematical functions on the control points are closed to these measured by motion capture (see figure 4).

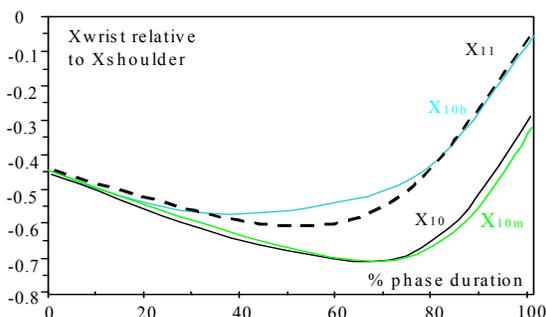


Figure 4 : Wrist X coordinate for trial 11 (X_{11}), trial 10 (X_{10}) obtained by motion capture, modelled trajectory of trial 10 (X_{10m}) and deformed trajectory of trial 10 (X_{10b}).

6.2. Perspectives

The modification of a trajectory with a kinematic method can give unrealistic results even if the use of inverse kinematics decreases this kind of possible errors. Indeed, the joint limits are not taken into account. The specification of forbidden areas for each joint can restrain these errors.

A limit of our model is due to the absence of the hand in our model. We know that the hand

movement cannot be neglected but the understanding of the hand movements requires a specific study that has not been made yet. Taking the ball trajectory at release into account would be another interesting extent of our model.

To conclude, this model could be used in a lot of applications including computer animation and sport science. For instance, this model is used in a virtual reality application which aim is to identify the parameters considered by the goalkeeper to react. Thanks to this model we are able to modify one parameter and to evaluate its influence on the goalkeeper's reaction [Bideau et al, 2003]. Moreover, it can validate our model if the simulated movements engender realistic goalkeeper's reactions.

Our model is also used in the design of a new motion capture system. In this system, a model-based interpolation is used to retrieve missing or hidden markers.

This kind of association between simulation and analysis seems to be a promising tool to improve knowledge on human movement that is generally complex to analyse in real situations.

ACKNOWLEDGEMENTS

This work has been supported by the French Ministry of Sport, Youth, the French Federation of Handball, the French Olympic Preparation committee and the “Conseil Régional de Bretagne”. This work is a part of RNTL “Mouvement” project, supported by the French Ministry of Industry.

REFERENCES

- Arnaldi B., Dumont G., Hégron G., Magnenat-Thalmann N., Thalmann D. 1989, “Animation control with dynamics”. In Proc. of Computer Animation'89. Pp113-123.
- Bideau B., Kulpa R., Ménardais S., Multon F., Delamarche P., Arnaldi B. 2003, “Real handball goalkeeper vs. virtual handball thrower”. Presence.
- Boulic R., Magnenat-Thalmann N., Thalmann D. 1990, “A Global human walking model with real-time kinematic personification”. The Visual Computer. 6(6). Pp344-358.
- Boulic R., Thalmann D. 1992, “Combined Direct and Inverse Kinematics Control for Articulated Figures Motion Editing”, Computer Graphics Forum. 11(4), Pp.189-202.
- Bruderlin A., Calvert T. 1996, “Knowledge-driven, interactive animation of human running”. In Proc. of Graphics Interface'96. Pp213-221.
- Cohen M. 1992, “Interactive spacetime control for animation”, in Proc. of ACM SIGGRAPH'92, 26, Pp293-302
- Feltner M., Dapena J. 1986, “Dynamics of the shoulder and elbows joints of the throwing arm

during a baseball pitch". International journal of sport biomechanics. 2, Pp235-259.

Gleicher M. and Litwinowicz P. 1998, "Constraint-Based Motion Adaptation". The Journal of Visualization and Computer Animation. 9. Pp65-94.

Hodgins J., Wooten W., Brogan D., O'Brien J. 1995, "Animating human athletics". In Proc. of ACM SIGGRAPH 1995. Pp71-78.

Li Y., Wang T., Shum H.Y. 2002, "Motion Textures: A Two-Level Statistical Model". In Proc. of ACM SIGGRAPH 2002.

Ménardais S., Arnaldi B. 2002, "A Global Framework for Motion Capture". Research report INRIA, N° 4360.

Multon F. 1998, "Biomedical Simulation of Human Arm Motion". In Proc. of European Simulation Multiconference, Manchester, Pp305-309.

Multon F., France L., Cani-Gascuel MP., Debunne G. 1999, "Computer Animation of Human Walking: a Survey". Journal of Visualization and Computer Animation. 10, Pp 39-54.

Van de Panne M., Kim R., Fiume E. 1994, "Virtual wind-up toys for animation". In Graphics Interface, Banff, Alberta, Canada. Pp208-215

Vicon. 2003, "PIGModeling". Technical reports, available from www.vicon.uk.

Watt A., Watt M. 1992, "Advanced and rendering techniques : theory and practice." ACM Press.

Witkin A., Popovic Z. 1995, " Motion warping". In Proc of ACM SIGGRAPH. Pp105-108.

Zeltzer D. 1982, "Motor control techniques for figure animation". IEEE Computer Graphics and Applications. 2(9), Pp53-59.

STOCHASTIC AND STRAIN-WEIGHTED SIMULATIONS OF CANCELLOUS BONE REMODELLING: SIMULATION RULES AND PARAMETERS

G SISIAS¹, CA DOBSON², R PHILLIPS³, MJ FAGAN² and CM LANGTON⁴

¹ *Department of Computing, School of Informatics, University of Bradford, Bradford, UK*

² *School of Engineering, University of Hull, Hull, UK*

³ *Department of Computer Science, University of Hull, Hull, UK*

⁴ *Centre for Metabolic Bone Diseases, University of Hull and Hull and East Yorkshire Hospitals NHS Trust, UK*

Abstract: Bone has well-defined structural and morphological properties, as well as cellular processes based on stimuli that control activity at microscopic level. Simulations that can take into account the above and can operate on real bone can be used to investigate scenarios such as normal age-related loss of bone, loss of bone due to disuse or osteoporosis or virtual drug treatment. The aim of this work is to define a set of simulation rules such that the cellular processes of bone can be modelled and used in scenarios that investigate bone remodelling and the effects on its mechanical properties.

Keywords: Simulation, cancellous bone, remodelling, stochastic, strain-weighted.

1. INTRODUCTION

Bone tissue consists of calcium hydroxyapatite mineral absorbed onto a collagen matrix. Throughout life there is a process of remodelling, where old bone is removed by osteoclast cells and new collagen fibres are laid down by osteoblast cells. This is under the control of physical activity and several hormones. There are two types of bone structure, cortical and cancellous. Cortical bone is predominantly solid and makes up the shafts of the long bones in the skeleton. Cancellous bone has a porous structure made up of an array of trabecular bone fibres interspersed with bone marrow, and is found near the joint surfaces of the long bones and within the individual vertebrae making up the spinal column. Bone grows under body forces and the trabecular fibres follow the principal lines of stress, as can be clearly seen, for example, in a cross-section of the hip.

2. BONE PATHOPHYSIOLOGY

The main responsibilities of bones are to withstand the mechanical forces exerted to them by muscles or gravity, protect the vital organs from possible damage, and provide a reserve of minerals such as calcium or phosphate to the body.

Most bones are composed of a dense outer shell of cortical bone surrounding the central porous cancellous (trabecular) bone. Cancellous bone can be considered to be a cellular structure, consisting of

an interconnecting 3D network of thin bars (trabeculae) interspersed with marrow, connective tissue and blood vessels (Baron 1993), with the porosity of cancellous bone typically ranging from 30% up to 95% (Gibson and Ashby 1988).

From birth to maturity, healthy humans exhibit an increase in bone mass of about 40 times. The peak of bone mass as result of growth is reached between the ages 20-30, and bone mass starts decreasing around ages 30-40, while by the age of 70 more than 30% of the original peak bone mass is lost. This long-term bone loss is temporarily accelerated on women after menopause, but after a few years this acceleration stops. This process of old bone replaced by new one is also known as bone remodeling. The phases of the remodeling process are resorption, reversal, formation and quiescence, and last totally about 180-200 days.

The main two types of cells responsible for the remodeling process are osteoblasts and osteoclasts. Osteoblastic cells are responsible for the production of the material for reposition (e.g. collagen) and are not normally found alone, but in groups of about 100-400, forming a bone reposition site. On the other hand osteoclasts are responsible for the resorption of bone, and are usually found in groups of 1-2 or even 4-5 cells. Such a group of similar type of cells is also known as a BMU, or a basic multi-cellular unit.

In bone remodeling, groups of osteoclasts firstly remove bone and after a reversal period, osteoblasts attempt to replace the same amount of bone that was resorbed. If less bone was repositied there is a negative BMU balance, and if more bone is repositied there is a positive BMU balance. If the same amount of bone lost is replaced, homeostasis is maintained, while the whole removal-reposition process occurs at the interface of bone with marrow. The acceleration of bone loss during menopause is accredited to osteoclasts digging larger holes at the surface of bone for longer time.

3. BACKGROUND

Siffert *et al* (1996) developed a computational model at the tissue level to study the effects of short- and long-term periods of disuse osteopenia and repair to elucidate the interrelationships between bone mass, architecture, and strength. The model is based on the principle that osteoclastic thinning of trabeculae and osteoblastic thickening is a surface occurring phenomenon. The structure under consideration is a highly idealised rectangular mesh where half the horizontal trabeculae are thinner than the vertical ones. In the model it is assumed that the stimulus S for adaptation to the mechanical loading is the local mechanical strain rate, according to which the trabecular surfaces are differentially formed and resorbed. Typical levels of tissue modulus have been determined to be around 5-8 GPa, and the study considers 7 GPa, while the loading history of the bone was assumed to be that of normal walking, at a frequency of 1-2 Hz. The components of the model were:

- boundary element method (computationally evaluates the local stresses and strains at each point in trabecular surface, assuming that the trabecular bone material is linearly elastic and isotropic);
- the local adaptation criterion;
- initial trabecular structure;
- the applied loading regime (e.g. angular).

The criterion that deciding if a trabecula's width will increase, decrease or remain unchanged is given by the following step function:

$$U = \begin{cases} K_1(S - S_{o-}), S < S_{o-} \\ 0, S_{o-} < S < S_{o+} \\ K_2(S - S_{o+}), S_{o+} < S \end{cases} \quad (1)$$

The *short-term* loading regime was simulated by a short period of disuse and the reapplication of loading before any horizontal trabeculae were lost. Recovery was simulated by reapplying the loads until a new adaptation equilibrium was achieved.

The *long-term* loading regime was simulated after a long enough period of disuse such that some of the horizontal trabeculae were lost, and the reapplication of loading until new adaptation equilibrium was achieved.

It was observed that there was a significant change in the volume ratio of -12% for short-term and -20% for long-term disuse. For the horizontal direction, the Young's modulus decreased 36% and 62% for short and long term disuse, respectively. The shear modulus reduced by 57% and 85% , and the Young's modulus change in the vertical direction was negligible, around 2% and 3% respectively.

The results of the simulation suggest that when the local strain rates are below resorption limits, trabeculae are thinned and the disuse process stops before trabeculae are lost, the structure remodels to a new equilibrium stage. The longer the disuse period (>12 weeks for short-term and >20 weeks for long-term) the more trabeculae are lost, and when the loading is reapplied, vertical trabeculae are selectively thickened than horizontal. Additionally, bone growth is observed at pre-existing trabeculae, and as such the behaviour of the model is in agreement with clinical studies which show that the effects of long-term disuse cannot be fully reversed. Consequently, there is a disproportional decrease in the mechanical competence of long-term disused bone compared to short-term disused bone.

The model developed by Lacy *et al* (1994) extended Reeve's model (Reeve 1986), simulating changes in trabecular thickness and hence in bone volume, considering resorption and formation over a large number of remodeling sites.

The model depends on a large number of hypothetical trabeculae subject to bone loss or formation. A BMU (basic multi-cellular unit) may initiate bone turnover at either of the two opposing sites of a trabecula. During the simulation the thickness can decrease, remain unchanged or increase, according to what point it is at in the simulation, until the trabecula perforates (thickness becomes 0) or the simulation finishes.

During the simulation there is a random chance for a trabecula to be initiated, based on the activation frequency. Each activated BMU remodels bone according to a series of steps, the parameters of which are drawn randomly from statistical distributions in accordance with clinical data.

The parameters controlling the model are taken or derived from static and dynamic histomorphometric studies and are:

- activation frequency;
- final resorption depth;

- resorption period;
- formation period;
- BMU balance (formation / resorption);
- initial trabecular thickness;

The trabecular thickness is drawn from a random number distribution, and is in agreement with Recker *et al* (1988), at 134 ± 34.4 (SD) μm for normal females from 55 to 64. The authors validate their model using clinical data from two histomorphometric studies (using Dual-energy Photon Absorptiometry) lasting 60 and 120 weeks predicting well the results of the biopsies with their model on both placebo and etidronate group.

The simulation involved 10 runs of 1000 trabeculae each (hence 2000 remodeling sites). The two most important simulation parameters are the activation frequency (increases likelihood of coincident activation) and resorption depth (increases likelihood of penetration). The main remodeling phases considered were bone resorption and formation. The reversal step between these two phases was not considered and justified by the unavailability of experimental data for representing reversal time.

The model developed by Thomsen *et al* (1994) was based on Reeve's model, mainly considering horizontal trabeculae, while attempting to be a treatment extension to Lacy's model. The model extends the treatment regimes, including a second anti-resorptive agent, estrogen, along with an anabolic agent, fluoride. The thickness of each trabecula is taken from histomorphometric studies and set to approximately 135 ± 24 μm , and totalling to about 600-630. The model is based on variation of the activation frequency and resorption depth, and describes the variation in the bone mass, the average trabecular thickness and the number of perforations over an extended period of time. The remodelling phases are in agreement with Frost and Eriksen, and are resorption, reversal, formation and quiescence.

The parameters that control the model are

- resorption period (σ_r);
- reversal period (σ_o);
- formation period (σ_f);
- initial trabecular thickness (w);
- resorption depth (d);
- critical trabecular thickness (w_c);
- "static" formation balance $\Delta\text{B.BMU}$ (b);
- activation frequency (μ);
- number of trabeculae – the only parameter fixed to 628 (N)

Given μ to be the activation frequency, the probability of starting remodelling on a trabecula under quiescence is given by uniformly distributed η :

$$\eta = \frac{1}{\frac{1}{\mu} - (\sigma_r + \sigma_o + \sigma_f)} \quad (2)$$

Consequently, the number of trabeculae undergoing remodelling are given by

$$N_r = N \cdot \eta \cdot (\sigma_r + \sigma_o + \sigma_f) \quad (3)$$

and under equilibrium, it should hold that:

$$N \cdot \eta = \frac{N_r}{\sigma_r + \sigma_o + \sigma_f} \quad (4)$$

The model simulates the remodelling process (mostly loss of bone) of horizontal trabecular struts in human vertebral body. Linearity in time is assumed during the resorption and formation phases, and the thickness of each trabecula is drawn from a Gaussian random number generator (RNG) with a mean and variance being in accordance to clinical results. Lastly, an even distribution is assumed when selecting trabeculae for remodelling.

Events such as menopause are represented by substituting certain parameters with newer ones and replacing them with original values after the end of the event. Since such events do not occur from one day to another, the whole event is modelled as smaller scale events over the transition period.

The remodelling process is initiated by assigning the resorption depth for trabeculae from a Gaussian RNG, and the time is also selected by a similar RNG. The possibility of disconnected trabeculae under remodelling increases when the bone becomes thinner than the critical trabecular thickness. The later is in agreement with clinical studies which show that trabeculae with thickness under a certain level do not exist.

Kinney and Ladd (1998) developed a finite element based model to examine the relationship between connectivity density and elastic modulus of trabecular bone, using cubic specimens prepared from human distal radii and L1 vertebrae using synchrotron microtomography. The 3D images were reconstructed into binary volumes of mineralised bone and soft tissue. Despite the instrument's maximum spatial resolution, the data were reconstructed into cubic elements of average edge 20 μm , while the true dimensions were 17.7 and 23.4 μm for the radii and the vertebrae, respectively. The final data sets used were cubic structures of about 3.5 and 4.5 mm having 7.7 and 7.1 million elements, respectively.

Connectivity scaling was explored by thinning and thickening the trabecular bone in each volume set by removing or adding one element each time (≈ 20 μm).

Two methods were utilised to simulate bone atrophy or recovery. The first method involved the identification and subsequent thinning (atrophy) or thickening (recovery) of all surface elements, without regard to the connectivity of the trabecular network. With this method trabecular connections could be formed or destroyed, and plates could be fenestrated or filled. In the second method all the surface elements were removed or added subject to the condition that the connectivity of trabecular structure did not change. With the second method bone mass could be removed or reposit without destroying or forming trabecular connections or fenestrating plates. Each specimen was thinned three times with each method, and additional data was prepared by thickening each specimen for five times from its fully atrophied condition. Finally, the original unthinned volumes were thickened twice to establish connectivity and modulus values that could be used in interpolating the results.

The authors note that a close inspection on the data sets prepared by the non connectivity-preserving algorithm shown several small plate perforations after a single thinning operation, clearly indicating that several plates were <40 μm thick. Without connectivity preservation, the atrophy model led to an increase in the connectivity with decreasing trabecular density. It was also observed that upon recovery plate fenestrations were removed while severely resorbed rods were not reconnected.

Both thinning methods resulted in a decrease in elastic modulus with trabecular bone density. In samples where lost connectivity was not restored, the original modulus for the equivalent trabecular bone density was also not restored. The results have also shown that due to the fact that connectivity is not dependent of the contact area, whereas mechanical load transfer is, there is no functional relationship between connectivity and elastic modulus. More importantly, a global measure of connectivity does not discriminate between trabeculae-like connections and fenestrated plates.

The authors conclude that irreversible connectivity reduction is one of the earliest manifestations of estrogen loss, and that early intervention to prevent possibly irreversible deterioration of the trabecular architecture after menopause is advised.

Silva and Gibson (1997) developed a two-dimensional model of human vertebral trabecular bone and investigated its mechanical behaviour using finite element analysis. Random reductions in the number and thickness of trabeculae were simulated.

The two-dimensional finite element model was generated using a technique based on Voronoi diagrams. A two-dimensional array of 20×20 points

spaced 1×1 mm apart was generated. The coordinates of the points in the square array were perturbed in each direction by a random amount of the range -0.3 to 0.3 mm, based on a uniform distribution random number generator. To create a model with the appropriate bone volume and degree of anisotropy, the Voronoi diagram (trimmed to approximately 17×17 cells) was scaled by 2.33 times in the traverse direction and 3.5 times in the longitudinal direction. Each Voronoi diagram was then converted into a finite element mesh for its elastic and ultimate mechanical properties.

Three "intact" finite element meshes were first generated and analysed. Each mesh was generated using a unique list of random numbers to perturb the nucleation points. Values of 0.213 and 0.153 mm were assigned to the trabecular thickness in the longitudinal and transverse directions, respectively. The resulting bone volume for each intact mesh was 0.134.

The sensitivity of modulus and strength to changes in trabecular microstructure was investigated independently reducing the thickness and number of trabeculae in each of the two directions. Each of the four parameters (longitudinal thickness, longitudinal number, transverse thickness and transverse number) was by an amount necessary to produce reductions in bone volume of 5%, 10% and 15%, while holding the other tree parameters at their intact values. Trabecular thicknesses were reduced uniformly, whereas the numbers of trabeculae were reduced by randomly removing trabeculae from the intact meshes.

Two additional analyses were performed, one to simulate aging and one to simulate a possible scenario for restoration of bone mass following the treatment of aged bone. To simulate the aging process, concurrent changes in the number and thickness of both longitudinal and transverse trabeculae were made to an intact mesh. The resulting mesh, with its random defects, qualitatively resembled the appearance of thin section of vertebral trabecular bone taken from old donors. The authors then simulated a scenario for the restoration of bone mass following drug treatment of aged bone, in which trabecular thickness increases without changes in the number of trabeculae. In this mesh of "treated" bone, the authors increased the thicknesses of the longitudinal and transverse trabeculae to restore bone volume to its intact value, while holding the number of trabeculae fixed at their aged values.

The authors concluded that the modulus and strength of the model were at least twice as sensitive to random reductions in the number of trabeculae as compared to bone volume-equivalent, uniform reductions in the thickness of trabeculae. For a case

simulating aged bone, in which the thickness and number of trabeculae were reduced concurrently, the modulus and strength were approximately 20% of their values for the intact (young) case. When a treatment that restores bone mass was simulated by increasing the thickness but not the number of trabeculae, the modulus and strength were increased by 60% and 75% respectively, compared to the aged case, but were still less than 40% of the values for the intact case.

The strengths of the model of Silva and Gibson (1997) are the accurate replication of the important microstructural features of vertebral trabecular bone. Also, vertebral trabecular bone was modeled in a generic fashion, rather than modeling the specific microstructure of individual specimens of bone. Furthermore, in this model the trabecular microstructure can be varied in a controlled fashion, and thus the effects of independent variations in microstructure on mechanical properties were investigated.

On the downside, the authors assumed a uniform thickness of trabeculae, with one value for all longitudinal trabeculae and a second value for all transverse trabeculae, not accounting for the natural variance in trabecular vertebral bone. Additionally, had the authors reduced trabecular thickness non-uniformly, they would probably have observed a greater decrease in strength for a given decrease in bone volume. Furthermore, trabeculae were reduced randomly rather than based on any initial state of stress or strain or based on any initial distribution of trabeculae thickness. As a result they might have overestimated the effects of trabecular removal on mechanical properties if resorption preferentially removes thinner and/or less heavily loaded trabeculae. Finally, the two dimensional nature of the model did not allow direct comparisons made between absolute values of modulus or strength predicted using their model and those measured on bone specimens experimentally.

4. SIMULATION RULES AND PARAMETERS

The aim of our work was to define and develop a set of simulation rules and parameters that allow the cellular processes of bone to be modelled and used in scenarios that investigate bone remodelling and the effects on its mechanical properties. The simulations are based on the concept of a basic multi-cellular unit (BMU). Models are represented by a regular 2D or 3D matrix of BMUs that correspond to either bone or marrow. A set of rules controls the activity of BMUs, such as the type of activity (resorption or reposition), amount of turnover, activity based on external stimuli and cell mobility.

The simulation ruleset operates at the microscopic level and is independent of the trabecular nature of bone. Consequently simulations do not only consider

uniform thinning or thickening of an artificially defined set of trabeculae, whilst bone activity and turnover is based on external mechanical stimuli. The parameters that control the amount of bone that can be resorbed or deposited are independent from each other and can be used to simulate natural bone loss, loss due disease or activity resulting from drug/hormone therapies.

4.1 Bone Remodelling Phases

Remodelling occurs on the surface of trabeculae (Frost 1997) with a three-stage cycle: activation, resorption and formation, such that the total cycle has a period of approximately 180-200 days. During remodelling, osteoclasts are responsible for the resorption of bone, forming a resorption pit that is subsequently filled with new collagen by osteoblasts. Typically, an osteoclast will remove the collagen to a depth of about 50 μm . The activity of osteoblasts is not always equal to that of osteoclasts and an imbalance in the osteoclastic and osteoblastic activity causes a net gain (+ $\Delta\text{B.BMU}$) or loss of bone (- $\Delta\text{B.BMU}$). In osteoporosis, there is a negative imbalance (- $\Delta\text{B.BMU}$) in remodelling usually caused by both increased osteoclast resorption and decreased osteoblast formation.

Each element in our simulations is represented by a square or cube of approximately 20 μm . In osteoporosis the actual size of - $\Delta\text{B.BMU}$ is about 4 to 5 μm . Although the minimum size of $\Delta\text{B.BMU}$ in the simulation is 20 μm , having a larger sized bone element reduces the computational requirements by a factor of 25 for 2D and 125 for 3D simulations, without unduly affecting the validity of the model (Fagan *et al* 1999). In the simulator, bone remodelling activity also occurs at the interface between bone and marrow.

4.2 BMU Activation Algorithms

The simulations start by identifying a set of elements to initiate resorptive or repository activity. The algorithm for this stage of the simulation determines the maximum number of elements to be probed in the model matrix using an activation frequency (*activation frequency* $F \in [0,1]$) and the total number of elements of the model. Each element that is randomly probed using a uniform deviation generator (Press *et al* 1996) is automatically activated, provided it is situated at the interface between bone and marrow.

4.3 Probabilistic Resorption And Reposition

During a resorption activation, a number of continuous bone BMUs along a bone/marrow perimeter of a matrix of BMUs are resorbed and become marrow. The maximum number of bone

BMUs to be resorbed for a particular activation is determined by its resorption *activation length function* (Equation 5). In similar terms, Equation (5) is used for repository activity as well.

$$L = \lceil b + a \cdot \text{random}() \rceil \quad (5)$$

A simulation run consists of a number of iterations N , where each iteration represents the remodelling that occurs over a period of time T . The *activation frequency* F defines the probability that a surface bone/marrow element will contain an activated BMU during the period T . For net resorption (net osteoclast activity), one or more bone elements become marrow, whereas for net formation (net osteoblast activity), one or more marrow elements are turned into bone. Net osteoclast activity is modelled as a resorption cavity that travels across the bone surface, whilst net osteoblast activity is modelled by a formation of collagen that travels along the bone surface. An activation consists of a channel of single elements that travels essentially in a straight line along a surface, whose length is controlled by Equation (5).

The ceiling function ($\lceil \dots \rceil$) rounds the result of the enclosed expression to the nearest integer towards $+\infty$. The function *random*() provides a normally distributed random number between $-\infty$ and $+\infty$ with mean 0 and variance 1 (Press *et al* 1996). For instance, constants a and b could be set to 1.8 and 6 respectively. Non-positive results of (5) are discarded, and since in 99.56% (Wonnacott and Wonnacott 1990) of cases $\text{random}() \in [-3, +3]$, L typically evaluates between 1 and 11. However, real study data (Eriksen *et al* 1990) report mean and standard deviation of bone turnover and can be used in relation to element or voxel size to determine appropriate values for a and b . Thus, constants a and b shift the probability distribution b units to the right or left and scale it by a (Law and Kelton 2000). By setting a to 0, L becomes deterministic in nature, allowing simulations to take place where the bone turnover is set at specific levels.

4.4 Cell Activity Control

Once an element is selected for resorption or reposition and the length of activity determined, the direction of an activation channel is chosen randomly. The four primary directions of travel are North, East, South and West (and Near and Far for 3D simulations). For an activated BMU, neighbouring elements to the N, S, E and W are checked. If only one of these is available the probability for the channel to go in that direction is 1.0. When two neighbours are available the probability of either direction to be chosen drops to 0.5, and decreases to 0.33 for three. Since an

element is considered to be on the surface when there is an element of the opposite type anywhere on the N, S, E or W of it, it is impossible to have four possible main directions of travel (and hence a probability of 0.25). The BMU activity thus attempts to progress the net resorption or reposition channel in this primary direction by L steps. The direction taken at each step is either the primary direction or a direction that deviates $\pm 45^\circ$ from the primary direction, provided that the element in this direction is of the same type as the element that started the activity. If this is not possible, a step that is $\pm 90^\circ$ from the main direction is attempted. If this too fails, a step of $\pm 135^\circ$ is attempted and as a last resort, $+180^\circ$. In general, when a step greater than $\pm 45^\circ$ is taken, the primary direction is also changed. This prevents a channel from moving backwards and forwards between two positions and creating a “deep pit” or a “high hill”. The step direction is chosen randomly from all possible directions with equal probability.

At the end of an iteration any isolated marrow or bone islands are removed. This seems reasonable, as it is known that unattached trabeculae are resorbed. Once the primary direction of a net osteoclast or osteoblast activation has been determined, a step of an activation can only be taken if one of the candidate elements for the step is on a bone/marrow surface.

4.5 Relationship Of Strain And Remodelling

Since bone adapts to loading conditions, bone areas under low strain are also preferentially resorbed and areas of high strain reinforced by reposition of bone. Areas of bone under normal strain are mostly unaffected by remodelling processes. Support for strain-remodelling is achieved by associating strain with each element in the 2D/3D matrix and using that to initiate resorption or reposition of bone based on the value of strain for any particular element. Strain energy density values are obtained from linear finite element analyses. Low values of strain would primarily contribute to loss of bone (disuse) and high values to addition of bone (Wolff 1896).

Two limits are associated with the above, where ε_1 and ε_2 are the strain-remodelling limits. Elements whose strain is less than ε_1 become potential candidates for resorption, whereas elements with strain above ε_2 become potential candidates for reposition of bone. The remaining elements whose strain is between ε_1 and ε_2 are not considered for any of the two types of activation.

$$StrainAdaptation(s, \varepsilon_1, \varepsilon_2) = \begin{cases} CL, s \in (-\infty, \varepsilon_1] \\ N, s \in (\varepsilon_1, \varepsilon_2) \\ BL, s \in [\varepsilon_2, +\infty) \end{cases}$$

(6)

Where s represents the strain-energy density associated with a probed element in the model matrix and ε_1 and ε_2 are strain-adaptation limits. Function (6) returns the type of activity to be initiated and can be osteoclastic (CL), none/no activity (N or neutral) and osteoblastic (BL).

5. VALIDATION AND VISUALISATION

5.1 Structural Morphology

The analysis of the morphology of the structures produced by our simulations (Fagan *et al* 1999, Langton *et al* 1998, Langton *et al* 2000) allows the indirect validation of the simulations by comparing the results with the literature, and facilitates the investigation of scenarios such as the effects of structural characteristics at the trabecular level to the structures' stiffness. Algorithms were defined that can compute morphological indices directly from 3D, offering a more accurate picture than current approaches where some metrics are calculated on a per voxel-plane basis (being effectively 2D). These algorithms have been incorporated into our simulations and are able to perform histomorphometric analyses on binary pixel and voxel maps without any user intervention (Sisias *et al* 2002).

The morphological indices computed are model independent and are derived directly in 3D using techniques such as volume ray-tracing. The first group involves indices such as mean trabecular thickness, mean trabecular separation (or spacing) and trabecular density (or number). The second group involves the computation of the star area and volume distribution of both bone and marrow elements. The third group considers simple indices such as bone and marrow fraction volume, surface roughness and perimeter/surface area. Other structural measurements involve the strain energy density distribution across all bone elements in a structure and the calculation of the structure's stiffness.

5.2 Interactive Visualisation

Three-dimensional visualisation of trabecular bone and its attributes is an essential tool in understanding this remodelling process for cancellous bone. It enables the bone researcher to quickly understand the dynamic behaviour of remodelling, the resulting geometry of the bone structure and it allows alternative remodelling scenarios to be compared. Phillips *et al* (2003) discuss stereoscopic

visualization of bone structures using a volume rendering technique based on transparency of voxels integrated with a reflection model. The method the authors employ is more appropriate than surface rendering as it allows the inside of trabeculae to be viewed. The volume rendering implementation is based on texture mapping. It runs on a 1.1 GHz Athlon PC with 512 MB RAM and an NVIDIA GeForce 2 Ultra graphics card. The texture mapping technique used is preferred to ray-casting as it is less computationally intensive and it provides real-time interactive visualization on mainstream hardware.

6. APPLICATIONS

Langton *et al* (1998) developed and applied a stochastic simulation of cancellous bone resorption to the simple two-dimensional lattice structure representing vertebral bone. The study described a stochastic simulation of net bone resorption at a microscopic level, exhibiting both trabecular thinning and perforation. Finite element analysis was used to quantify the effects of resorption on the mechanical properties of bone after each simulation iteration. The structure after each step was analysed with FEA with a simple compressive load, to compute the nodal displacements. The relationship between relative stiffness and density as function of the simulation step was derived, along with the relationship between stiffness and bone porosity. In the simulations the structure began to suffer from loss of vertical trabecular connectivity from step 3 and started to collapse from step 4. By step 8 the structure totally lacked connectivity and mechanical integrity. Relative stiffness decreased more rapidly than density. Consequently the stiffness decreased faster than porosity for the first few steps and levelled to zero for the final steps as the structure lost connectivity and collapsed.

Fagan *et al* (1999) investigated the effects of mesh density and model size to the simulations of Langton *et al* (1998). Structures with 50% less and 50% more elements for both horizontal and vertical trabeculae than the structure of Langton *et al* (1998) were created. These were subsequently subjected to the same loading conditions as above and the results of their finite element analyses compared to those from the previous experiments. Furthermore the effect of model size was examined by setting the number of trabeculae to 3×3 and 9×9. As in the previous experiments, simulations were repeated five times with each of the models.

Langton *et al* (2000) used a simplistic symmetric lattice structure consisting of 5×5 trabeculae with constant width and intertrabecular spacing to stochastically resorb and rebuild bone until the structure's original stiffness was regained. The structure was resorbed using an activation frequency

of 0.05 (5%) and the activation length function L of Section 4. Constants a and b were set to 1.8 and 6, respectively. At the 95th percentile, L was 120 ± 70.6 μm at 20 μm resolution. Resorptive activity was continued until nominal resorptions of approximately 10%, 15%, 20%, 25% and 30% below the original density were achieved. The simulation was then modified to create an anabolic effect (with the same parameters as above) where bone elements were added at the bone-marrow interface stochastically, providing a rate of net formation ($+\Delta B.BMU$). The simulation of anabolic treatment was applied at the resorbed structures until the original stiffness had been regained. The simulations were repeated three times for each of the nominal resorptions. The simulations started with the intact structures, at a relative density and stiffness of 1.0. The stochastic simulation reduced density to the various levels. At that point anabolic treatment was simulated, until original density and stiffness were reached. Although original density was eventually reached, stiffness was not totally restored. Restoration of stiffness required density to increase to levels above 100%, especially for the most severely depleted structure (at 30%).

7. Discussion And Conclusions

The various models described address the issues of normal or hormonally affected bone growth, loss or adaptation from different and concentrated perspectives. Some models consider structural characteristics of cancellous bone such as trabecular network, whereas others target bone simulation at the microscopic level, such as multi-cellular units of a few μm in size. Models based on structural characteristics mainly tend to simulate the behaviour of large sections of bone, based on statistical data obtained from clinical studies. On the other hand models that operate on the microscopic level try to closely represent the small simulated structures and consider the activity of individual or groups of cells.

Model validations tend to present difficulties. The results obtained from statistically based simulations are compared to clinical studies, and mainly consider normal or hormonally affected bone loss or growth. Similarly the results obtained from microscopic simulations are related to real bone samples taken from biopsies of normal or osteoporotic patients of either sexes or women only when simulating the effects of menopause. Although the later simulations are generally aimed to be more accurate, they suffer from the fact that once bone is taken from a human, growth is non-existent, either the sample was taken postmortem or growth stopped after the biopsy. Additionally, biopsies from live human give very small samples, are invasive and cannot supply a second sample from the same region as the original after the end of treatment or menopause. More

importantly, mainly due to high radiation levels, samples might be impossible to take non-invasively from live subjects.

Microscopic simulations, although more accurate, tend to suffer from the high number of elements of the models and the cellular processes and finite element analyses considered on every element. Once technological restrictions relax, it should be possible to simulate the activity of cells on fine resolution voxel-based data sets, taking under consideration stochastic factors and bone adaptation. However, algorithmic efficiency in terms of storage and operations can play a decisive role in the size of structures and complexity of rules that can be employed. In terms of software development, though, more exotic solutions to projects pose restrictions in the availability of methods to be employed, particularly for finite element analyses.

The methods outlined are versatile in terms of the scenarios that can be investigated. As the scenarios increase in complexity and size, it becomes necessary to consider significant revisions of the underlying software to accommodate new modes of operation. As the software grows in size (presently the entire simulator suite is about 70 KLOC) alterations are more difficult to implement and test.

In comparison to the models reviews, the simulation ruleset outlined operates at the microscopic level and is independent of the trabecular nature of bone. Consequently simulations do not only consider uniform thinning or thickening of an artificially defined set of trabeculae, whilst bone activity and turnover is based on external mechanical stimuli. The parameters that control the amount of bone that can be resorbed or deposited are independent from each other and can be used to simulate natural bone loss, loss due disease or activity resulting from drug/hormone therapies. However, this versatility comes at a cost, as it is computationally expensive and causes the manifestation of the usual problems associated with microscopic simulations.

8. References

- [1] Baron, R.E. (1993) Anatomy and ultrastructure of bone. *Primer on the metabolic bone diseases and disorders of mineral metabolism*. 2nd ed., 3-10.
- [2] Eriksen, E.F., Hodson, S.F., Eastell, R., Cedel, S.L., O'Fallon, W.M. and Riggs, B.L. (1990) Cancellous bone remodeling in Type I (Postmenopausal) osteoporosis: quantitative assessment of rates of formation, resorption, and bone loss at tissue and cellular levels. *Journal of Bone and Mineral Research*, 5 (4), 311-319.
- [3] Fagan, M.J., Dobson, C.A., Ganney, P.S., Sias, G., Phillips, R. and Langton, C.M. (1999) Finite element analysis of simulations of cancellous bone resorption. *Computer Methods in Biomechanics and Biomedical Engineering*, 2, 257-270.

- [4] Frost, H.M. (1997) On our age-related bone loss: insights from a new paradigm. *Journal of Bone and Mineral Research*, 12 (10), 1539-1546.
- [5] Gibson, L.J. and Ashby, M.F. (1988) Cellular solids: structure and properties, *Pergamon Press*.
- [6] Kinney, J.H. and Ladd, A.J.C. (1998) The relationship between three-dimensional connectivity and the elastic properties of trabecular bone. *Journal of Bone and Mineral Research*, 13 (5), 839-845.
- [7] Lacy, M.E., Bevan, J.A., Boyce, R.W. and Geddes, A.D. (1994) Antiresorptive drugs and trabecular bone turnover: validation and testing of a computer model. *Calcified Tissue International*, 54, 179-185.
- [8] Langton, C.M., Haire, T.J., Ganney, P.S., Dobson, C.A. and Fagan, M.J. (1998) Dynamic stochastic simulation of cancellous bone resorption. *Bone*, 22 (4), 375-380.
- [9] Langton, C.M., Haire, T.J., Ganney, P.S., Dobson, C.A., Fagan, M.J., Siasias, G. and Phillips, R. (2000) Stochastically simulated assessment of anabolic treatment following varying degrees of cancellous bone resorption. *Bone*, 27 (1), 111-118.
- [10] Law, A.M. and Kelton, W.D. (2000) *Simulation modeling and analysis*. 3rd ed. USA, McGraw-Hill Higher Education.
- [11] Reeve, J. (1986) A stochastic analysis of iliac trabecular bone dynamics. *Clinical Orthopaedic Rel. Research*, 213, 264-278.
- [12] Recker, R.R., Kimmel, D.B., Parfitt, A.M., Davies, K.M., Keshawar, N. and Henders, S. (1988) Static and tetracycline-based bone histomorphometric data from 34 normal postmenopausal females. *Journal of Bone and Mineral Research*, 3, 133-144.
- [13] Siffert, R.S., Luo, G.M., Cowin, S.C. and Kaufman, J.J. (1996) Dynamic relationships of trabecular bone density, architecture, and strength in a computational model of osteopenia. *Bone*, 18 (2), 197-206.
- [14] Silva, M.J. and Gibson, L.J. (1997) Modeling the mechanical behaviour of vertebral trabecular bone: effects of age-related changes in microstructure. *Bone*, 21 (2), 191-199.
- [15] Phillips, R., Grunchev, J.-A., Ward, J.W., Fagan, M.J., Dobson, C.A., Langton, C.M. and Siasias, G. (2003) Stereo visualisation of 3D trabecular bone structures produced by bone remodelling simulation. *11th Annual Medicine Meets Virtual Reality Conference*. California, USA.
- [16] Press, W.H., Teukolsky, S.A., Vetterling, W.T. and Flannery, B.P. (1996) *Numerical recipes in C*. 2nd edition. USA, Cambridge University Press.
- [17] Siasias, G., Dobson, C.A., Phillips, R., Fagan, M.J. and Langton, C.M. (2002) Histomorphometric algorithms for the direct derivation of morphological indices of simulations of strain-adaptation in cancellous bone. *6th International Conference on Information Visualisation: Medical Visualisation (IEEE IV2002)*. London, UK, pp. 592-599.
- [18] Wonnacott, T.H. and Wonnacott, R.J. (1990) *Introductory statistics*. 5th ed. Republic of Singapore, John Wiley & Sons, Inc.

1. Acknowledgements

The authors gratefully acknowledge the financial support of the EPSRC, charity OSPREY, Action Research and the Hull and East Yorkshire Hospitals NHS Trust.

Contact: George Siasias, Department of Computing, School of Informatics, University of Bradford, Bradford, BD7 1DP, UK. Tel.: (+44) 01274 233908, Fax: (+44) 01274 233920, E-mail: G.Siasias@bradford.ac.uk.

MODELLING POPULATIONS OF PROKARYOTIC CELLS: the n -Layered mRDG Approximation

G. CHLIVEROS, M.A. RODRIGUES and D. COOPER

*Computer Vision Pattern Recognition & AI Group, Computing Research Centre
Sheffield Hallam University, Sheffield S1 1WB, UK*

http://www.shu.ac.uk/scis/artificial_intelligence

Abstract: In this paper, explicit expressions for the Scattering Amplitude elements of spherically symmetric inhomogeneous particles using the modified RDG approximation (mRDG) are derived. Computer simulation algorithms have been developed for the calculation of Scattered Light Intensity (full and backscattering) from a multi-layered sphere with an arbitrary number of layers. All quantities are estimated within the biological cell domain and in particular that of prokaryote. We have extended a previously proposed size distribution to account for the evident size asymmetry in nature. Simulation results show that the proposed model's rapid calculations are comparable in performance with that of Mie or RDG models. Finally, to the best of our knowledge, the included relative error study between these theories and for n -layered spheres is the first to appear.

Keywords: Light Scattering, Prokaryotic Cells, Asymmetric Populations, mRDG, Mie Scattering

1 INTRODUCTION

Light scattering measurements and in particular multi-angle (laser) light scattering has been of great interest in many fields of microscopic characterisation. In particular it has been indicated that laser scattering techniques will play a significant role in partial identification [Newman 1987], characterisation [Van de Merwe et.al. 1997] and clinical examination [Mourant et.al. 1998] of bacteriological samples. Optical data obtained from a circular array of photo detectors are usually interpreted by means of the Rayleigh-Gans-Debye (RDG) approximation [Wyatt 1993] or Mie theory homogeneous models, even though other theories have been developed (e.g. [Draine and Flatau 1994]).

However, most prokaryotic cells are of a complex makeup. In general the cell presents a structure that consists mainly of the cell wall, the plasma or cytoplasmic membrane, the cytoplasm and the nucleoid. Other morphological characteristics may also appear such as a slime layer (capsule) outside the cell wall or inclusions within the cell's cytoplasm (e.g. spores, granules). Therefore, in order to generate a more accurate representation of the cell, one would model it as having various compartments within its volume and within these compartments the refractive index is different from that of the surrounding objects. In cells where the overall morphology is that of a sphere (cocci), or if we allow for an approximate representative spherical model and for non-symmetric particles, each of the structures internal or external to the plasma membrane can be modelled as a different layer in an n -layered spherically symmetric inhomogeneous particle.

Biological particles, including bacteria, contain weakly scattering material, mainly because most of their bodies contain a high percentage (70% to 86%) [Schlegel 1997] of water. This alone supports the use of RDG. However conditions underlined in this approximation pose size restrictions and possible rise in the relative error. To accommodate for this, in [Shimizu 1983] an extension of RDG has been provided, also known as mRDG. In [Sloot and Fidgor 1986] this approximation has been generalised to a two-layered spherical particle and successfully applied for predictions in nucleated blood cells, but with no consideration of size variations in cell populations.

In this paper we extend the theory of mRDG to a spherically symmetrical particle/cell with an arbitrary number of layers and corresponding relative refractive indices. Population variations in size have also been accounted for and for asymmetry (positive or negative skewness of a size distribution) in nature. Intensity expression for the latter is provided within. Finally, from previous studies [Wyatt 1973, Volkov and Kovach 1990] we advocate the use of mRDG for biological, prokaryotic cells and provide a comparison of mRDG and Mie derived models for the n -layered sphere.

2 THE n -LAYERED SPHERE MODEL

As previously mentioned we consider the case of a spherical model for the prokaryotic cell as an inhomogeneous particle consisting of a multi-layered sphere with an arbitrary refractive index within each layer.

Suppose that there are n layers, such that the i th layer has outer radius r_i and relative refractive index m_i . Thus for a radially changing $m(r)$,

$$m(r) = \begin{cases} m_1, & r \in (0, r_1] \\ m_2, & r \in (r_1, r_2] \\ \vdots & \\ m_n, & r \in (r_{n-1}, r_n] \end{cases} \quad (1)$$

It is known [Bohren and Huffman 1998] that in the RDG regime the scattering amplitude S of a cell of volume V at scattering angle θ and for perpendicular polarisation to the scattering plane (and hence the subscript \perp in S_\perp below) can be expressed as follows:

$$S_\perp(\theta) = \frac{jk^3}{2\pi} \int_V (m(r) - 1) \exp\left(j2kr \sin \frac{\theta}{2}\right) dV \quad (2)$$

In Equation 2, $S_\perp(\theta)$ is a complex number and j denotes $\sqrt{-1}$, whilst k is the propagation constant in the water medium ($k = 2\pi/\lambda$, where λ is the wavelength of the incident light). For a spherical cell, the integrand in Equation 2, in polar coordinates, depends only on the distance r from the origin, and consequently the triple integral can be replaced by a single integral, with the volume element $dV = 4\pi r^2 dr$. We have:

$$S_\perp(\theta) = j2k^3 \int_0^{r_n} r^2 (m(r) - 1) \exp\left(j2kr \sin \frac{\theta}{2}\right) dr \quad (3)$$

In the RDG approximation it is assumed that the applied field inside the particle equals that in the medium. Hence, the propagation constant in and out of the particle's region is unchanged. Shimizu [1983] has extended the RDG by altering the propagation constant to accommodate for the contributions resulting from the field inside the particle. As a result, within the phase lag expression the particle's refractive index is taken into account so that now k is replaced by $km(r)$. With hindsight and using the method of slices [Wyatt 1973] Equation 3 is replaced by

$$S_\perp(\theta) = j2k^3 \int_0^{r_n} r^2 (m(r) - 1) \frac{\sin\left(2km(r)r \sin \frac{\theta}{2}\right)}{2km(r)r \sin \frac{\theta}{2}} dr \quad (4)$$

Evaluating Equation 4 in the region $r \in [0, r_n]$ and using Equation 1 we now get

$$S_\perp(\theta) = j2k^3 \left((m_1 - 1) \int_0^{r_1} r \frac{\sin\left(2km_1 r \sin \frac{\theta}{2}\right)}{2km_1 \sin \frac{\theta}{2}} dr + \dots \right. \\ \left. + (m_n - 1) \int_{r_{n-1}}^{r_n} r \frac{\sin\left(2km_n r \sin \frac{\theta}{2}\right)}{2km_n \sin \frac{\theta}{2}} dr \right)$$

resulting in

$$|S_\perp(\theta)| = k^3 \sqrt{2\pi} \left(K_{1,1} J_{3/2}(2km_1 r_1 \sin \frac{\theta}{2}) + \dots \right. \\ \left. + (K_{n,n} J_{3/2}(2km_n r_n \sin \frac{\theta}{2}) - K_{n,n-1} J_{3/2}(2km_n r_{n-1} \sin \frac{\theta}{2})) \right) \quad (5)$$

where $J_{3/2}$ is the Bessel function of order $\frac{3}{2}$, we write $r_0 = 0$, and, for $i, \ell \in \mathbb{N}$,

$$K_{i,\ell} = (m_i - 1) \sqrt{\left(\frac{r_\ell}{2km_i \sin \frac{\theta}{2}}\right)^3} \quad (6)$$

For a more compact model we write

$$G_{i,\ell}(\theta) = J_{3/2}(2km_i r_\ell \sin \frac{\theta}{2}) \quad (7)$$

so that Equation 5 now becomes

$$|S_\perp(\theta)| = k^3 \sqrt{2\pi} \sum_{i=1}^n (K_{i,i} G_{i,i}(\theta) - K_{i,i-1} G_{i,i-1}(\theta)) \quad (8)$$

bearing in mind that $K_{i,0} = G_{i,0} = 0$. The expression in Equations 6, 7 and 8 predicts amplitude of light scattered from a single cell and it is the n -layered sphere extension model. It can be applied to any population of n -layered spheres and would lead to better approximations of light scattered phenomenon on real cells by simulated models. In effect its physical meaning corresponds to the fact that a cell of n layers will scatter light proportional to the sum of n homogeneous spheres of corresponding r_n and m_n , by subtraction of contributions arising from the $(n - 1)$ homogeneous spheres of corresponding r_{n-1} but having the same refractive index, that is m_n .

This generalised expression correctly predicts the effect of removing layers. Putting $m_{k-1} = m_k$ will result in a multi-layered sphere where the $(k - 1)$ th layer will disappear. This is true since the previous $(k - 1)$ th and k th layers will merge to a new layer with $m_{\text{new}} = m_k = m_{k-1}$, of thickness¹ t_{new} such that $t_{\text{new}} = t_k + t_{k-1}$. Furthermore, if $m_k = 1$ then the k^{th} layer becomes redundant, which is true since this layer becomes transparent to incoming light and as such does not contribute to the scattering amplitude.

The light intensity from such a cell, and for perpendicular incident polarisation, can be expressed in terms of S_\perp using the following expression:

$$I(\theta) = \frac{I_0}{2(kr)^2} |S_\perp(\theta)|^2 \quad (9)$$

where $r = r_n$ is the overall radius of the spherical cell and I_0 is the intensity of the incident light.

¹Note that, for example, $t_k = r_k - r_{k-1}$

3 SINGLET POPULATION

For cells that appear alone, that is, where there is no binding of cells together, and for low densities so that multiple scattering is avoided, the average scattering pattern can be calculated using a size distribution. The term “size” in the current context should be interpreted as the radius of the cell, but in general would be thought of as the length of the minor or major axis of an ellipsoid form (e.g. rod like cells). The cell size is denoted by s . We use a probability density function $P(s)$ for the size, and assume that we have N size ranges with mid-points s_1, s_2, \dots, s_N . The relative frequencies of the cell samples in the ranges are approximated by the density function at the mid-points, so that the mean light intensity at scattering angle θ is given by

$$\langle I(\theta) \rangle = \frac{\sum_{i=1}^N I(\theta)_{r=s_i} P(s_i)}{\sum_{i=1}^N P(s_i)} \quad (10)$$

Multiple scattering is a problem that cannot be addressed using Equation 10. However, Equations 6–8 are used for modelling an aggregate’s discrete scattering elements of any bounded configuration with no multiple scattering.

Often a Gaussian distribution of cell sizes is assumed. However, the normal distribution has long tails, which is rather unrealistic since, in the bacteria domain, sizes do not exceed a specific range. Moreover, from a variety of sources of variability, usually only a few are dominant. This results in a positively or negatively skewed distribution, which does not resemble the familiar Gaussian symmetry. Consequently, we have adopted a distribution first proposed in [Wyatt 1973], but here we have allowed for κ in Equation 12 to be assigned independently at the left and right of the mode. The density function is proportional to

$$P(s) = \begin{cases} (1 - z^2)^4 & \text{for } z \in [-1, 1] \\ 0 & \text{for } z \notin [-1, 1] \end{cases} \quad (11)$$

where

$$z = \begin{cases} 1.084(s - s_0)/(\kappa_{\text{left}} s_0) & \text{for } s \leq s_0 \\ 1.084(s - s_0)/(\kappa_{\text{right}} s_0) & \text{for } s > s_0 \end{cases} \quad (12)$$

The spread of the distribution is dictated by the constant κ which is assigned independently at the left and right of the mode s_0 , resulting in an asymmetric distribution that avoids long tails. It should be evident that for $\kappa_{\text{left}} = \kappa_{\text{right}}$ the distribution is symmetric and s_0 becomes the mean; whilst κ is approximately equal to $3\hat{\sigma}/s_0$ with $\hat{\sigma}$ being the variability measure (standard deviation) of the symmetric distribution. It is known that in any non-synchronised culture and in nature we expect a variation in size of at least 30% ($\kappa_{\text{left}} + \kappa_{\text{right}} \geq 0.30$). The latter applies not only to singlet spheres but also to any other configuration of cocci bacteria.

4 SIMULATION RESULTS

Bacteria sizes vary considerably, from half micrometer up to several micrometers. In particular, cocci (spherical morphology) would be said to have a radius r within the range $0.5\mu\text{m} \leq r \leq 1.2\mu\text{m}$ with a few exceptions such as *Sarcina ventriculi* with a $4\mu\text{m}$ radius and spore inclusions. In scattering experiments, cells are usually suspended in water based media and so the relative refractive index m is close to unity and the cytoplasm’s refractive index value is close to 1.35, resulting in a selected range for m in the studies reported here as $1 < m < 1.3$.

Following the criteria set by [Hoekstra and Sloot 2000], we present a relative error study for values of relative refractive index and radius as discussed. However, since we are dealing with multiple layers, the examination of single particle scattering is introduced in more detail. Hence, for each cell size defined by an overall radius, the thickness of each layer is defined by the use of uniform random numbers. The relative error is estimated over an average of R runs, where for each run a corresponding random relative refractive index value has been provided within the range of interest. In the analysis, only the average refractive index m of the cell is illustrated for each value of radius r .

The error metric E_R is equivalent to [Hoekstra and Sloot 2000] but here we examine the light scattering intensity as opposed to the phase matrix relations. In particular, the error is a measure of the difference between intensities estimated by Mie and mRDG models and is normalised as:

$$E_R = \frac{\sum_{i=0}^N |\log I^{\text{Mie}}(i\Delta\theta) - \log I^{\text{mRDG}}(i\Delta\theta)|}{(N+1)(\log I^{\text{Mie}}(0) - \log I^{\text{Mie}}(\theta_o))} \quad (13)$$

The values used in the simulations were $N = 91$, $R = 30$ and $\Delta\theta = \pi/N$. Moreover, at a scattering angle θ_o the light intensity of the Mie scattering function (I^{Mie}) is at minimum. Figure 1 depicts typical light intensity patterns for the Mie and mRDG models which are the basis for error evaluation through Equation 13.

Many authors including [Hoekstra and Sloot 2000] have concluded that for a homogeneous sphere the mRDG model covers a significant part of the domain; particularly if one allows for error of 12% as compared to Mie scattering. However, we have found that in the case of multi-layered spheres this relative difference doubles. In particular, Figure 2 depicts the error map between Mie scattering model and mRDG for two layer spheres. The gray scale represents the average relative error from 0% (white) to 33% (black). Generally speaking, in Figure 2 the error does not exceed the limit of approximately 23%, even though small areas of 33% do appear. The latter can be verified by consulting

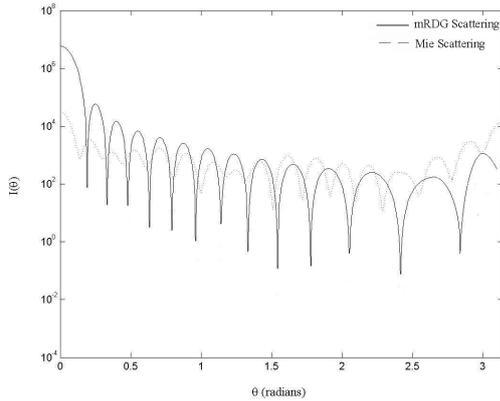


Figure 1: Layered mRDG and Mie light scattering patterns for a two layer concentric sphere.

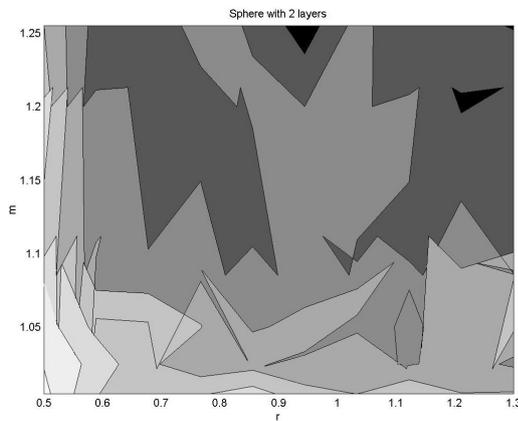


Figure 2: $n = 2$. Error map between mRDG and Mie scattering for a two layer spherical model.

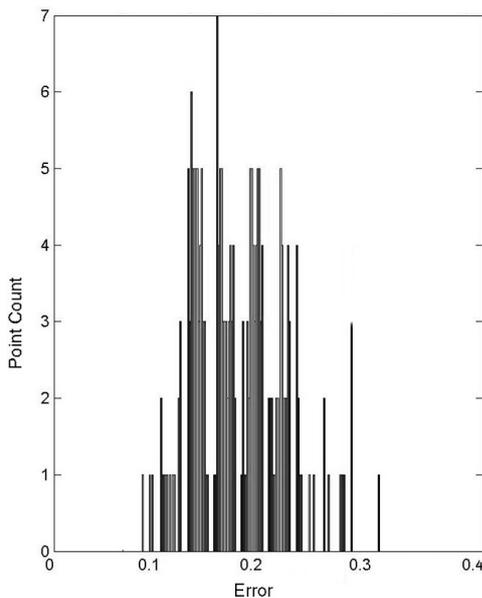


Figure 3: Error histogram for a two layered sphere.

the error histogram of Figure 3 which shows that most difference between the two models lie between 15 and 23%. This error or difference is consistent throughout the two models either for 2, 3, 4 or 5 layers. The mRDG model is, in fact, just an alternative representation for Mie scattering in this context. Moreover, if one considers that Mie algorithms are at least 100 times slower (or more depending on programming skills), as opposed to their RDG or mRDG counterparts, there are significant advantages in using the alternative representation of mRDG model as proposed here.

In Figures 4 to 6 it must be emphasized that as the number of layers increases the maximum relative error margin slightly shifts towards higher r values and covering a larger m value margin. As a matter of fact [Volkov and Kovach 1990] state that for near index particles (high water content) the key factor in the Mie scattering behaviour is the thickness of the layers. As such, it may seem rather surprising that the relative error increases not due to the r values but due to the average refractive index as it is evident in Figure 5. This may mean that Mie theory is not particularly sensitive to changes in refractive index for larger values of radius. This indeed may have given rise to the relative error not attributed to the mRDG approximation. Returning to the earlier rare example of *Sarcina ventriculi*, in an experiment of $r = 4\mu\text{m}$ and for various m values, the average relative error was found to be in the region of 3 to 27%; with the latter arising as $m \rightarrow 1.3$. Finally, within the domain of Prokaryotic cells such large m -values are rarely found and, as such, the use of mRDG model as proposed here is justified.

Testing the relative error of bacteria populations, as introduced in Section 3, has been performed using the same procedure. The population analysis yields very similar results and so further illustrations are not included. We must however highlight the fact that as the spread of the size distribution increases the relative error remains within the same margins. Therefore, the apparent smoothing of sharp maxima (or minima) in the scattering intensity does not indicate degradation in performance of the n -layer mRDG model.

5 CONCLUSIONS

In this paper we have derived a new model for the multi-layer sphere problem based on the mRDG approximation and used it to simulate light scattering phenomena in bacteria cells. In order to assess the performance of the model, computer algorithms were developed in Matlab and compared with the equivalent Mie scattering model. An error parameter was defined based on a measure of the difference between Mie and mRDG scattering. All simulations have been conducted using sizes and refractive indices in accordance with values found in bacteria cells.

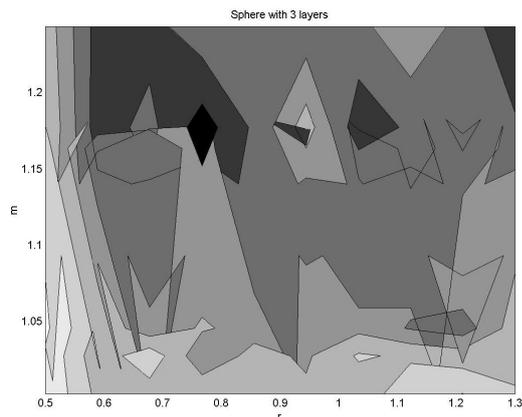


Figure 4: $n = 3$. Error map between mRDG and Mie scattering for a three layer spherical model.

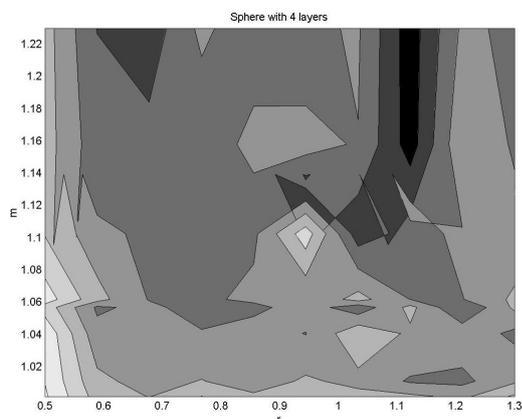


Figure 5: $n = 4$. Error map between mRDG and Mie scattering for a four layer spherical model.

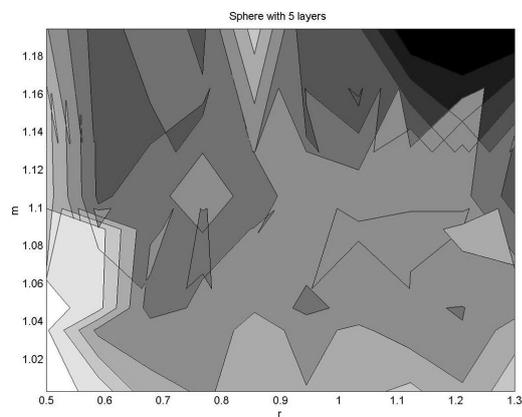


Figure 6: $n = 5$. Error map for a five layer spherical model between mRDG and Mie scattering

It appears that the difference between the two models is at its maximum at about 25%. We have used the term *relative error*, which does not necessarily portray the expected error under true experimental conditions. In particular, one has to bear in mind the much faster computation of the mRDG models as opposed to the Mie equivalents. This can be explained as follows. Calling t the number of terms to be calculated in the Mie series and n the number of layers, and l_{mie} the scattering coefficients, there would be a minimum of $(l_{mie}nt)$ calculations. The equivalent number for mRDG scattering would be $(l_{mrdg}(2n - 1))$. In our implementation of both models on the same platform, Mie models were at least 100 times slower than the RDG or mRDG counterparts. As a result, for real time or time critical applications the mRDG approximation is expected to be favoured over other more complex theories.

The consistency of errors throughout the two models indicate that the mRDG is a convenient alternative representation for light scattering phenomena and its superior computational performance brings obvious advantages to cell characterisation.

Further research include relative error studies with other theories that can be applied in the domain of interest such as Anomalous Diffraction scattering (AD) and variants of this approach, higher order RDG, Discrete Dipole Approximation (DDA) and Physical Optics (PO) to name just a few. The most important further development of this work would be the generation of true scattering patterns from benchmark prokaryotic cells, and consequently, compare the mRDG, AD and PO calculations with rigorous numerical methods such as the DDA. Research is underway and will be reported in the near future.

Acknowledgments

We acknowledge partial financial support from *Trident Management Ltd* (UK).

References

- [Bohren and Huffman 1998] Bohren C.F., Huffman D.R., 1998. *Absorption and Scattering of Light by Small Particles*. New York: Wiley
- [Draine and Flatau 1994] Draine B.T., Flatau P.J. 1994. Discrete Dipole approximation for scattering calculations, *J.Opt.Soc.Am.A*, 11(4), 1491-1499.
- [Hoekstra and Sloot 2000] Hoekstra, A.G. and Sloot P.M.A. 2000. Biophysical and Biomedical Applications of Non-spherical Scattering. In: M.I. Mishchenko, J.W. Hovenier and L.D. Travis (eds.) *Light Scattering by Nonspherical Particles: Theory, Measurements and Applications*. Academic Press: 585-602.

- [Mourant et.al. 1998] Mourant J.R., Freyer J.P., Hielscher A.H., Eick A.A., Shen D., Johnson T.M., 1998. Mechanisms of light scattering from biological cells relevant to non invasive optical-tissue diagnosis, *Applied Optics*, 37(16), 3586-3593.
- [Newman 1987] Newman C.D. 1987. *Measurement of the scattering matrix as a means for bacterial identification*. University of New Mexico: Ph.D. thesis.
- [Schlegel 1997] Schlegel H.G. 1997 (7th ed.). *General Microbiology*. London: Cambridge University Press
- [Shimizu 1983] Shimizu K., 1983. Modification of the Rayleigh-Debye Approximation, *J.Opt.Soc.Am.*, 73, 504-507.
- [Sloot and Fidgor 1986] Sloot P.M.A., Fidgor C.G. 1986. Elastic Light Scattering from nucleated blood cells: rapid numerical analysis, *Applied Optics*, 25(19), 3559-3565.
- [Van de Merwe et.al. 1997] Van de Merwe W.P., Li Z.-Z., Bronk B.V., Czege J. 1997. Polarised light scattering for rapid observation of bacterial size changes, *Biophysical Journal*, 73, 500-506.
- [Volkov and Kovach 1990] Volkov N.G., Kovach V.Yu. 1990. Scattering of Light by Inhomogeneous Spherically Symmetrical Aerosol Particles, *Izvestiya Atmospheric and Oceanic Physics*, 26(5), 381-385.
- [Wyatt 1973] Wyatt, P.J. 1973. Differential Light Scattering Techniques for Microbiology. In: J.R. Norris and D.W. Ribbons (eds.) *Methods in Microbiology*, Volume 8. Academic Press: 183-263.
- [Wyatt 1993] Wyatt P.J. 1993. Light scattering and the absolute characterisation of macromolecules, *Analytica Chimica Acta*, 272, 1-40.

BIOGRAPHY

G. Chliveros received a *BEng(Hons)* degree in *Electronics and Instrumentation Systems* in 1999 from the University of Wales (UK), and the *MSc* degree in *Statistics* in 2000 from the University of the West of England (Bristol, UK). He is currently pursuing a PhD in Pattern Recognition at Sheffield Hallam University under the supervision of Professor M.A. Rodrigues and Dr D. Cooper.

His interests include the development of optoelectronics instrumentation in laser scattering systems and pattern recognition research under the domain of aquatic bacteria. Within this framework he is also interested in mathematical-statistical modelling and simulation of bacteria morphology and fractal aggregates. He is a member of the IEE (MIEE).

VISUALISING SPECIATION IN MODELS OF CICHLID FISH

ROSS CLEMENT

*Department of Artificial Intelligence and Interactive Multimedia,
Harrow School of Computer Science
University of Westminster
Email: clemenr@wmin.ac.uk*

Abstract: An Agent-Based model of speciation in cichlid fish has been implemented. When run, this generates large amounts of trace data in which speciation is an implicit, near unobservable, processes. Fuzzy C-Means Clustering is used to identify species extant at the end of simulation, and the power set of these species is the potential set of ancestral species. Membership values for all fish in each of these theoretical ancestral species are calculated, and total set membership for each of these species is plotted against time. The resulting graph is to be a clear visualisation of the process of speciation, and the appearance and disappearance of intermediate species. Our approach allows the visualisation of speciation resulting in larger numbers of final species than was possible using previous techniques based on measuring correlations between explicit properties of modeled organisms, and is also unaffected by changes to the properties used to model fish.

Keywords: Evolution, Speciation, Data Visualisation, Fuzzy Sets, Cichlids

1. INTRODUCTION

The Cichlid fish are one of the great mysteries of evolution [Barlow, 2000]. In Lake Victoria, hundreds of species of Cichlid have evolved from one or two ancestors in approximately 14,000 years [Seehausen, 2002]. In the tiny crater lake Barombi-Mbo, Cichlid fish have speciated despite any observable physical separation between populations [Schliewen, Tautz, & Pääbo, 1994].

Computer models investigating plausible hypotheses for speciation in Cichlids [Turner & Burrows, 1995; Lande, Seehausen, & van Alphen, 2001], and sympatric (without physical barriers) speciation [e.g. Kondrashov & Kondrashov, 1999] have concentrated on the division of one species into two. In such models speciation can be observed by manual viewing of individual (modelled) fish, or by measuring emergent correlation between two fish properties (such as colour and female preference) using a simple measure such as Pearson's correlation coefficient [e.g. Kondrashov & Kondrashov, 1999] or by graphing explicit numerical characteristics of individuals [van Doorn & Weissing, 1999]. In such small-scale simulations, simple measures lead to an easily understood visualisation showing that speciation has occurred, and a simple trace of the process of speciation. These methods are not suitable for simulation of the evolution of large species flocks from single (or a few) ancestors as detailed information about the number, and timing, of speciation events is not revealed. This is unfortunate as the true mystery of Cichlid evolution is exactly how these species flocks arise.

A large scale agent based simulation has been built for the investigation of simulations of Cichlid speciation [Clement, *in prep*]. This systems includes environments that are less abstract than previous simulations, and which are designed to model the environment and characteristics of Lake Victoria rock-living Cichlids. The system allows very flexible creation of models, with agents used to model fish, and properties of the environment such as food sources. The shoreline of Lake Victoria, where rocky regions are separated by sandy, or muddy, regions where rock Cichlids are not found [Seehausen, 1996]. In the simulation system, this is modelled by multiple agent arenas (representing individual rocky reefs), with the (parameterised) possibility of fish migrating between these reefs. With a sufficient number of sufficiently different types of food sources, large numbers of species arise in the simulation. However, as the number of species rises, it quickly becomes impossible to manually extract objective traces of the number and histories of species. This is made more difficult by the generality of the simulation system itself. Fish can be designed using a number of different modelling methods, including the choice of directly modelling numerical phenotypes, or using genetic based models with loci, alleles, and genetic linkage. Hence any general method for tracing species must be independent of fish properties.

Our agent based simulation is most similar to agent based simulations used in Ecology [e.g. Ginot, Le Page, & Souissi, 2002] except that our aim is to understand Cichlid fish speciation, rather than the ecology of the lakes. However, it is extremely unlikely that speciation can be understood (or even

has much meaning) without a detailed understanding, and modelling, of the underlying ecology of the fish.

There is no general agreement in Biology as to what a species is. Popular definitions of species as being groups capable of interbreeding, but not being capable of interbreeding with organisms outside the group are not applicable to natural systems such as Cichlid flocks. Hence a large number of species concepts have been developed [e.g. Paterson, 1993; Futayama, 1998]. Different from observations of natural systems, simulations allow the reproductive history of a fish to be traced both forwards and backwards in time, for as many generations as the simulation is run. Hence, in this work we adopt a new species concept. Two fish of the same species are likely to have common descendants many generations on, and common ancestors many generations back. The research reported in this paper describes the use of Fuzzy methods for the tracking of the emergence and disappearance of species during simulations according to this species concept.

2. METHODS

The data visualisation [e.g. Fayyad *et al*, 2002] method described in this paper is run independently of the simulation program. A simulation is performed by building a model, which includes a definition of the number of agent arenas and then the assignment of agents (both fish, and environmental agents such as food sources) to arenas. Both arenas and agents typically have large numbers of parameters, including frequently building agents by selection of building blocks. The simulation is then run for a predefined time period, and a trace of the simulation is saved on disk. As well as other information, this simulation includes records of all fish born, and their parents. Hence the exact ancestral history of any fish is known.

In order to visualise speciation, we first need to establish the number, and membership of species in the end-state of the simulation. There is no explicit species marker, and hence species groups need to be discovered from implicit patterns in the final population. In biology, species concepts (descriptions of what is and is not a species) is a highly contentious issue, and there is no general agreement on how a species should be defined. In this work, we use two separate species concepts. First, fish from the same species are likely to share common ancestors over (relatively) recent times, while those in different species will only share ancestors at much earlier times. Secondly, if we observe fish breeding, then the two fish breeding are likely to be of the same species.

To obtain the number and memberships of final species groups a critical time period is chosen (by the user, typically about 2000 time steps – each step being roughly equivalent to a week) back from the end of the simulation. All fish from this critical time period which have surviving descendants are found. A binary vector is created for each surviving (at the end of simulation) fish, with a bit for each potential ancestor at the critical time period. This bit is set to 1 for potential ancestors which are actually an ancestor for some fish F , and 0 if the potential ancestor is not actually an ancestor. E.g. if we had eight potential ancestors at the critical time period, and a particular fish F was descended from potential ancestors 0, 2, and 7, then F 's vector is:

$$\text{vector}(F) = 10100001$$

These vectors are then clustered by Fuzzy C-Means Clustering (Bezdek, 1981). As the exact number of species is not known, clusterings are attempted from a (parameterised) minimum species number, up to a maximum species number. The first clustering where the sum of Euclidean distances between each vector and the set centres falls to less than 1.0 is taken as the correct speciation for this set of fish. The set of these (fuzzy set) final species is referred to as $Final = \{A, B, C, \dots\}$.

The next step is to discover the speciation history that lead to these species (e.g. A , B , and C) being present in the final steady state (which occurs in simulations, though is unlikely to happen in real life). To do this we first propose a set of potential ancestral species, which may have existed during speciation. If the set of initial species is S , then the set of potential species that may have occurred during speciation is the power set S^* . E.g. if $Final = \{A, B, C\}$, then $S^* = \{ \}, \{A\}, \{B\}, \{A, B\}, \{C\}, \{A, C\}, \{B, C\}, \{A, B, C\}$. A set such as $\{A, B\}$ represents a species that was the ancestor of final species A and B , but was not an ancestor of C . $\{ \}$ represents a species that was not an ancestor of any final species (i.e. “any other” species), and $\{B\}$ represents the final species B itself.

It is impossible to apply crisp species labels to fish in the process of speciation. At some point in the simulation, the species $\{A, B\}$ will exist, and at some later point, this species may be absent, and the species $\{A\}$, and $\{B\}$ will be present, but this is not an instantaneous event. Speciation is a process which takes time, and the aim of this research is to visualise this process. To track the history of species, we then define a method of calculating the (fuzzy) membership of each fish in each species, and then track the total sizes of these sets over time. This allows us to plot the history of species (represented by fuzzy sets) without having to assign crisp species labels to individual fish.

After fuzzy clustering, only fish alive at the end of the simulation have defined species membership. And, these are only memberships in the final species (e.g. $\{A\}$, $\{B\}$, and $\{C\}$), not the power set of potential ancestral species. Fish from earlier times are labelled by summing the membership weights of all their descendants, and then normalising these weights so that the largest such weight is 1.0). E.g. a fish F that has 27.7 A descendants, 35.7 B descendants, and 0.3 C descendants is given weights $w_A(F)=0.776$, $w_B(F)=1.0$, $w_C(F)=0.008$.

To find set memberships of all fish in all potential ancestral species, the following calculation is performed. This calculates the membership of one fish in one potential ancestral species.

$$\mu_{S_i^*}(F_j) = \prod_{Sp \in S_i^*} w_{Sp}(F_j) \times \prod_{Sp \notin S_i^*} w_{Sp}(F_j) \quad (1)$$

This calculation is based on the assumptions that the set of potential ancestors is an exhaustive set, and that all species are mutually exclusive.

Speciation is then observed by tracking two properties over time. First the total membership of each potential ancestral species is tracked over time. Secondly for each breeding where at least one descendant survived, a 'characteristic' species footprint of this breeding is generated by averaging the weights of the mother and father, and mapping this onto membership of the potential ancestral species using (1).

3. RESULTS

Results have been good in the sense that a reliable and repeatable (across different trials) method for tracking speciation has been created. The initial clustering into species is particularly reliable, usually resulting in clusters of fish which have identical ancestor sets at the critical time period, without ancestors being shared by any fish assigned to different species. The following two graphs show the visualisation of speciation in two different cases. Figure 1 shows speciation in a system with two different food sources sufficiently different to motivate speciation into two distinct species. Figure 2 shows speciation in the case of three sufficiently different food sources, and the emergence of three species.

In both cases, speciation appears more or less static for quite some time, before ancestral species disappear fairly suddenly. In the three species case, speciation was much faster than in the two species case. However, small changes to parameters of the model caused large differences in the speed, and

result of speciation. Hence, no conclusions can currently be made from this until far more is known about the factors that lead to speciation.

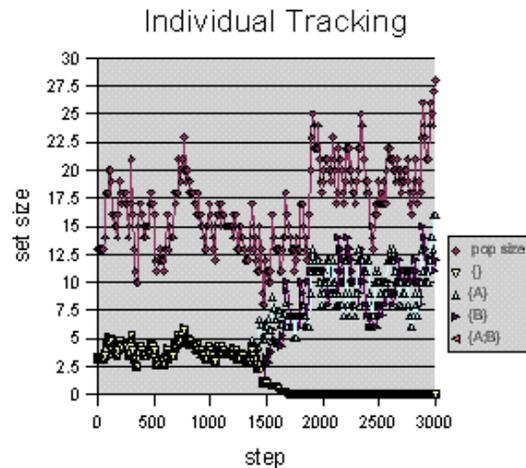


Figure 1a: Two-Species Individuals

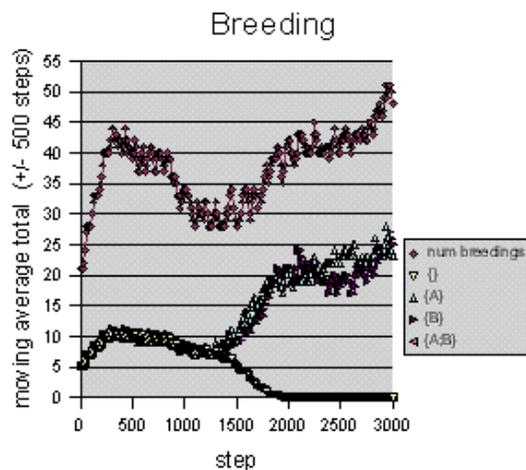


Figure 1b: Two-Species Breedings

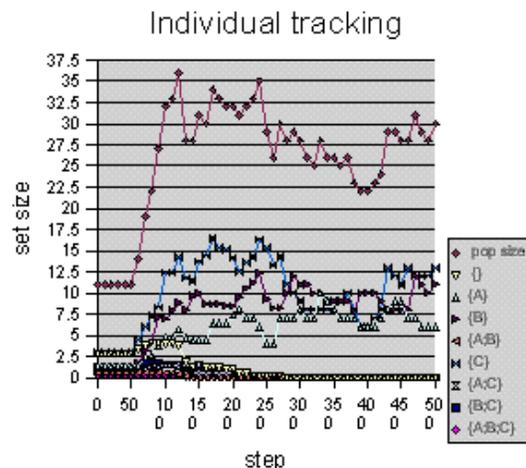


Figure 2a: Three-Species Individuals

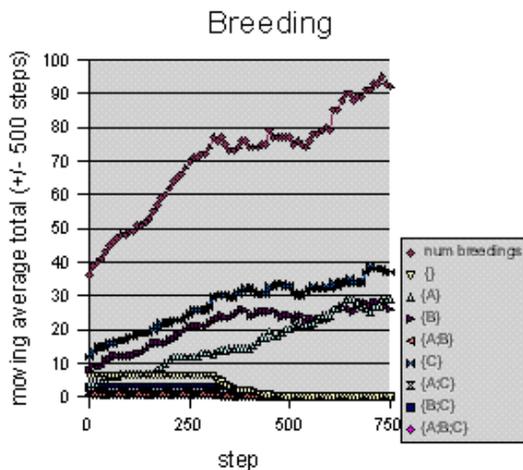


Figure 2b: Three-Species Breeding

4. CONCLUSIONS AND FUTURE WORK

The results show both clarity, and are reproduced across multiple independent trials. The initial fuzzy clustering of fish from the final surviving set performs far better than expected (and far better than several previous attempts to discover species). Typically, all fish from a single species share an identical set of ancestors (from the critical period), and no fish from different species share ancestors from this period. This is partially a result of designing systems that result in stable multi species populations, and the careful (and often experimental) choice of time spans for the simulation such that hybridisation between species had effectively ceased well before the critical period. However, we feel the results clearly indicate that the correct set, and number, of species is being found.

It is more difficult to evaluate the quality of the tracking of speciation over time, without an exact definition of species. However, tracking species both in terms of individuals, and mating events, give broadly comparable results. This supports the claim that the patterns being graphed are true representations of speciation, rather than aspects of speciation only applicable for a single species concept. Also, examination of the exact numerical traces of species (fuzzy set) membership, the exact point when an ancestor species finally disappears (total membership falls to zero) can be detected, allowing an objective measure of the time where speciation is complete.

It is planned to stop future work on this visualisation method and concentrate on using it to learn as much as possible about speciation in the circumstances where it can be used. Future developments in visualisation of speciation will be designed when there is a much better understanding of the modelling of speciation, and exactly what experiments need to be performed to learn more about the theoretical properties of various theories of speciation.

5. ACKNOWLEDGEMENTS

This work benefited greatly from discussions with a large number other researchers in both Computer Science, and Biology, including George Turner, Robert John, Peter Innocent, and Michael Walters.

6. REFERENCES

- Barlow, G.** 2000. *The Cichlid Fishes: Nature's Grand Experiment in Evolution*. Perseus.
- Bezdek, J.** 1981. *Pattern Recognition with Fuzzy Object Function Algorithms*. Plenum Press.
- van Doorn, G. & F. Weissing F.** 2001. "Ecological versus sexual selection models of sympatric speciation". *Selection* 2: 17-40
- Fayyad, U, Grinstein, G., & Wierse, A.** 2002. *Information Visualization in Data Mining and Knowledge Discovery*. Morgan Kaufmann.
- Futayama, D.** 1998. *Evolutionary Biology*. Sinaeur.
- Ginot V., Le Page C. & Souissi, S.** 2002. "A multi-agents architecture to enhance end-user individual-based modelling". *Ecological Modelling* 157: 23-41.
- Kondrashov, A. & Kondrashov, S.** 1999. "Interactions among quantitative traits in the course of sympatric speciation". *Nature* 400: 351-354.
- Lande, R., Seehausen, O., & van Alphen, J.** 2001. "Rapid sympatric speciation by sex reversal and sexual selection in Cichlid fish". *Genetica* 112/113: 435-443
- Paterson, H.** 1993. *Evolution and the Recognition Concept of Species*. John Hopkins University Press.
- Seehausen, O.** 2002. "Patterns in fish radiation are compatible with Pleistocene desiccation of Lake Victoria and 14,6000 year history for its Cichlid species flock". *Proc R. Soc. Lond. B. Biol. Sci.* 269: 491-7.

Seehausen, O. 1996. *Lake Victoria Rock Cichlids*. Verduijin Cichlids.

Schliewen, U, Tautz, d, Pääbo, S. 1994. "Sympatric speciation suggested by monophyly of crater lake Cichlids". *Nature* **368**:

Turner, G. & Burrows, M. T. 1995. "A model of sympatric speciation by sexual selection". *Proceedings of the Royal Society of London Series B Biological Sciences* **260**: 287-292.

7. BIOGRAPHY



Ross Clement received the degree of BSc in Cell Biology (1985), and MSc in Computer Science (1987) from the University of Auckland, New Zealand. He received the degree of Doctor of Engineering (1991) from The Toyohashi University of Technology, Japan. Since 1993, he has been first a Lecturer, then a Senior Lecturer in the Department of Artificial Intelligence and Interactive Multimedia of the Harrow School of Computer Science of the University of Westminster. His research interests include the Simulation of Cichlid Speciation and Evolution, and the application of Artificial Intelligence methods in Education.

QUALITATIVE-QUANTITATIVE ANALYSIS OF THE WATER FLOODING OF NATURE PARK “KOPACKI RIT”

ZELJKO JAGNJIC

ZDENKO TADIC

FRANJO JOVIC

*Faculty of Electrical Engineering
University of Osijek
Kneza Trpimira 2b
HR-31000 Osijek, Croatia
E-mail: zeljko.jagnjic@etfos.hr*

*Hidroing Osijek
Krizavicev trg 3
HR-31000 Osijek, Croatia
E-mail: hidroing@os.tel.hr*

*Faculty of Electrical Engineering
University of Osijek
Kneza Trpimira 2b
HR-31000 Osijek, Croatia
E-mail: franjo.jovic@etfos.hr*

Abstract: This paper presents the analysis of relationships between relative water levels on Drava and Danube river and water level inside the Nature park “Kopacki rit” during the summer 2002. Due to high complexity of complete system under study, usual quantitative techniques couldn’t be applied. Qualitative-quantitative analysis has set up some basic relationships and showed possibilities in dealing with complex water flooding systems. For the modelling procedure AI identification tool was used, based on circular qualitative-quantitative algebra.

keywords: process modelling, qualitative and qualitative reasoning, process identification, ecological system

1. INTRODUCTION

There are still a few places in Western and Central Europe that have not been completely changed through human manipulation and usage. One of them is Nature park “Kopacki rit” situated in Croatia, at the inlet of Drava river into Danube. The surface of the protected area is about 230 square kilometres. Nature park “Kopacki rit” represents a complex ecological system with 40 different habitats and more than 1600 animal and plant species. Due to almost completely preserve biodiversity, continuous monitoring and analysis are identified as necessarily steps in process preservation.

Due to high complexity of surface waters flow and undoubtedly high complexity of underground waters, it is impossible to establish a specific model that would suit the actual situation inside the Nature park “Kopacki rit”. Conventional analysis techniques, such as differential equations, aren’t suitable for such system investigation because of lack of parameters values and boundary conditions. On the other side qualitative reasoning aims to develop representation without precise information. If we possess quantitative information, it can be easily incorporated into qualitative mechanism, but even without it we can qualitatively analyse the system under study even if don’t know a complete system structure. However, due to incomplete specification, qualitative reasoning may generate a set of possible behaviours that still require an expert’s effort to interpret. Nature park “Kopacki rit” can be observed as a continuous-variable cyclic dynamic system with feedback loops and states, but very slow and inert. Process monitoring and interpretation don’t have to be performed in real-time. The structure of complex ecosystem of the

Nature park “Kopacki rit” was investigated in [Mihaljevic, 1999], and complete management strategy was proposed in [Jovic, et. al., 2001]. Those articles postulate that the basic force function for growth and disturbance in the Nature park “Kopacki rit” is flooding (i)regularity. Since the flooding (i)regularity in the park depends on Danube and Drava water levels, it was expected that these changes would affect water level inside the park. This analysis investigates mutual connection between high water levels of Danube and Drava river and water level inside Nature park “Kopacki rit”, by means of the qualitative-quantitative mechanism.

In the next two sections we deal with quantitative information used for identification of high tide wave and define inflexion points. These phenomena are directly identified through water levels. In the fourth section a qualitative-quantitative modelling method is used for data analysis. Relevant qualitative-quantitative models are presented. We conclude with discussion.

2. HIGH TIDE WAVE IDENTIFICATION

Data necessary for the analysis were obtained from measurement points on Drava and Danube river and on Lake Sakadaš, measurement point inside the Nature park “Kopacki rit”. Observed time period was between 10.08.2002. and 10.09.2002. Because the water level was measured each hour, there were 768 values obtained from each measurement point during these 32 days. Measurement points were: Batina and Vukovar at the Danube river; Osijek, Belisce and Donji Mihaljac at the Drava river and Lake Sakadaš inside Kopacki rit. Relative position

of all measurement points can be viewed in Fig.1. High tide wave input was at Batina and its influence on water levels on all other measurement points was observed during 32 days.



Figure 1: Relative position of all measurement points

Water levels at all observed measurement points can be viewed in Fig.3. Some key-spots should be noticed from Fig.3. Except from points where maximum water levels occurred, interesting are inflexion points, i.e. points at which water level become higher or lower than water level at other measurement point and at the same time water level is higher than 400 cm. Those spots are designated as 'a', 'b', 'c' on Fig.2. Also strong qualitative similarity between high tide shape between Batina, Kopacki rit, Vukovar and Osijek should be noticed.

From Fig.2. some general conclusion about water levels can be stated. There are time lags between maximum water level at Batina and all other measurement points. For example, maximum water level inside the Nature park "Kopacki rit" was reached about 25 hours after it reaches maximum value at Batina. Maximum water level at Vukovar was reached about 41 hours after it reaches maximum value at Batina. If we look again at Fig.2. connection between maximum water level at Batina and Vukovar is a natural consequence of river flow. But connection between maximum water level inside Nature park "Kopacki rit" and Batina should be further analysed. Also maximum water level at Belisce and Donji Miholjac can't be directly connected with maximum water level and high tide wave at Batina.

3. INFLEXION POINTS

In the previous section it was stated that inflexion points are points at which water level become higher or lower than water level at other measurement point and at the same time water level is higher than 400 cm. The first inflexion point, designated as 'a' on Fig.2. is on the **ascending** wave front and describes situation in which water level inside Nature park "Kopacki rit" becomes higher than water level at Vukovar. Characteristic values for that point are presented in Table 1. It is obvious that water level inside the Nature park "Kopacki rit" increases much faster than the water level at Vukovar. Water level at Batina was higher than 500 cm and inside the park water level was higher than 400 cm.

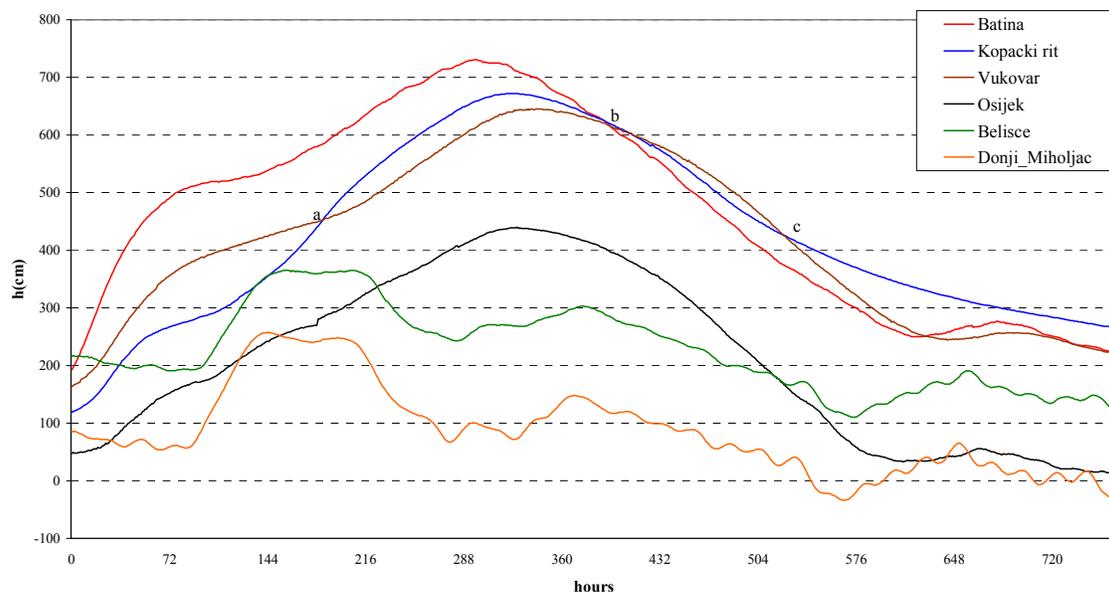


Figure 2: Water levels at measurement points

When Danube water level becomes higher than 400 cm at the entry of the park water from Danube starts to freely flow over into the park. Direct consequence is deceleration of water level increase at Vukovar. The park acts like a natural pool.

Table 1: Inflexion point: Kopacki rit-Vukovar

	Average water level (cm)	Average increase (cm)
Kopacki rit	438	2.66
Vukovar	449	0.87

The second inflexion point, designated as 'b' on Fig.2. is on the **descending** wave front after maximum water level was reached and describes actually three different situations. Characteristic values for each of these three situations are presented in Table 2, Table 3 and Table 4 respectively.

Table 2: Inflexion point: Batina-Kopacki rit

	Average water level (cm)	Average decrease (cm)
Batina	615	-1.71
Kopacki rit	619	-1.21

After maximum water level was reached at Batina, high tide wave front is moving downstream toward Vukovar. After the high tide wave front passes by the entry point for the park, draining of the park will start. Still, average decrease at Batina will be higher than average decrease inside the park, what will result in higher water level inside the park.

Table 3 represents situation in which relative water level at Vukovar becomes higher than relative water level at Batina.

Table 3: Inflexion point: Batina-Vukovar

	Average water level (cm)	Average decrease (cm)
Batina	615	-1.71
Vukovar	615	-0.94

Average decrease at Batina is higher than average decrease at Vukovar. Vukovar is downstream from the park, so draining of the park will affect water level at Vukovar.

Table 4 represents situation in which water level at Vukovar becomes higher than water level inside the park.

Table 4: Inflexion point: Vukovar-Kopacki rit

	Average water level (cm)	Average decrease (cm)
Kopacki rit	590	-1.46
Vukovar	592	-1.18

This situation is direct consequence of the park draining. Since the Danube water level at the park entry point is lower and lower, draining of the park is faster. It can be argument with average decrease,

which is higher than for the inflexion point Batina-Kopacki rit ($1.21 < 1.46$).

The third inflexion point designated as 'c' on Fig.2. is also on **descending** wave front and describes situation in which water level inside Nature park "Kopacki rit" becomes higher than water level at Vukovar. Similar situation was identified on ascending wave front (inflexion point 'a'). Characteristic values for that point are presented in Table 5.

Table 5: Inflexion point: Vukovar-Kopacki rit

	Average water level (cm)	Average decrease (cm)
Kopacki rit	435	-1.1
Vukovar	442	-2.1

Water level inside the park is very close to the value of 400 cm. When water level becomes lower than 400 cm, draining of the park almost stops. Entry point is higher than 400 cm, so there is no more direct water flowing out in Danube. High tide wave front passed Vukovar so average decrease at Vukovar is higher than average decrease inside the park. Result of that is higher water level inside the park than at Vukovar.

Analysis of the inflexion point was done with quantitative values, i.e. values of water levels. Though it seems very simple, it is far from trivial and it required an expert's assessment. Certainly it provided good basis for qualitative analysing, by setting up some initial relationship between measurement points. Also it should be noticed that direct influence between Drava water level and water level inside the park wasn't even taken into consideration.

4. IDENTIFICATION OF QUALITATIVE MODELS

Identification of the inflexion points has set up good basis for qualitative analysis. Expert's assumption about mutual connection between Drava and Danube water levels and flooding (i)regularity inside the park was confirmed. It can be postulated that the park is uncontrollably flooded at extreme high and uncontrollably dried at low water levels of the Drava and Danube river and semi-controllably flooded at medium water levels. Qualitative models were generated using AI identification tool, similar to neural networks but exhibiting algebraic explicit forms of the solution, based on the circular quantitative to qualitative information conversion [Jovic, 1997; Jagnjic, 2001]. Some basic characteristic of the quantitative-qualitative algebra incorporated into AI identification tool called Medusa2000 will be presented here. Detailed overview can be found in [Jovic, 1997].

- a) The primary data for the model are always taken from the set of quantitative sampled variables, i.e. discrete valued function or n-element vector.
- b) Quantitative values can be mapped into the qualitative space, called qualitative data series, by the ranking procedure $R\{V_i\} \rightarrow \{v_i\}$
- c) The qualitative distance between the two n-element qualitative vectors $\{v_i\}$ and $\{v_j\}$ is given as the sum of squared rank differences among the corresponding vectors points k.

$$d = \sum (\Delta_k)^2, \text{ where } \Delta_k = v_{ik} - v_{jk} \quad (1)$$

- d) The similarity between the two n-element qualitative vectors can be evaluated by using the qualitative correlation coefficient r, given by [Petz, 1985]:

$$r = \frac{\frac{n(n^2-1)}{6} - d - \sum A_1 - \sum A_2}{\sqrt{\left(\frac{n(n^2-1)}{6} - 2\sum A_1\right)\left(\frac{n(n^2-1)}{6} - 2\sum A_2\right)}} \quad (2)$$

where d is the distance defined earlier, and A1 and A2 represent the effect of the same rank repetition for the two respective variables. If z denotes the number of a single rank repetition, the value Ai for the variable $\{v_i\}$ is given as:

$$A_i = \frac{z(z^2-1)}{12} \quad (3)$$

The above features set the ground for the qualitative process modelling procedures based on the similarity calculus.

For the modelling purpose basic concept was proposed in the form of level dependent system behaviour, as presented in Fig.3.

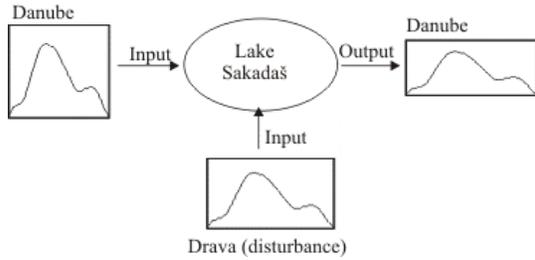


Figure 3: Basic concept

Previous quantitative analysis set up basic suppositions as follows:

1. All relative water levels are at the same altitude.
2. Time constant of ascending and descending waves are equal (for $h > 400$ cm).
3. Model correlations depend on time lags/lead.
4. Drava influence can be neglected – thus process disturbance can't be expected (for $h \gg$).

For the purpose of the modelling procedure, the basic set of the measurement values were divided into 32 sections, each with 24 measurement values. According to identified time lags and natural location of each measurement point, example of data necessary for the modelling procedure are presented in Table 6.

Kop0813 designates target function: water level inside the park during the 13.08.2002. All models that were generated for observed time period will not be presented here.

Table 6: Example of basic data set

	t0	t1	t2	...	t23
Bat0811	329	334	340		432
Bat0812	435	438	441		489
Bat0813	491	493	495		516
Os0811	62	65	64		111
Os0812	111	114	115		151
Os0813	152	154	155		172
Bel0811	204	204	204		195
Bel0812	195	197	197		191
Bel0813	191	191	191		197
DMih0811	72	72	72		69
DMih0812	70	71	72		58
DMih0813	59	60	61		86
Kop0813	267	268	269		285

Actually the whole situation that occurred during the observed time period was divided into three sections according to water level inside the park. First sections describes situation when water level inside the park was lower than 400 cm (the first 165 hours on ascending wave front and from 547 till 768 hours on descending wave front). The second section describes situation when water level inside the park was between 400 and 600 cm (from 166 till 254 hours on ascending wave front and from 414 till 546 hours on descending wave front). The third section describes situation when water level inside the park was higher than 600 cm (from 255 till 413 hours). For each section one general model was chosen. Target function for all models was water level inside the park. Those models are presented in Table 7.

Table 7: Qualitative-quantitative models

Hours	Water level (cm)	Model	Corr. coeff. r
0-165	less then 400	A	1
166-254	[400-600]	B	1
255-413	higher then 600	C	0.998
414-546	400-600	D	1
546-768	less then 400	E	0.81

A) $Batina * Osijek - 0.29 \frac{Osijek}{Belišće}$

B) $Batina * Osijek$

C) $Osijek + 0.12 \frac{Batina}{Belišće}$

D) $Batina$

E) $\frac{Osijek}{Belišće} + 0.2 \frac{Donji_Miholjac}{Osijek}$

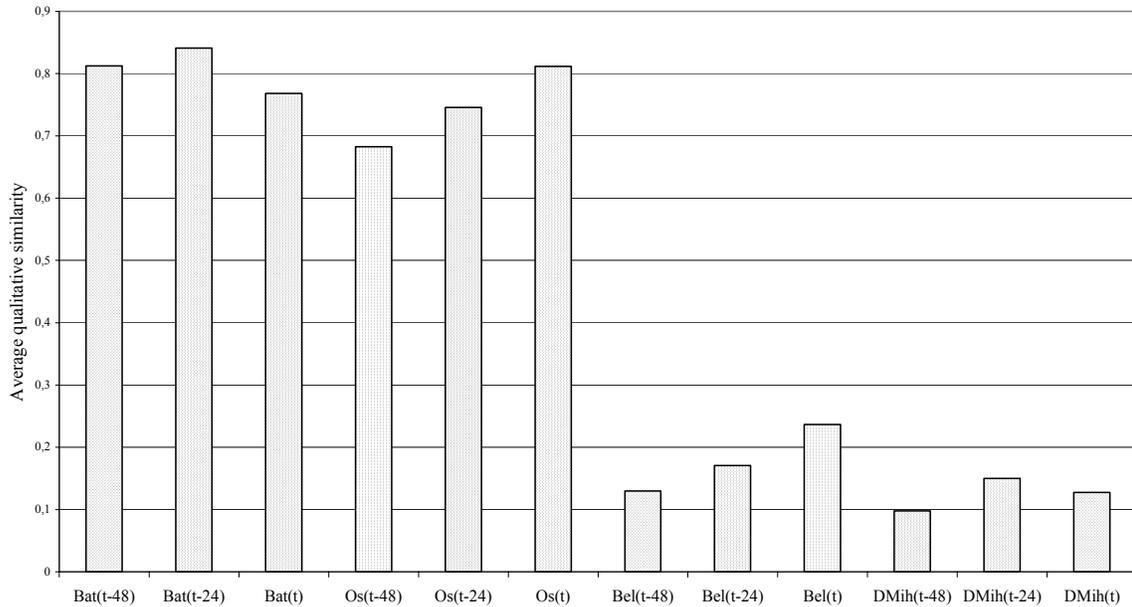


Figure 4: Qualitative similarity for Kopacki rit

All generated models shows high correlation coefficient with target function – water level inside the park. Also it should be noticed that the primary modelling variable in the case of ascending wave front is Batina. Model that is slightly different is model generated for descending wave front and water level below 400 cm. This model couldn't be explained and fully understood even by experts. Correlation coefficient was also rather low; only 0.81. But some expectations were confirmed. For medium water level (between 400-600 cm) on both ascending and descending wave front, similar model was generated. For the high water level, during the occurrence of maximum point, primary modelling variable is Osijek. It is explained with the fact that for the both maximum points, at Osijek and inside the park, approach to the maximum is much more smoother and uniform than for the Batina. Additional explanations of generated models were confirmed with qualitative similarity calculus. Fig.4 shows average qualitative similarity between Kopacki rit and all other measurement points. From Fig.4. it can be seen that the biggest qualitative similarity is between Batina with 24-hour time lag, Batina with 48-hour time lag and Osijek without time lag.

5. DISCUSSION

In this paper analysis of mutual connection between water levels on Drava and Danube river and water level inside the Nature park "Kopacki rit" was investigated. Although some previous analysis showed high determinacy of that connection [Jovic

and Mihaljevic, 1998], extreme high water levels during the summer 2002. year has offered new possibilities for exploration of new phenomenon. Such situations aren't usual so they demand to be analysed and investigated. Water levels on Drava and Danube measurement points were basic data set. The presence of the noise shouldn't be comment in detail. Because of that our investigation started very carefully, expressing only things that are actually natural consequences of water flow (identification of high tide wave). Identification of inflexion points helped us in setting up basic relationships between measurement points. Due to high complexity of the complete system, it wasn't possible to apply usual quantitative techniques for modelling procedures. Introduction of qualitative-quantitative modelling was right choice. Given models has confirmed experts expectation about mutual relationships. During the modelling it was found out that one measurement point is obviously missing. It is measurement point at the inlet of Drava into Danube river. This point was identified as a major point for complete analysis. But, even without her some basic relationships were set up. How does complete flooding period affect biodiversity inside park should be investigated in a due time. Since we mentioned that complete system is very slow and inert, first data about those changes will be accessible in future. Our task is to continue with monitoring and preserving this earth paradise.

REFERENCES:

- Jagnjic, Z., 2001, "Qualitative-quantitative process modelling by expansion and coding method", *Master thesis*, Faculty of Electrical Engineering and Computing, Zagreb
- Jovic, F., 1997, "Qualitative Reasoning and a Circular Information Processing Algebra", *Informatica*, 21, Pp. 31-47.
- Jovic, F., Mihaljevic, M., 1998, "A discrete water level model for water management in the 4D GIS concept of the Kopacki rit", In. Proc. of the Wydział Techniki Uniwersytetu Śląskiego, *Pretwarzeni i ochrona danych*, Katowice, Poland, Pp. 198-208.
- Jovic, F., Mihaljevic, M., Jagnjic, Z., Horvatic, J., 2001, "Management Strategy of the Periodically Flooded Nature Park "Kopacki rit" (Croatia)", in Proceedings of the *IFAC Workshop on Periodic Control Systems PSYCO 2001*, Cernobbio-Como, Italy, August 27-28.2001., Pp.51-55.
- Mihaljevic, M., 1999, "Kopacki rit – Pregled istraživanja i bibliografija Kopacki rit (Surveillance of Investigation and Bibliography)", Hrvatska akademija znanosti i umjetnosti, Zavod za znanstveni rad, Zagreb-Osijek.
- Petz, B. 1985. "Basic statistical methods", SNL Press, Zagreb, Croatia.



ZELJKO JAGNJIC was born in Osijek, Croatia and went to University of Zagreb, where he studied electrical engineering and computing. He obtained B.Sc.E.E. and M.Sc.C.S. from the same university in 1997 and 2001 respectively. From 1998 he is

working at Faculty of Electrical Engineering, University of Osijek, at Department for Computing at Laboratory for Artificial Intelligence as young researcher. His research interests encompass qualitative methods of modelling and development of fast algorithms for data analysis based on artificial intelligence methods and procedures for reasoning about complex systems. He is also interested in development of intelligence engines for games.

INTERACTIVE SPLINE MODELLING OF HUMAN ORGANS FOR SURGICAL SIMULATORS

R.J. TAIT^{1,2}, G. SCHAEFER¹, U. KÜHNAPFEL², H.K. ÇAKMAK²

¹*School of Computing and Mathematics, The Nottingham Trent University, U.K.*

²*Institut für Angewandte Informatik, Forschungszentrum Karlsruhe, Germany*

Abstract. This paper investigates a new method of interactive modelling to approximate position and shape of real patients' organs. The method is particularly successful in that it greatly reduces the time needed to create patient specific models. KisMo [1] is an interactive graphics based modelling software for creating 3D models with elastodynamic behaviour for use with VS-One - the Virtual Endoscopic Surgery Trainer (VEST) developed by the Institute for Applied Informatics (IAI) at the Forschungszentrum Karlsruhe. Software modules have been designed and implemented to enable interactive and intuitive modelling of closed spline curves on tomographic image data (CT, MR), which form a closed spline surface approximating position and shape of real patient organs.

1 INTRODUCTION

In the past CT and MRI volume data sets have been represented as a series of 2D slices. Volume rendering [2,3] is an alternative technique that increases the users ability to view the data. These different representations of the volume data greatly affect the user's ability to visualise hidden structures.

Most of the traditional methods of modelling rely upon edge following techniques to create contours that map a surface to the volume data. Taylor et al. [6] describe a method of positioning a set of geometrically defined 2D slice probes along an organ's centreline; the 2D slices of data represent cross-sections of the organ. The use of thresholding and level set methods [4] on the 2D slices provides a set of contours that, when interpolated, result in a finished model. An alternative technique by Miller and colleagues describes a method of extracting closed geometric models from volume data [5]. The geometrically deformed modelling technique employs a simple predefined object that is deformed based on a set of constraints. The initial stage of the modelling process involves either embedding into or surrounding the required organ with the predefined object. The predefined object then grows or shrinks to fit the organ within the volume.

Since the human body is mainly made up of a variety of organs, the medical consequence of organ modelling is very important, ranging from heart surgery to minimally-invasive surgery. Surgery planning for predicting the outcome of surgery or rehearsing complex operations requires an accurate representation of an organ. It is

therefore essential to improve the accuracy of the modelling process.

By using an interactive modelling process it will be possible to create more accurate patient specific organ models. Additional benefits include a reduction of time needed to create a specific model, with reduced user input being replaced by better use of body scan data. If successful the accurate representation of the physical human form, more specifically the realistic modelling of the internal organs, will not only improve current simulation systems but will also enlarge the applications of medical simulation resulting in improved performance and reduced training costs.

The rest of this paper is organised as follows: Section 2 describes the interactive spline modelling technique while Section 3 goes into some implementation details. Section 4 presents results and Section 5 concludes the paper.

2 INTERACTIVE SPLINE MODELLING OF HUMAN ORGANS

Before the completion of this work KisMo [1], the interactive graphics based modelling software for creating 3D models with elastodynamic behaviour (developed in C++), consisted of three predefined shapes: the flat, the pipe, and the ball, each of which is constructed from a spline surface. Volume rendering is used in KisMo as a modelling aid for the approximation of location and shape of organs inside of medical image data.

KisMo achieves a high level of flexibility by representing its predefined objects as individual spline patches. A total of four control points - each

with their own derivatives and interpolation points - are used to define a surface patch. The interpolation points can be described as representing a sub-patch. Like the control points the interpolation points can be interactively positioned so as to increase the level of accuracy possible.

The recognised technique of placing a vector inside of a vector has made possible the creation of a matrix structure. The matrix structure is used to hold control points and the control points in turn hold their own matrix of interpolation points. This has been made possible because although both matrices are derived from the same class and use the same member functions they are instantiated with different types. Therefore the matrix structure can be thought of as a container of containers. Specific functionality implemented in the matrix structure includes:

- Append and delete from the end of the structure.
- Insert and remove from anywhere in the structure.

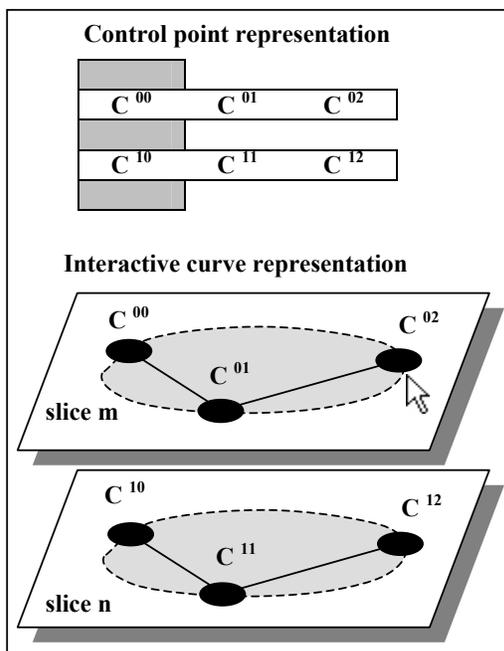


Figure 1 Model representations

Figure 1 shows the matrix and interactive curve representations of a model. When coupled with mouse interactions the additional functionality provided by the matrix enables the user to build a model interactively by placing control points directly at their required location thereby removing the need to morph a predefined object. A list

containing the slice numbers of a volume data-set at which accepted curves are positioned is also kept, this list is used to determine the position of the currently accepted construction curve as it is appended or inserted into the surface matrix.

To create an accurate model of an organ several series of connected control points that form curves can be placed on individual volume slices. The user constructs a curve by placing moveable control points on a slice in an ordered fashion to ensure that twisting of the model does not occur. To increase the accuracy of the modelling process moveable interpolation points are placed between control points. These interpolation points are also stored in a matrix structure and therefore have their own special cases. These special cases are similar to those experienced during the adding and deleting of control point in that they appear during the same operations: when the very first control points is the subject, when a control point between two existing points is the subject and when the end control point is the subject.

3 IMPLEMENTATION ISSUES

A construction curve is used to outline an organ in the volume dataset and appears as in red, the currently selected control point also appears red in colour. The construction curve allows the user to add, remove and drag control points as required. Control points are either appended or inserted depending on the position in the construction curve of the currently selected control point.

A user is free to move through the volume dataset using a volume rendering dialog box and can accept a complete curve on any slice that does not already contain a curve. A curve is either appended or inserted into the surface matrix depending on the position in the interactive surface of the current construction curve.

Figure 2 shows that a relatively accurate interactive surface can be constructed, after a short period of appending and inserting curves to the surface matrix. When appending or inserting curves to the surface matrix the new construction curve is always a copy of its neighbouring accepted curve and therefore appears in the same position. The position at which the accepted curve is to be inserted is determined by looping through the slice number list until a value greater than that of the slice number of the construction curve is found, the construction curve is then inserted before this position.

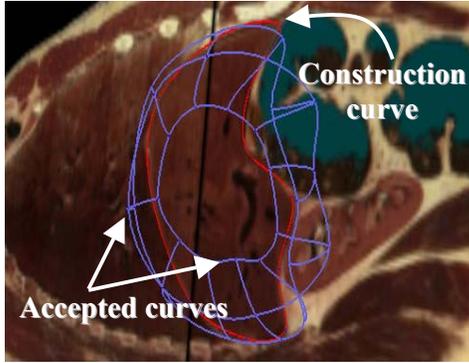


Figure 2 Several curves added to the interactive surface

A guessing algorithm has been implemented to minimise the work required by the user to drag control points of a newly appended or inserted curve into their required positions. A simple linear interpolation between two existing curves is used to guess the position of the control points in the construction curve. Clearly, the smaller the distance between accepted curves the more accurate a guess will be achieved. In most circumstances new curves are appended to the end of the interactive surface, in these cases the guess algorithm creates a copy of the outside curve to which the construction curve neighbours. In some situations the user may wish to finely adjust the positions of control points of an already accepted curve. Moving the construction curve to a slice containing an accepted curve that requires updating will cause the control points of the accepted curve to turn to green. Fine-tuning of the accepted control points is now possible.

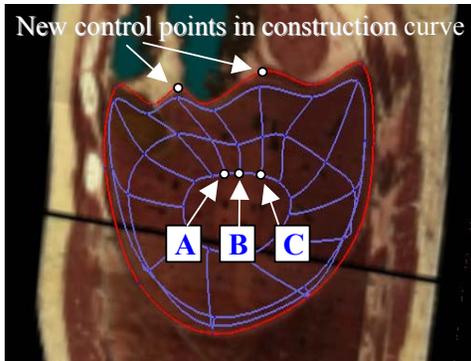


Figure 3 Adding control points through the interactive surface

Figure 3 shows that an example where a number of curves have been accepted and where in order to maintain an accurate interactive surface two extra control points have been inserted into the construction curve. To maintain a working data structure, two complete columns of control points (A) and (C) have also been inserted into the surface matrix. A control point that is appended (C) to the

construction curve has its position set at the mouse coordinates, all other control point positions in the appended column are set as

$$\forall i: C^i = B^i + (B^i - A^i)$$

Control point (B) that is inserted into the construction curve also has its position set at the mouse coordinates, all other control points positions in the inserted column are set as

$$\forall i: B^i = 0.5(A^i + C^i)$$

This method of appending or inserting means that control points are placed at positions that require the minimum amount of adjustment by the user in order to drag them into their correct positions.

Each control point has a number of interpolation points associated with it. The position of these interpolations is calculated from control and derivative point positions. It is possible for the user to change the number of interpolation points in a given section of the spline surface during the modelling process. The number of interpolation points of each model in the simulation scenario has to be adapted to the capabilities of the used hardware to guarantee real-time object deformation simulation.

Once the modelling process has been finished, the user has the option to finely adjust the finished model by repositioning individual interpolation points so that a more accurate representation of an organ's boundary can be made. Vectors that start at the control point's centre are used to define the derivative points. The derivatives can also be used to control the shape of the curve at a control point. Once activated the user can grab a derivative and change the shape of the curve at the host control point.

The modelling operations can be performed on slices in all three geometric viewing planes thus the user has the ability to choose a modelling orientation which best suits the model to be produced. This is desirable since certain orientations of a given organ are easier to model than others.

Many of the operations performed on the matrix representation of the model described in this paper - append, insert, delete and remove - are available in the Standard Template Library. However implementing them as member functions of a custom-built data structure enables additional functionality to be added at any time making the matrix structure a practical approach.

4 RESULTS

The new modelling technique was used to model the liver of the visible human as an elastodynamical object and test the object deformation with *VS-One*, the Virtual Endoscopic Surgery Trainer (VEST, shown in Figure 4) developed by the Institut für Angewandte Informatik (IAI) at the Forschungszentrum Karlsruhe in co-operation with Select-IT VEST Systems AG, Bremen-Germany.

The system provides an input box with force-feedback instruments and an endoscopic camera mock-up. The instrument and camera positions are tracked and due to the user's interactions in the virtual world, forces and torques are calculated and transmitted back to the input devices.

The surgical scene is specified in a hierarchical model-database in which the geometry of the objects, their elastodynamic behavior, and the kinematics of the instruments are defined. Deformable objects form the foundation of the surgical simulation. Their behavior is defined by physical characteristics like mass, stiffness and damping. Real-time collision algorithms enable the trainee to manipulate deformable objects using a physical instrument set and perform virtual interactions.



Figure 4 *VS-One* – The Virtual Endoscopic Surgery Training System

KISMET (Kinematik Simulation, Monitoring and Off-Line Programming Environment for Telerobotics) is a visualisation and simulation software developed at Forschungszentrum Karlsruhe. Because of its capabilities KISMET was found to be an ideal simulation platform for

computer aided surgery and is used in *VS-One*. It provides a rich set of virtual surgical instruments used for simulated grasping, clipping, cutting, suturing, tearing tissue, irrigation and suction.

The visible human dataset was used to approximate the correct location and shape of the liver. KisMo supports multiple image data formats but also DICOM, which is a standard image format produced by CT- and MR-scanners. This enables the use of any patient's body scan to create a simulation scenario with deformable objects for surgery training with individual patients' organs.

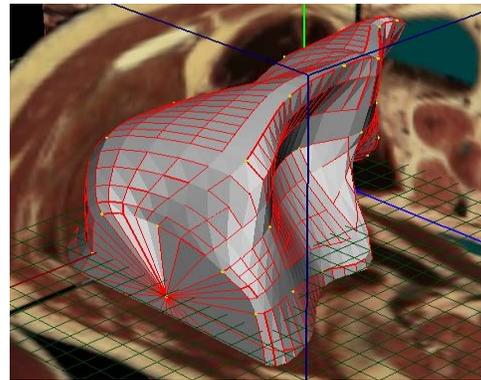


Figure 5 Model of the liver of the visible human in KisMo

Figure 5 shows the modelled liver in the visible human data set. KisMo automatically creates a mass-spring mesh for object deformation simulation for any modelled object. The deformation parameters can be set by the user and managed in a physical properties database.

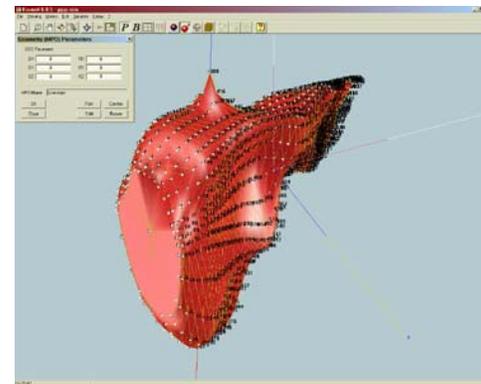


Figure 6 Deformation simulation in KISMET: Testing the physical parameters

Any created model can be stored in a model-database and re-used for creating new simulation scenarios. After exporting the model to the simulation software KISMET the deformation parameters can be tested. Figure 6 shows the modelled liver during the deformation simulation in KISMET. The mass-spring mesh is visible; a mass

knot is picked and elongated, which causes the object to deform.

Building a simulation scene needs modelling the organs and setting up correct physical parameters, materials and textures. Finally an inter-object connection has to be defined to enable interactions between the objects. After exporting the scenario to the *VS-One* system with the provided virtual surgical instruments the trainee can practice the operation. Figure 7 shows an example of a training session on virtual cholecystectomy.

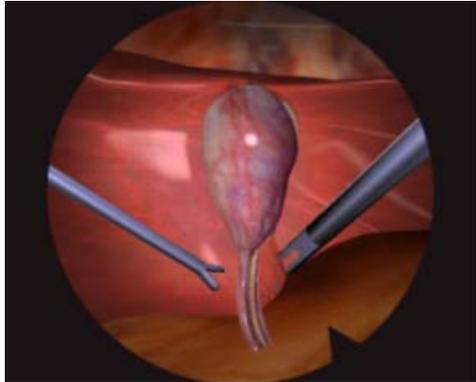


Figure 7 Realistic simulation scenario for cholecystectomy (gallbladder removal) in *VS-One*

5 CONCLUSIONS

This paper describes a method for accurate recovery of patient specific organ models for use in surgical simulation. The new modelling technique introduced produces significantly better and suitable models for surgical simulation than semi-automatic segmentation and reconstruction methods. It enables a high degree of interactivity and full control over the model generation. The modelling person, who should have medical background, can use his expert knowledge to identify structures and create accurate organ models.

Another advantage of this method is that a physical layer of the model is generated automatically during the modelling process, which enables deformation simulation, whereas segmentation and reconstruction methods only generate static 3D-models.

One benefit of our new modelling technique is the small amount of time needed to generate a basic model from a volume dataset. The full elastodynamical liver model in Figure 5 was created in less than 8 minutes by an expert user.

Future work should concentrate on better guess functions which also use image data information to speed up the modelling process and the quality of models.

6 REFERENCES

- [1] Çakmak, H., and Kühnapfel, U., 2000. Animation and Simulation Techniques for VR-Training Systems in Endoscopic Surgery. *Eurographics Workshop on Animation and Simulation 2000 (EGCAS 2000)*, 173-185, ISBN: 3-211-83549-0.
- [2] Drebin, R.A., Carpenter, L., and Hanrahan, P.: Volume Rendering. *Computer Graphics*, 22 (4), 65-74.
- [3] Levoy, M., 1988. Display of Surfaces from Volume Data. *IEEE Computer Graphics and Applications*, 8 (3), 29-37.
- [4] Museth, K., et al., 1999. Level Set Surface Editing Operators. Computer Science Department, California Institute of Technology.
- [5] Miller, J.V., et al., 1991. Geometrically Deformed Models: A Method for Extracting Closed Geometric Models From Volume Data. *Computer Graphics*, 25 (4), 217-226.
- [6] Taylor, C., Parker, D., and Wang, K., 2001. Image Based Geometric Modeling of the Human Aorta. Department of Mechanical Engineering, Stanford University California.

HIGH PERFORMANCE & PARALLEL SYSTEMS

DESIGNING A DISTRIBUTED JVM ON A CLUSTER

JOHN N ZIGMAN AND RAMESH S SANKARANARAYANA

*Department of Computer Science
The Australian National University
Canberra, ACT 0200, Australia
{john@cs.anu.edu.au, ramesh@cs.anu.edu.au}*

Abstract dJVM provides a distributed Java virtual Machine (JVM) on a cluster. It hides the distributed nature of the underlying machine from a Java application by presenting a single system image (SSI) to that application. dJVM is based on the Jikes RVM [Alpern et al, 1999] (a JVM written entirely in Java) and is the first distributed implementation of the Jikes RVM. This provides a framework for exploring a range of distributed runtime support algorithms on large clusters. Implementing this system using the Jikes RVM raises a number of issues that are addressed in this paper.

keywords: Cluster, Java, Java Virtual Machine, Single System Image.

1 INTRODUCTION

A significant number of server side applications are currently written in Java. The main advantage of Java programs is their portability, principally as a result of a clearly defined Java Virtual machine [Lindholm and Yellin, 1999]. In the past, the performance of Java programs have been much worse than that of corresponding C or C++ programs, resulting in the limited use of Java for writing applications that needed quick response times, like server applications. However, improvements in just-in-time (JIT) compilers have enabled Java programs to perform almost on par to similar C and C++ programs. This has resulted in Java being used to implement a significant proportion of server applications.

Server applications are typically multi-threaded, with limited interaction between threads servicing different clients. Scalability and performance are two important issues with such applications. Clusters of commodity hardware can provide a cheap solution to both of the above issues. However, to facilitate the use of such hardware without introducing additional programming complexity, it is necessary to provide an abstraction that efficiently uses the distributed nature of the hardware, while maintaining a unified view of the system. This allows a programmer to concentrate on the task of reducing the level of synchronization without the need to address issues of distribution.

There are many projects working on solving this problem. The approach taken by them to provide an SSI can be broadly divided into three categories:

1. *Provide an implementation above the JVM.*
This is typically implemented by transforming the Java program from the non-distributed form into a form that incorporates the bytecode to implement distribution. These transformations can be done either:
 - Statically—by transforming the Java classes prior to execution [Caromel and Vayessiere,

1998; Launay and Pazat, 1997; Objectspace; Philippsen and Zenger, 1997]

- Dynamically—by transforming the Java classes upon loading using a replacement class loader technique [Marquez et al, 2000].
- However, this is not completely hidden from the program because of Java's introspection facilities.
2. *Build the JVM on top of a cluster enabled infrastructure.* For example, a distributed shared memory [Ma et al, 1999; MacBeth et al, 1998; Yu and Cox, 1997]. While this presents a single system image of the cluster, it is incapable of taking advantage of the semantics of Java to improve efficiency and performance.
 3. *Build a cluster aware JVM.* This is the approach we have taken. The JVM presents an SSI to the application, but is itself aware of the cluster. This opens up possibilities for optimization based on the semantics of Java. As far as we know, there is only one other group [Adidor et al, 1999] that has taken a similar approach.

Hicks et. al. [1999] provide extensions to the Java language to support distributed applications. However, the programmer has to make use of these extensions to distribute the objects and hence this does not provide a true SSI.

Our cluster aware implementation of a Java Virtual Machine is dJVM, which stands for *distributed Java Virtual Machine*. It is based on the Jikes RVM Alpern et al [1999] and provides an SSI to Java applications. The target machine for the dJVM is a 96 node, 192 processor machine, Bunyip [Bunyip] running Linux. It has Fast Ethernet communication hardware using M-VIA [NERSC] and a Linux implementation of the VI Architecture [VIArch] to provide low software overhead on inter-node communication. This will provide a good platform for evaluating the scalability of dJVM and distributed runtime support algorithms.

The Jikes RVM is written entirely in Java and provides an extensible framework for distributed

virtual machines. There are two compilers in the Jikes RVM: the Baseline compiler and the Optimizing compiler. The Baseline compiler does not perform any analysis and translates Java bytecodes to a native equivalent. The optimizing compiler performs many aggressive optimizations. It can run on itself, producing competitive performance with production JVMs. This facility is leveraged to improve the performance of any extensions. In addition, it provides several facilities including those for escape analysis, data dependence analysis and synchronization graphs. These are used, with extensions where required, to assist in the analysis of programs for load distribution. The initial design of dJVM targets the Baseline compiler; further development will be on the Optimizing compiler.

As far as we are aware, this is the first distributed implementation of a JVM written entirely in Java. One of the big advantages of such a JVM is that transformation and optimization mechanisms developed can be used both on application programs and on the JVM itself. The Jikes RVM exposes additional features that enable manipulation of system classes. This allows us to do the following:

1. Reconfigure the core VM, as well as the application, for distribution (or any other purpose like persistence or optimization).
2. Regenerate already loaded code to improve functionality as more of the application is loaded into the system.

The first point above can only be partially exploited, and the second not at all, in a JVM that is not written in Java. In developing the dJVM, we have only made marginal modifications to the Jikes compilers. This allows us to use the optimizing compiler and all of its various features to their full potential. All of the code will be made available under CPL.

The first goal, that of achieving an SSI, has been met. We have developed a prototype version of the dJVM that runs on workstations connected via Ethernet, as well as on several nodes of the Bunyip cluster. We are now in the process of modifying the prototype to use the current release of the Jikes RVM. This will be followed by optimizations on the system to improve performance. In order to enable an SSI, the following were some of the important issues that had to be addressed:

Infrastructure—In order to construct a distributed VM, several infra-structural components needed to be altered. These include inter-node communication, the booting process and the use of system libraries.

VM Modifications—The VM handles the manipulation of remote data in addition to local data. In order to achieve this, class loading, method invocation and object access mechanisms had to be enhanced.

Object Allocation and Placement—The allocation and placement of objects is crucial to distribution.

Mechanisms to provide local and remote allocation of objects had to be put in place.

In this paper, we look at the above issues and provide broad outlines of our solutions. Implementation details are not dealt with in this paper. Section 2 deals with infrastructure, Section 3 with modifications to the Jikes RVM, Section 4 briefly discusses object allocation and placement and Section 5 outlines current status and future directions.

2 INFRASTRUCTURE

dJVM employs a *master-slave* architecture, in which one of the nodes acts as a *master* node and the rest are *slave* nodes. All global data is held at the master node, which also owns all of the classes in the system. Once the system is up and running, all of the class loading occurs initially at this node as well.

2.1 Building and Booting

The core of the Jikes RVM is used to generate the contents of the initial virtual machine image. This image incorporates a number of key classes essential to the functionality of the Jikes RVM including: class loading, type description structures, compiler(s), memory management and scheduling infrastructure. The build process utilizes the class loading, type and compilation systems to read and generate the essential components of the Jikes RVM image. Additionally, a number of classes, used to support remote objects and remote execution, are included in that image. Furthermore, not all runtime support classes become global; some of these remain local to a node. Consequently, only a subset of the core classes have transformations for distribution applied to them. The resulting image boots initially in a non-distributed context and later in a distributed context.

Booting loads the generated image into memory and executes a two phase boot process. The first completes the initialization of some of the runtime support structures for class loading, memory management, transforms and compilation that could not be incorporated into the boot image. The second phase sets up scheduling support for multi-threading, allowing the daemon and main threads to execute. This provides adequate support to enable the boot phase for the distributed virtual machine.

The dJVM must activate the communication layer to support distribution. The master node coordinates the connection between all the slave nodes, resulting in each node connecting to every other node in the cluster. Once the connections are established, the classes initially loaded are given a consistent identity across the system, thereby enabling the communication daemons to function correctly.

The communication system (see Section 2.2) is started prior to enabling the remote class loading. It initiates the message systems and threads for handling

requests. Consequently, it must be initiated after the scheduler is operational.

After the communication system is initiated, the local and remote class identities must be resolved (including any classes that are needed for this phase). Once this is done, all globally usable statics initialized at boot time must be coalesced. In general, it suffices to indicate that these values are now held by the master node.

Finally, remote class loading is enabled. From this point on, all class loading (see Section 3.1) on a slave node will interact with the master node. All class loading by the master will still be done locally.

The application thread is blocked until the communication processes and remote class loading are available, at which point the application thread can start executing the desired `main` method.

2.2 Communication

The communication mechanisms employed in dJVM provide a simple abstraction over the underlying interfaces. This abstraction is designed to satisfy an initial set of requirements: independent memories (and the management of those memories) and, initially, flexibility.

Highly targeted hand crafted solutions can provide the best performance at the cost of flexibility and development time. A more flexible system introduces impediments such as copying and allocation. Consequently, the initial communication system design is a compromise between our initial need for flexibility and our long term goal of performance.

To minimize the impact of internal communication design on the rest of the system, a communication manager interface is provided. It hides the underlying communication hardware being used, and the management of the resources associated with synchronous and asynchronous messages. It is responsible for initializing the system, bringing up a substrate (Section 2.2.1), a message registry (Section 2.2.2) and a pool of message processing threads (Section 2.2.3).

2.2.1 Substrate

Using an abstract object to provide an interface to the underlying communications allows different communications mechanisms to be plugged in. Two different implementations (or substrates) have been developed:

- TCP/IP—providing a simple and reliable implementation for our initial system, and
- MVIA—a lower overhead solution for local networks.

The role of both substrates is to provide startup and shutdown of connections, and to handle outgoing and incoming messages.

Startup in a reliable and static set of nodes is simple. A node is designated as the coordinator and all initial

connections are made with it. In turn, the coordinator informs all the nodes about all the other nodes. Each establishes the connections required with its peers. Once the connections are established, each node may send messages to and receive messages from any other node. Later development will include the case where nodes join and leave the set while it is running.

Outgoing messages are encoded into a buffer before being sent. A buffer is obtained from the substrate which maintains a pool of buffers, eliminating allocations requiring *garbage collection* (GC). This pool can be expanded if an inadequate number of buffers are currently available. Transmitted buffers are sent back to the pool to be reused. This buffer mechanism adds one level of copying to the transmission process. However, in the case of MVIA, it allows these buffers to be permanently locked in memory and used directly by the hardware.

Incoming messages are handled by message receiver threads. Each receiver thread blocks on a read from a connection. It processes the incoming packets and links them together to form a message; no additional copying is involved. The assembled message is then decoded (see Section 2.2.2) and executed, after which it waits for the next message.

2.2.2 Messaging Model

Flexibility plays a significant role in the initial system design.

To achieve this, a class tree of message types is developed, which allows for quick and easy extension. Such a system introduces some overhead in the form of additional method calls and translation costs.

Each message type is described by a message class, which extends and implements a common abstract class. Each message class encapsulates the code to *send*, *decode* and *process* a message. Thus, the message functionality is message type dependent and may in part be determined at send time as either:

- A *synchronous message*, requiring a response before processing continues (for obtaining data, locks and invoking methods), or
- An *asynchronous message*, which does not need to block the sender. An asynchronous message:
 - does not require a response (commonly used for GC messages), or
 - requires a response (may be used for some system load monitoring or other non time-critical information).

Send The send method first requests a buffer from the substrate and encodes itself into that buffer. A message requiring a response registers itself with the communication manager before sending it through the substrate to the target node, and waits for the response message to notify it.

Decode Upon receipt, the message type is determined and the appropriate decode method is invoked. The

decode method recovers the message from the buffer and does any initial processing where appropriate. A message that will only take a short time to execute may process itself and immediately generate a response (if required), e.g. getting a field of a remotely held object. A method that would potentially take a long time to execute (such as a method invocation) may grab an available message process thread for later processing.

Process This contains the code to perform the actual processing of the message. It may be invoked either by the decode method (for short duration operations) or by a handler thread.

2.2.3 Message Processing Threads Pool

A pool of threads waiting to handle incoming requests is managed by the communication manager. A message that only takes a short time to process may be handled immediately. Other messages will be handed over to a handler thread, thus freeing its message receiver thread to process other incoming messages. The handler thread will process the message and, on completion, place itself back on the queue waiting for the next message to process.

3 VM MODIFICATIONS

3.1 Class Loading and Resolution

The Jikes RVM maintains descriptions of types in the form of **VM_Class**, **VM_Field** and **VM_Method** objects. Loading a Java class generates a set of these objects to describe the type information of that class. In a distributed context this raises issues of acquisition and ownership of class information. Furthermore, the dynamic class loading mechanism of Java provides an opportunity to intercept the incorporation of classes and codes into the executing system. These issues are discussed in this section.

3.1.1 Distributed Class Loading

In a distributed system, it is necessary to have a commonly agreed to identification of classes, and in the dJVM we use a centralized class loader to achieve this. Centralized class loading has some advantages and disadvantages. It provides a simple single point of coordination, but does create a bottleneck. However, class loading becomes less common as the program executes, and consequently this is not seen as a performance priority in long running applications.

The class loading strategy employed must accommodate the dJVM boot process and normal running. Therefore, it has two phases:

Booting—an initial boot phase, prior to becoming a member of the cluster, in which classes must be loaded locally. Classes loaded locally must have their identity resolved with classes that are also present on the remote machines prior to activating centralized class loading.

Running—each class is loaded through a master node, ensuring a commonly agreed to identity for all newly loaded classes.

A set of objects are used to describe the type of a class, i.e. fields, methods and interfaces. A class type remains constant during the lifetime of the JVM. As such, each object describing this high level type information can be copied. Its identity is maintained by mapping each local copy to the same global identifier (UID).

In addition to replicating type information, for performance purposes, it is necessary to replicate literal values and static finals. This requires the class loading process to obtain these values and place them in the local VM's table of contents (JTOC). Furthermore, the local dictionaries used to maintain indexes to this data must also be updated.

Once loaded, a class can be instantiated. Instantiation compiles all static and virtual methods needed. Compilation can be done locally, generating code objects that are only visible within a node. The code generated is placed in arrays of type **INSTRUCTION** which can be directly executed. Each object has a TIB¹ (Type Information Block) as part of its header that describes some low level type specific information which includes a method table. In a homogeneous system, it is possible to replicate the TIB objects and the method code objects (this will be explored later).

The final phase is class initialization. This executes the static initialization code **<clinit>** for a class. In a JVM, class initialization happens only once. However, in the dJVM some of the runtime support structures are local to each node and must be initialized on each node where it is used. Thus, class initialization:

- for a runtime support class (specified by implementing **DVM_LocalOnlyStatic**), occurs once on each node that uses it, or
- for a globally used class, occurs once on the master node.

Recall that the runtime support classes are for the internal management purposes of the dJVM and not for use by the application.

3.1.2 Dynamic Class Loading

The dynamic class loading mechanism of Java allows the definition and use of user class loaders. Loading classes through this mechanism provides opportunities to modify class definitions and code [Marquez et al, 2000]. This provides a powerful tool which has been used to implement persistence and can be used to effect distribution. Mechanisms at the user class loader level suffer from two drawbacks:

- User class loaders are prevented from operating on system classes. Although this does not prevent wrapper classes from being used to redefine

¹ A TIB is an object used to describe object type information.

system class behaviour, it does impede the development of effective transformations as well as the efficiency of those transformations.

- Once a class has been loaded into a virtual machine, its signature and its place in the hierarchy becomes immutable.

As the Jikes RVM is written in Java, these two disadvantages disappear.

A small set of bytecode transformation tools [Zigman, 2002] integrated into the Jikes RVM are used to provide hooks for applying code transformations for effective distribution. The use of these tools minimizes the intrusiveness of the modifications to the compilation systems within the Jikes RVM by allowing transformations of class hierarchies, class signatures and method bytecodes including system classes. If necessary, these changes can be masked from the application code through a modified introspection mechanism.

3.2 Method Invocation

When a method is invoked on an object, there are two main issues that need to be addressed before the method can be executed. The first is to determine where the object containing the method resides. The second is to decide where the method should be executed. We deal with the first issue in Section 3.4. Here, we look at the latter issue, that of method execution.

Once we locate the home node of the object, we need to decide where to execute the invoked method. We can migrate the object to the node that invoked the method and execute it there. However, effective object placement locates objects at nodes based on execution pattern and hence, random movement of objects is to be avoided. Therefore, we will not pursue this approach and will use the following options based on context:

- Where the method is a static method, or is a method that does not access fields of its object, or accesses only *immutable* (that is, *read only*) fields of its object, the method is executed locally, since the method code is replicated and available locally. The immutable fields of the object are cached locally to reduce remote accesses.
- In all other cases, the method is executed on the node where the object is located, through the remote method invocation mechanism described in Section 2.2.

If an object is known to be immutable, then that object is replicated and cached locally. The LID to UID mapping of that object is changed to indicate that it is locally cached.

When we execute a method on a remote node, the execution context of the corresponding thread changes, along with the physical identity of the thread. However, the global identity of the thread must not change. In order to ensure this, each logical thread has

a unique global identifier and the mapping of that identifier to a local physical thread (**VM_Thread**) is changed at each node where it executes. This requires special handling and caching of application level threads, i.e. threads that extend **java.lang.Thread**.

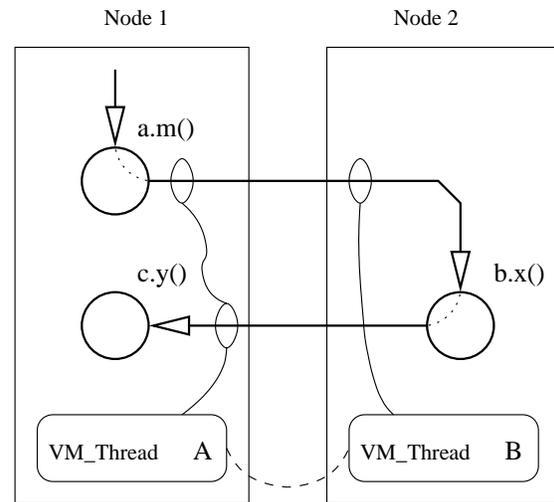


Figure 1: Local Thread Reuse

This raises the question of the reuse of local thread structures. Figure 1 depicts a thread, *Thread A*, instigated on *Node 1* that remotely calls a method **b.x()** on *Node 2*. The local thread structure, *Thread B*, is given the same global identity as *Thread A*. Method **b.x()** performs a remote call to method **c.y()** back on *Node 1*. Local *Thread A* is reused, continuing the processing of threads call chain. This is more efficient than allocating a new thread at *Node 1* that logically should have the same global identity as *Thread A*. Our design differs in this respect from that of Aridor et al [1999], where they create a new thread at *Node 1* in such a scenario.

Method calls are synchronous. Thus, a call made to another will block the instigating thread until that call is satisfied. However, as in the above example, if the same global thread calls a method on that node, then the blocking thread is interrupted and informed of the new incoming operation it is required to perform. Once that operation has been completed, it then waits for the completion of the remote method call.

To effect a remote invocation, a message encapsulating the identity of the thread and the method to be executed, along with the appropriate parameters, is generated. This message is passed to the target node, where a local thread resource is assigned (if one already hasn't), the parameters placed on the stack and the appropriate method invoked. This mechanism can be implemented either by:

Inline code modification—Code is constructed at compile time to retrieve the parameters from the stack (converting each reference from a local to a global representation), packing them into a message, which is sent as a request to the target node. Upon receipt of an invocation request, the

parameters must be unpacked onto the stack and the method invoked. However, such a method can pollute the instruction cache of the initiating site.

Proxy methods—Each method that is compiled has two additional methods generated, *proxy* and *stub*. The first packs and sends a request and the second unpacks and invokes the method. This requires determining whether an object is local or remote. Aridor et. al. [2001] state that “We cannot make this determination by using different classes for the master and proxy, or by adding a field to the application, as the introspection APIs would make this visible to the application, violating SSI”. However, this can be circumvented by modifying the introspection mechanism in the Jikes RVM runtime libraries, so that the mutations in the class definitions are hidden from the application.

Clearly, the second solution is cleaner and more flexible. For expediency, in the initial prototype we used the first solution, but have moved to the second solution with the port of Jikes RVM 2.2.0.

Exceptions and interrupts must also be accounted for. A thread that handles an incoming request must catch all possible exceptions from the application code. Once an exception is caught, it must package that exception and return it to the node that initiated the call, where it will be re-raised. By contrast, an interrupt must be propagated along the call chain to the node currently executing the global thread, where it is finally dispatched to the underlying **VM_Thread**.

3.3 Data Access

In broad terms, there are two areas that need to be considered: globally referenced data (i.e. static variables) and instance data (i.e. objects and arrays).

3.3.1 Globally Referenced Data

Each node must manage its local resources. The local resource management information in a non-distributed JVM is global data. However, in a distributed JVM, most of that information is global only within the context of that node. Hence, the set of static variables in a distributed JVM can be divided into two mutually disjoint sets, namely:

- the set of static variables that are global to all the nodes, and
- the set of static variables that are global within a specific node.

One way to implement the above is to encapsulate the node specific information in an object that is visible only within the specific node. Another is to change the code that accesses static data according to the data that is being accessed. We take the second approach.

A runtime test is necessary to determine if a static variable is held locally or remotely. The Jikes RVM (and its runtime structures) are written in Java which raises the following two issues:

1. Determine whether or not the static variable under consideration is just a local runtime support variable.
2. If not, check whether it is of the type that may be held locally or remotely. If so, test where it is actually located by using a local runtime structure. If care is not taken, then the code generated to test whether a static variable is locally or remotely held may itself require a similar test, resulting in an infinite loop.

In the Jikes RVM, the static fields are held in a Java Table of Contents (JTOC). Associated with each field is a descriptor that identifies the category and type of information held, e.g. literal **int**, static field **long**. The set of descriptors is an array of bytes, held as a static array in **VM_Statics**, and is referenced from the JTOC. Each descriptor has two unused bits and we use these to indicate whether it is a read only field and/or a remotely held field.

In general, the runtime support classes in the Jikes RVM only contain static variables that are used locally within a node. This is communicated to the compiler through a simple annotation method commonly used in Java—an empty interface **DVM_LocalOnlyStatic** implemented by any class that contains static data that is always accessed locally. The code generated to access a static field of such a class is unchanged from that of the original compiler. For other classes, the test described above is performed. Clearly, this does not introduce any overhead for locally used static variables. However, it does introduce some overhead for other static variables. The overhead will be reduced in later implementations by combining the descriptor and JTOC information into the one array.

3.3.2 Instance Data

The approach taken to implementing the reference faulting mechanism, outlined in Section 3.4, dictates the compiler changes necessary for handling instance data. In particular, there are four different types of code to consider: object and array access, code execution, lock operations and type checks.

Field access (**getfield** and **putfield**) and array element access (**aaload**, **aastore** etc.) dereference an object² to determine the memory location of the data. The software detection of remote references mentioned in Section 3.4 necessitates a test of the reference itself. A local reference is accessed in the normal manner, whereas a remote access is initiated by calling a static method, which generates a message that contains a description of the remote operation (see Section 2.2).

² An array element access is considered to be a variant of field access.

3.3.3 Type Operations

The remaining operations are type checking operations. Explicit type checking operations **checkcast** and **instanceof** can interrogate the types through remote calls. We cache this information locally, since an object's type remains unchanged during its life time. Any interrogation of an object to obtain its TIB is intercepted to obtain the TIB from the local cache.

3.4 Object Location

We use a reference faulting mechanism to determine whether an object is available locally, or is only available remotely. This is achieved by using an appropriate global and local addressing scheme for objects. Each object has an associated *universal identifier* or *UID* that uniquely identifies the object in the cluster. The UID needs to be resolved into an object address at a specific node. One of the ways in which the UID can be allocated is *centrally*, where the allocation of UIDs is done by a master node in the cluster. However, this could lead to a bottleneck at that node. We have chosen to use a *decentralized* approach, where each node in the cluster allocates a UID for an object that it owns, from a range of UIDs under its control. A UID is generated when an object reference is exported for the first time. The node that owns an object is called the *home node* of that object. While this eliminates the above mentioned bottleneck problem, it does lead to more complicated updates resulting from object movement from one node to another.

At any given node in the cluster, an object reference either points to a local object or to a remote object. In our implementation, a local object has an associated *object identifier* or *OID*. This is identical to the address of the object at that node and thus avoids any overheads incurred through indirection tables or indexes. A remote object has an associated *local logical identifier* or *LID* at that node. This LID needs to be mapped to the UID of the object to determine its exact location in the cluster.

In the Jikes RVM, all object and array addresses are 4 byte aligned. We use this property to make the reference faulting mechanism work. All the LIDs are misaligned, while the OIDs, being actual object addresses, are not. This can be implemented by either:

Software—Misaligned addresses can be detected by examining the LSBs (least significant bits) of an address and branching if not zero. This introduces a couple of instructions into the instruction pipeline. Importantly, no indirection or additional loads are required.

Hardware—In the Jikes RVM, checking array bounds and object types are 4 byte aligned operations, and their interrogation via a misaligned address will cause a hardware trap. Most accesses will be to local objects, so the added expense of a hardware

trap for remote objects will be outweighed by zero overhead for local access.

We have currently implemented this using software, but intend to implement the hardware faulting mechanism.

3.5 Locking

Locking operations are directed to the home node of the object, and in the case of locks on classes they are directed to the home node of the class (the master node). The thread identifier sent with the lock is the global identifier of the thread, for obvious reasons. A thread that already has a lock can acquire additional locks on the same object. The number of locks and unlocks must be equal. For efficiency, additional requests need not be sent to the home node. A local count can be kept, and an unlock request sent to the home node once the count reaches zero.

The explicit lock operations **monitorenter** and **monitorexit** are handled by the Jikes RVM runtime system and do not need compiler modifications. Implicit locks on object instances (**synchronized** methods) are similarly handled. However, implicit locks on statics must be directed to the home node of the class. In the case of classes that implement **DVM_LocalOnlyStatic**, this is done locally. In all other cases, it is done by the master node.

4 OBJECT ALLOCATION AND PLACEMENT

In order to enable the distribution of objects across the nodes in the cluster, there should be a way of remotely allocating an object on a specified node. In the Jikes RVM, the **VM_Allocator** class does the work of object allocation. In dJVM, this is replaced with an allocator that directs requests to a standard local allocator or to an allocator on another node. On initialization, each node will have an instance of an allocator that directs allocation requests to the local node. Each allocator instance acts as a placeholder, enabling the remote invocation mechanism to be used to effect remote allocation requests to specific nodes. The UID of this placeholder object is known to all the other nodes. A remote allocation request is passed on to the placeholder object of the node at which the allocation is to be made, which then allocates the object locally at that node.

Introducing a local or remote allocation decision process at runtime can be expensive. Compile time analysis can eliminate some of these decisions by generating local allocation code where it is clearly sensible to do so. Object placement is important for load balancing and performance improvements. Ideally, a new object should be placed on the node where it is most required. Aridor et. al. [1999; 2001] enumerate a number of techniques and patterns used to improve efficiency and these will be incorporated into the dJVM. Additionally, we will examine further techniques using escape analysis, call chain, and static

and dynamic profiling information to enhance object placement.

5 CONCLUSIONS AND FUTURE WORK

In this paper, we present some of the design issues that we came across in developing dJVM. We also outline some solutions to these issues. Currently, we have a working prototype that uses the baseline compiler. We are working on modifying this prototype to use the latest version of the Jikes RVM. Once this is done, we will look at the following issues:

Optimizing Compiler—The optimizing compiler implements a range of analysis and optimization techniques. These optimization techniques can be applied to the Jikes RVM (and hence our extensions) as well as the application code. Consequently, we intend to use facilities such as escape analysis and profiling, to feed into the object placement and migration decision making processes.

Proxy/Stub—For code execution a proxy/stub mechanism provides a cleaner implementation of code invocation. This will be mixed with the reference faulting scheme, to provide minimum overhead for field and array accesses, while providing a clean implementation for remote calls.

Communication—Flattening the communication hierarchy and removing all but essential object creation and data copying.

From our experience with building the dJVM, we feel that the JVM specification should allow for a system class loader facility that minimizes the set of system classes that cannot be modified. Although, such a mechanism does raise significant security issues.

Concurrently with the development of dJVM using the optimizing compiler, we will investigate techniques to improve performance. We intend to use techniques such as code analysis, and static and dynamic profiling, for determining object placement and migration, object caching and thread migration. We also intend to implement efficient distributed garbage collection algorithms.

6 ACKNOWLEDGEMENTS

The dJVM project is funded under the ANU-Fujitsu CAP Research program.

7 REFERENCES

Bowen Alpern, Anthony Cocchi, Derek Lieber, Mark Mergen and Vivek Sarkar. Jalapeño—a Compiler-supported “Java Virtual Machine for Servers”. In Workshop on Java for High-Performance Computing (with ICS99), Rhodes Greece, June 1999.

Yariv Aridor, Michael Factor and Avi Teperman. “CJVM: a Cluster Aware JVM”. In First Annual Workshop on Java for High-Performance Computing (with ICS99), Rhodes Greece, June 1999.

Yariv Aridor, Michael Factor and Avi Teperman. Implementing Java on Clusters. In Euro-Par 2001, LCNS 2150, Rhodes Greece, 2001, Springer-Verlag, Pp722-732.

D Caromel and J Vayssiere. “A Java framework for seamless sequential, multi-threaded, and distributed programming”. In ACM 1988 Workshop on Java for High-Performance Network Computing, INRIA Sophia Antipolis, Greece, 1998.

DCS. Bunyip (Beowulf) Project. <http://tux.anu.edu.au/Projects/Bunyip/>.

Michael Hicks, Suresh Jagannathan, Richard Kelsey, Jonathon T Moore and Cristian Ungureanu. “Transparent Communication for Distributed Objects in Java”. In ACM 1999 Conference on Java Grande, San Francisco, California, USA, 1999, ACM Press.

P Launnay and J Pazat. “A framework for parallel programming in Java”. EUT Report 1154, IRISA, December 1997.

T Lindholm and F Yellin. “The Java Virtual Machine Specification”. 2nd Ed, 1999.

M J M Ma, F C M Lau, C L Wang and Z Xu. “JESSICA: Java-Enabled Single System Image Computing Architecture”. In Ronald Morrison, Mick Jordan and Malcom Atkinson, editors, International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA99), Las Vegas, July 1999.

M MacBeth, K McGuigan and P Hatcher. “Executing Java threads in parallel in a distributed-memory environment”. In IBM Centre for Advanced Studies Conference, Canada, November 1998.

Alonso Marquez, John N Zigman and Stephen M Blackburn. “Fast portable orthogonally persistent Java”. *Software: Practice and Experience*, 30(4):449-479, April 2000.

NERSC. M-VIA. <http://www.nersc.gov/research/FTG/via/>.

Objectspace. Voyager. <http://www.objectspace.com/products/voyager/>.

Michael Philippsen and Matthias Zenger. “JavaParty—Transparent Remote Objects in Java”. *Concurrency: Practice and Experience*, 9(11):1225-1242, November 1997.

VIArch. VIArch. <http://www.viarch.org/>.

Weimin Yu and Alan L Cox. “Java/DSM: A Platform for Heterogeneous Computing”. *Concurrency: Practice and Experience*, 9(11):1213-1224, November 1997.

John N Zigman. “Bytecode Transformation Tools for Jikes RVM”. <http://www.wastegate.org/systems/>, 2002.

LOAD BALANCING BY DOMAIN DECOMPOSITION: THE BOUNDED NEIGHBORS APPROACH

F.BAIARDI A.BONOTTI L.FERRUCCI L.RICCI P.MORI

*Dipartimento di Informatica, Università di Pisa
via F.Buonarroti, 56125-Pisa (Italy)
baiardi,mori,ricci@di.unipi.it*

Abstract: This paper presents a new domain decomposition approach whose main goal is the computation of a load balancing partition while reducing the overhead to compute such a partition. In the proposed approach, the number of neighbours of each sub-domain returned by the decomposition can be bounded by an user supplied value. This reduces the communication overhead of the application. We describe an algorithm implementing our decomposition strategy and apply our approach to WaTOR, a classical dynamical simulation problem. We report also some preliminaries result to prove the effectiveness of our approach.

keywords: Partitioning, Mapping, Load Balancing, Cluster Computing

1 INTRODUCTION

Domain Decomposition is a technique exploited both to balance the load and to reduce communications in parallel applications [Culler and Singh, 1998]. It is applied when the data set of the application can be regarded as a *physical domain* where the computation performed on each element D requires the knowledge of a small subset of data close to D only. Several parallel applications belong to this class, a classical example being that of cellular automata [Sloot and Talia, 1999].

This technique partitions the domain into a set of sub-domains with the same computational load and assigns each sub-domain to a distinct process. Each process updates the elements belonging to its sub-domain S and communicates with the other processes only to update the elements located on the boundary of S .

The domain decomposition problem becomes challenging when *dynamical applications* are considered because in these applications the decomposition of the domain has to be updated during the computation. The update is required to take into account that the number of the elements of the domain and/or their positions are dynamic, i.e. change during the computation.

Several approaches have been proposed in the last years. Each strategy takes into account the trade-off between the overhead introduced by the dynamic partitioning of the domain and the benefits obtained by balancing the load. *Orthogonal Recursive Bisection* [Salmon, 1990] is a domain decomposition strategy exploited in many parallel application. This technique

produces an optimal balance of the work, at the expense of a large computational complexity. On the other side, simpler solutions often result in decompositions of the domain characterized by an unsatisfactory load balance.

This work describes the *BoundedNeighbours* approach, a domain decomposition technique whose main goal is to reduce the overhead introduced by partitioning the domain while preserving an acceptable balance of the work. Another interesting feature of our approach is that it allows the user to *bound* the number of neighbours of each sub-domain. Since each sub-domain is assigned to a different process, this bounds also the number of communications of each process and, hence, the overall communication overhead.

Section 2 reviews existing proposals. The main features of our strategy are described in section 3, while section 4 presents the *Bounded Neighbours* implementation. Finally, section 5 shows the application of *Bounded Neighbours* to *WaTOR*, a classical irregular distributed simulation problem. Some significant performance results are described as well.

2 RELATED WORKS

Several domain partitioning techniques have been proposed in the last years. While most of them consider 2-dimensional domains, almost all of them can be easily extended to cover a larger number of dimensions.

Applications defined on irregular and/or dynamical domains generally require a dynamic partitioning of

the domain. Nevertheless, some dynamical applications exploit *scattered decomposition* [Saltz, 1990], a static decomposition technique. Scattered decomposition partitions the domain into a set of rectangular zones, the *templates*. Each template is further divided into a set of rectangular regions, the *granules*. Corresponding granules belonging to different templates are assigned to the same process. The resulting load is balanced only when the domain is characterized by a uniform distribution of the load to the granules. The main advantage of this technique is that the decomposition defines a set of regular communication patterns. On the other way, a satisfactory load balance may be obtained only when the size of the granules is rather small.

Since the communication overhead due to a granule increases with the ratio between the perimeter and the area of the granule, a larger number of granules improves load balancing at the expense of increasing the ratio between the communication overhead and the computational one.

Dynamic decomposition techniques update the domain partition when the number and/or the position of the elements are modified. A well known approach is that of [Salmon, 1990; Simon, 1994], the *Orthogonal Recursive Bisection (ORB)*. *ORB* initially splits the domain into two rectangular sub spaces with the same load. The set of processes is partitioned into two subsets as well, and each subspace of the domain is assigned to a subset of processes. The procedure is recursively applied until a single subspace is assigned to each process. In general, *ORB* achieves a good balance, but its computational cost is high because of the complexity of determining the cuts of the domain. Furthermore, each process records the partitions through a binary tree, built during the load balancing phase. This tree is visited during the computation to detect the neighbours of each process. This visit introduces a further overhead in the computation. Note that, in the worst case, an high number of neighbours may result for each process. *ORB* was originally proposed for an hypercube architecture and its implementation is greatly simplified if the number of processes of the application equals a power of 2.

A simpler approach considers the domain as a grid that is partitioned into blocks of contiguous rows. The boundaries of each block are dynamically re computed to balance the load. The main advantage of this technique is its simplicity. Furthermore, each process has two statically defined neighbours. The main disadvantage is that the balancing is not satisfactory, because of the coarse grain of the partitioning.

The *cost zone* [Singh et al, 1995] or *space filling curves* [Singh et al, 1995; Baden and Pilkington, 1995; Moon et al, 2001] are exploited for applications

defining a hierarchical subdivision of the domain .

3 BOUNDING NEIGHBORS

Bounded Neighbours is a domain decomposition strategy whose main goal is to *reduce the overhead* of dynamical domain partitioning while producing an *acceptable load balance*. Furthermore, *Bounded Neighbours* allows the user to *bound* the number *NS* of neighbours of each sub-domain. This implies a reduction of the overhead due to communications. As a matter of fact, the process associated with a sub-domain requires elements belonging to its neighbours when it updates the elements on the border of its partition only. In our approach, the number of processes exchanging data with each process of the application is bounded by *NS*. Since each communication with a distinct partner implies a new *start-up phase*, this reduces the communication overhead. It is worth noticing that the computational cost of the start-up phase of each communication is high, in particular when considering applications developed on *workstation clusters*. Furthermore, several optimizations can be applied to reduce the communication cost between a single pair of partners.

In *Bounded Neighbours* the value of the parameter *NS* can be defined by the user. Each decomposition produced by *Bounded Neighbours* satisfies the *bounded neighbours condition*, i.e. the number of neighbours of each sub-domain does not exceed *NS*.

Bounded Neighbours generalizes the simple domain decomposition that assigns blocks of consecutive rows of the grid to each process. In our approach, each row can be further subdivided into segments and each segment can be assigned to a different sub-domain. The leftmost part of Figure 1 shows a decomposition produced by *Bounded Neighbours*. This strategy returns an optimal load balancing, but, in general, the bounded neighbours condition is not satisfied. This constraint is considered in a second phase, when the cuts produced by the first one are shifted to produce a legal decomposition. *Bounded Neighbours* defines a set of simple conditions which imply the bounded neighbours one. Consider, for instance, a $n \times m$ grid and suppose $NS = 2$, i.e. the number of neighbour of each domain is bounded by 2. In this case, the following conditions guarantees that the bounded neighbours condition is satisfied.

- each row of the grid includes at most one cut;
- each sub-domain includes at least m points of the grid

These conditions can be easily checked by considering the grid decomposition. For instance, in Figure 1, process P3 includes exactly m points of the grid.

[Bonotti, 2002] defines similar conditions for the more general case. It is worth noticing that the number of cuts that can be applied to each row increases with the value of NS . Furthermore, the balancing is improved by a larger number of cuts. Nevertheless, our experiments show that an acceptable compromise between communication overhead and load balancing may be achieved by low values of NS .

Fig. 1 compares our approach with blocks of rows decomposition (shown in the central part of the figure) and with *orthogonal recursive bisection* (shown in the right part). Our approach produces a better load balancing with respect to the first one because a row can be cut and the resulting subset of the row can be assigned to different processes. In the block of row decomposition, the number of neighbours of each process is equal to 2. This can be obtained also in our approach, by setting NS to 2.

In general, the *ORB strategy*, achieves a better load balancing. On the other hand, in the worst case, it may result in a large number of neighbours of a given sub-domain.

Furthermore, in our approach, the computation of cuts is straightforward. Instead, *ORB* [Salmon, 1990] requires a parallel median finder algorithm which, in turn, results in a large amount of communications to implement domain decomposition.

4 THE IMPLEMENTATION

This section describes a *MPI* algorithm to implement the Bounded Neighbours strategy. If n is the number of processes of the application, the algorithm partitions the domain into n sub-domains and assigns Dom_i to process P_i . Each process exploits a data structure, the *cut-array*, to store the initial and the final coordinates of each sub-domain, i.e. the *cuts* of the grid.

This structure is initialized when the elements are distributed to the processes and is updated after each load balancing step. Note that this structure does not store the exact location of the elements in the other sub-domains produced by a partition, it only describes the partition of the domain among the application processes.

The Bounded Neighbours algorithm consists of four phases:

- *Trade off Evaluation*
- *Cuts Computation*
- *Cuts Checking*

- *Data Exchange*

In the first phase, *Trade off* evaluation, processes exchange their current load. This information is exploited to evaluate the trade-off between the overhead introduced by the execution of the algorithm and the unbalance of the computation. The following phases are executed only if the trade-off is significant. In the *Cuts Computation* phase the processes compute the new partitions, i.e. the new cuts to balance the load. This phase can produce an illegal partition of the domain, i.e. a partition where the number of neighbours is larger than NS . In the following phase, *Cuts Checking*, a partition may be updated to produce a legal solution. Finally, in the *Data Exchange* phase, the processes exchange data to build the new partition of the domain. In the following, we will describe each phase in more detail.

4.1 Trade-Off Evaluation

We assume that each process, after each load balancing step, stores in a local data structure the cuts defining the partition of the domain and the current load of any other process. The load may be changed because, during a computational step, the number and/or the position of the elements in the domain may be modified. In this phase, processes exchange their *current load*, i.e. the number of elements currently belonging to a sub-domain. This communication is implemented by a *MPI Allgather* primitive.

After the collective communication, each process computes the optimal load and evaluates the trade-off between the overhead due to the execution of the load balancing algorithm and the benefits of a balancing step. The trade-off is defined by the following criteria:

- *Number of elements*

The current number of elements can be easily computed by summing the current load of any process. If this value is smaller than a *threshold percentage* P of the total number of positions of the grid, the load balancing step is not executed, because, the overhead of the load balancing is not balanced by the resulting speed up of the computation.

- *Maximum Unbalance*

The overall execution time of a computation step is determined by the execution time of the slowest process, i.e. the process P_i owning the sub-domain Di including *MaxD*?, the largest number of elements. The load balancing algorithm is executed only if the difference between the *MaxD* and the optimal number of elements of each process is larger than *threshold value* V .

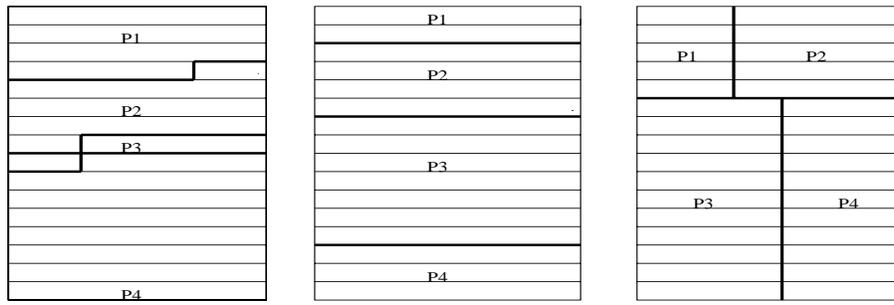


Figure 1. Load Balancing Strategies

The user can modify P and V to tune its application. An example is discussed in Section 5.

4.2 Cuts Computation

The *Cuts Computation* phase computes the new partition of the domain that assigns the optimal number of elements to each process. If the resulting partition does not satisfy the Bounded Neighbours condition, it will be modified in the next phase, *Cut Checking* which always generates a legal partitioning.

The *Cut Computation* phase consists of two steps. Let us denote by *old partition*, the partition computed in the previous load balancing step and by *new partition* that computed in this phase. In the first step, each process computes the *intersections* between the cuts defining the new partition and the sub-domains defined by the old partition. This computation exploits both the information gathered in the *trade-off evaluation* phase and the *cut-array* storing the cuts of the old partition.

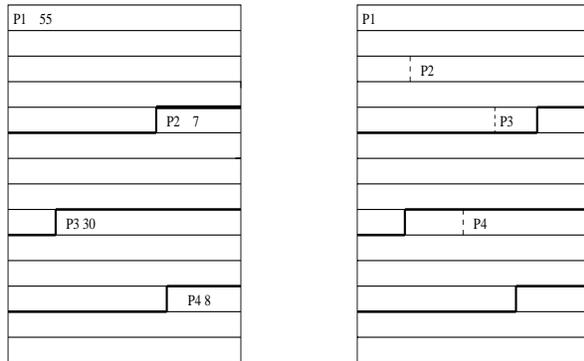


Figure 2. Cuts Computation

Consider, for instance, Figure 2. The leftmost part of the figure shows the old partition of the domain, which is recorded in the *cut-array*. For each sub domain we show a reference to the process owning the

sub-domain and the current load of that process. This information, gathered during the *trade off evaluation* step, is exploited to compute the optimal amount of elements to be assigned to each process, 25 in the considered example. In the right part of the figure, the *new cuts* are shown through dashed lines. It is worth noticing that the exact location of a new cut can be determined only by the process P_i owning the domain including the cut. Since any other process P_j , $j \neq i$, does not know the exact location of the elements in Dom_i , P_j can only determine the *number of cuts* of the new partition intersecting Dom_i . During this step, each process P_i builds a list, *ListCuts* and an array, *Receivecuts*. *Listcuts* stores the coordinates of all the cuts intersecting its domain. *Receivecuts* is an n elements array, where the j -th position records the number of cuts intersecting Dom_j , for any $j \neq i$.

Consider again Figure 2. Process P_1 computes the coordinates of the cuts that intersects the first sub-domain of the old partition and stores them in its *Listcuts*. All other processes store the value 2 in the first position of their *Receivecuts* array.

The code implementing this step is shown in Figure 3. We suppose that the variable *TotBalance* and *Optbalance* record, respectively, the total number of elements of the grid and the optimal amount of work to be assigned to each process. The i -th position of the array *ElPart* records the current load of process P_i .

When all the processes have completed this step, the exact location of the new cuts is notified by each process to any other one. This communication step is implemented by a loop executed in parallel by all the processes, according to an *SPMD* programming style. To notify its cuts to the other processes, at each iteration process P_i executes a distinct *MPI* broadcast communication as a sender for each of its cuts. Each process P_j $j \neq i$ executes, in turn, k *MPI broadcast*, as a receiver, where k is the value stored in *Receivecuts*[i].

```

Tot = TotBalance;
Diff= OptBalance;
i =0; CutPos=0;
Listcuts =  $\emptyset$ ;
while Tot>0
  if (Elpart[i]<Diff)
    Diff = Diff - Elpart[i];
    Tot = Tot - Elpart[i];
    i = i+1;
  else
    %calcolo nuovo taglio
    ElPart[i] = ElPart[i] - Diff;
    if myrank()== i
      Listcuts=Listcuts  $\cup$  NewCutUpdate(CurPos, Diff)
    else
      ReceiveCuts[i]=ReceiveCuts[i]+1;
    endif
    Tot = Tot -Diff;
    Diff= OptBalance
  endif
endwhile

```

Figure 3. Cuts Computation

4.3 Cuts Checking

This phase checks the *Bounded Neighbours* condition for the partition produced by the previous step. If appropriate, it modifies the partition as well. In the following, we show the implementation in the case where $NS = 2$. [Bonotti, 2002] describes the more general case. Each process considers the sub-domains in a sequential order and it checks the following conditions where Dom is the sub-domain that is currently considered.

- Each sub-domain Dom should include at least m grid elements, m is the number of columns of the grid. In this way Dom_i completely separates Dom_{i+1} from Dom_{i-1} . This implies that each domain has 2 neighbours. If this condition is not satisfied, all the processes shifts forward the cuts of the domains following Dom , in order to associate at least m points of the grid to Dom . This operation can introduce a certain amount of unbalance. However, our experiments show that this case is not very frequent and may arise only when the dimension of the grid is small with respect to the number of available processes.
- If Dom includes m or more grid elements, each process checks if the the bounded neighbours condition can be verified by all the sub-domains considered after Dom . This is possible if the number of grid cells from the final cut of Dom to the end of the grid is larger than m times the number of sub-domains still to be considered. If this condition is violated, then this is the first

domain violating the condition because the domain are considered one at a time. Then Dom and the following sub-domains can be updated to satisfy the condition.

4.4 Data Exchange

The phase implements, the actual exchange of the elements. Let OP_i be the old sub-domain associated with P_i and NP_j its new sub-domain. Each process:

- sends to P_j each element belonging to the intersection of OP_i with NP_j
- receives from P_j any element belonging to the intersection of NP_i with OP_j

In the example of Figure 2, process P_3 sends some elements to P_4 because some elements in its old sub-domain now belongs to P_4 . It also receives some elements from P_1 and all the elements of the partition assigned to P_2 .

5 WaTOR: AN IRREGULAR DYNAMICAL SIMULATION

The load balancing strategy defined in Section 4 has been exploited to implement WaTOR, a classical distributed simulation problem. This problem, originally introduced in [Dewdney, 1984], defines an idealized world where fishes and sharks move randomly, feed, breed and die. Plankton is located randomly at the vertices of the grid. Fish eat plankton, while sharks eat fishes.

The exact rules describing the behaviour of fishes and sharks are given in [Dewdney, 1984].

[Fox et al, 1988] observes that, even if these rules are too simple to describe a realistic biological population, the parallel implementation of *WaTOR* presents a number of interesting features characterizing more advanced parallel applications as well. First of all, the application is characterized by a very in homogeneous and dynamic load distribution. As a matter of fact, the distribution of the elements in the domain is not uniform and the number of the elements of the domain changes dynamically due to their death and breeding. Second, a *conflict resolution strategy* has to be defined to solve the conflicts arising among animals. For instance, two animals can decide to move to the same point of the grid, or two sharks can decide to eat the same fish. Since no specific rule is specified in [Dewdney, 1984], any choice is acceptable. In the parallel implementation, conflict resolution is more complex because it can involve several processes. As a matter of fact, conflicts can arise among processes which concurrently try to update the same cell of the grid belonging to a border of the sub-domain. Several strategies

have been proposed in [Fox et al, 1988], ranging from the simplest one which simply eliminates the animal losing the conflict, to that defining a complex *rollback strategy*.

Let us briefly describe the main characteristics of our implementation. Our implementation models the problem domain as a two-dimensional toroidal grid. Fishes and sharks are located at the vertices of this grid and can move only to the four nearest-neighbours vertices. A straightforward implementation stores the grid in a rectangular two-dimensional array. The drawback of this solution is that processes can spend a significant amount of time to examine empty regions of the ocean. For this reason, an application process P exploits a *linked list* storing only animals present on the grid. The grid is represented by a one dimensional array of n elements, where n is the number of the rows of the grid. Each element of the array is null if the corresponding row of the grid does not belong to P , otherwise it includes the list of all the animals in the row. This solution reduces the memory required for rows not belonging to the process. Furthermore, rows can be added or deleted during load balancing without modifying the rows not involved in the operation.

The conflicts are solved by avoiding the concurrent update of a grid point. In turn this is achieved by properly ordering the communications among the processes. The computation alternates a computation step and a load balancing step. During the computation, first each process updates the upper border of its domain and sends it, through an immediate *MPI* communication, to its neighbour. Then, each process updates the inner part of its domain. Before updating the lower border of its domain a process receives the updated upper border from its neighbour, it updates its lower border and it sends it back. After each computation phase, *Bounded Neighbours* is executed.

5.1 Results

This section shows some preliminary performance results. The experiments have been performed on Backus, a cluster of 8 PCs running Linux and connected by a Fast Ethernet network switch [Danelutto, 2003].

We have tested the effectiveness of the Bounded Neighbours algorithm through the Wator implementation and compared the result obtained considering grids of different sizes. For each case, we have compared the speed-up obtained when load is not balanced versus that achieved by exploiting *Bounded Neighbours*. The speed-up achieved for a 600×300 grid, rs. for a 1000×2000 grid are shown in the leftmost, rs. in the rightmost part, of Figure 4. The parameters of the simulation are shown in table 1.

The percentages are referred to the total number of elements of the grid. As far as concerns the 600×300

<i>number of fishes</i>	20 %
<i>number of sharks</i>	30 %
<i>initial amount of plancton:</i>	30%
<i>fish survival time</i>	7
<i>shark survival time</i>	3
<i>fishes breeding age</i>	10
<i>sharks breeding age</i>	12
<i>number of simulation steps :</i>	200

Table 1. Simulation Parameters

grid, we have that the overhead introduced by *BoundedNeighbours* equals the benefits obtained by balancing the load. As a matter of fact, the speed-ups achieved in the two experiments are comparable. In the case of the 2000×1000 grid, the speed-up that is achieved by balancing the load is larger that the one that is achieved without load balancing. Furthermore, the former speed-up is close to the optimal one.

We have also investigated the relation between the percentage of unbalanced load and the execution time, in the case of 8 processors and a 1000×2000 grid. Figure 5 shows the execution times corresponding to different unbalance degrees, i.e. to different values of the threshold value U , see section 4.1. The values shown on the x-axis corresponds to different percentage of unbalance with respect to the size of the grid. The value 0 corresponds to the execution time obtained when *Bounded Neighbours* is not exploited. The results show that, even for low unbalances, the execution time decreases when *Bounded Neighbours* is applied.

6 CONCLUSIONS

This paper has presented the *Bounded Neighbours* approach to domain decomposition. The most important feature of this approach is that the number of neighbours of each process is bounded. An implementation of the *BoundedNeighbours* approach has been developed and its effectiveness has been tested through WATOR, a classical irregular distributed simulation problem. Current implementation supports two neighbouring sub-domains for each domain produced by the partition. Preliminary results show a good trade off between the overhead introduced by the load balancing algorithm and the reduction of the execution time. We are planning to modify the decomposition procedure in order to support more cuts for each row. Furthermore, we will exploit our algorithm in the implementation of more complex simulation problems, such as real life problems described by cellular automata.

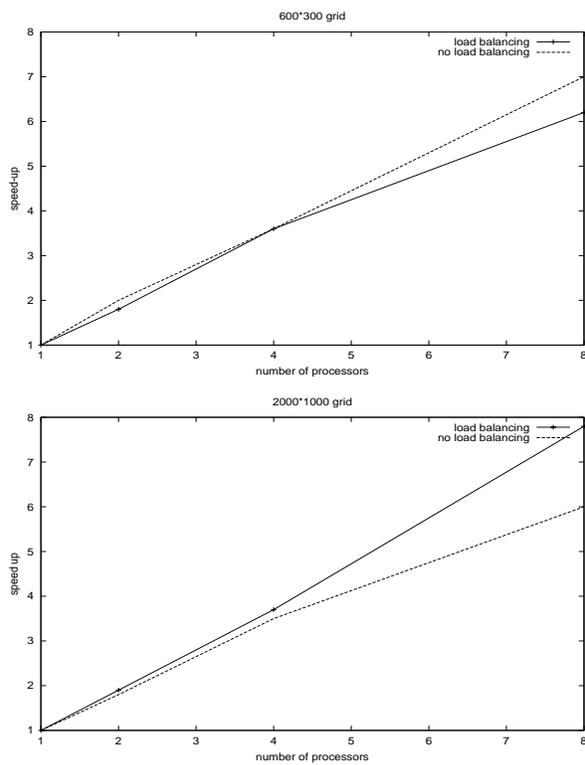


Figure 4. Water Scalability

References

- L.Ferrucci A.Bonotti. 2002 Domain partitioning: The bounded neighbours approach. Technical report, Dipartimento di Informatica.
- J.P.Singh D.E.Culler. 1998 *Parallel Computer Architecture: A Hardware/Software Approach*. Morgan Kaufmann.
- Dewdney. 1984 Computer recreation. *Scientific American*, December .
- D.M.Nicol and J.H.Saltz. 1990 An analysis of scattered decomposition. *IEEE transaction on Computers*, 39.
- G.Fox, M.Johnson, G.Lyzenga, S.Otto, J.Salmon, and D.Walker. 1988 *Solving Problems on Concurrent Processors*, volume 1. Prentice Hall.
- J.K.Salmon. 1990 *Parallel Hierarchical N-Body Methods*. PhD thesis, California Institute of Technology.
- J.P.Singh, C.Holt, T.Totsuka, A.Gupta, and J.L.Hennessy. 1995 Load balancing and data locality in adaptive hierarchical n-body methods: Barnes hut, fast multipole and radiosity. *Journal of Parallel and Distributed Computing*, 27(2):118–141.
- S.B.Baden J.R.Pilkington. 1995 Dynamic partitioning of non-uniform structured workloads with space-filling curves. *IEEE Transactions on Parallel and*

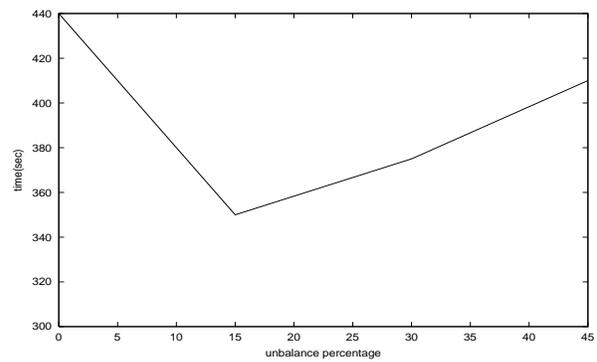


Figure 5. Execution time vs. Unbalance Percentage

- Distributed Systems*, 7(3):228–299, March 1995.
- M.Danelutto. 2003 The backus environment. Technical report, www.di.unipi.it/danelutto.
- B. Moon, H.V. Jagadish, C. Faloutsos, and J.H. Saltz. 2001 Analysis of the clustering properties of the hilbert space-filling curve. *Knowledge and Data Engineering*, 13(1):124–141.
- H. D. Simon. 1994 Partitioning of unstructured problems for parallel processing. In *Computing Systems in Engineering*, volume 2, pages 135–148.
- J.P. Singh, C. Holt, T. Totsuka, A. Gupta, and J.L. Hennessy. 1995 Load balancing and data locality in adaptive hierarchical N -body methods: Barnes-Hut, fast multipole, and radiosity. *Journal of Parallel and Distributed Computing*, 27(2):118–141.
- P. Sloot and D.Talia. 1999 Cellular automata: promise and prospects in computational science. *Future Generation Computing Systems*, (16).



Laura Ricci is currently an Assistant Professor at the Department of Computer Science, University of Pisa. Her research interests include the parallelization of irregular adaptive applications, the definition of methodologies and tools for teaching concurrency and the design of abstract interpretation based tools for parallelizing compilers.

A NOVEL REDUNDANT DATA UPDATE ALGORITHM FOR FAULT-TOLERANT SERVER-LESS VIDEO-ON-DEMAND SYSTEMS

T. K. HO and JACK Y. B. LEE

*Department of Information Engineering
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong*

Email: {tkho2@ie.cuhk.edu.hk, jacklee@computer.org}

Abstract: Recently, a new server-less architecture is proposed for building low-cost yet scalable video streaming systems. In this architecture, video data are distributed among user hosts and these hosts cooperate to stream video data to one another. To improve reliability, data and capacity redundancy are introduced to sustain node failures. However, the data placement as well as the redundant data in the system will need to be updated whenever new nodes join the system. This study is a first step in investigating the problem of updating redundant data when growing such a server-less system by assimilating new nodes. Results show that the redundancy update overhead is very significant and even exceeds that in data reorganization. To tackle this problem, this study presents a novel Sequential Redundant Data Update (SRDU) algorithm that takes advantage of the structure of Reed-Solomon Erasure Correction codes to reduce the redundancy update overhead by as much as 75%. Numerical results show that by further delaying the update of redundant data until adding multiple nodes, say 10, we can further reduce the redundancy update overhead by as much as 97%.

Keywords: Server-less, video-on-demand, redundant, update, reliable.

1. INTRODUCTION

Peer-to-peer and distributed computing has shown great potentials in high-performance computing applications. Apart from computational problems, data and I/O-intensive applications can also benefit from the inherent scalability offered by distributed architectures. One such architecture, called server-less video-on-demand architecture, recently proposed by Lee and Leung [Lee and Leung, 2002a] adopted this completely decentralized approach to eliminate the need for costly high-capacity video servers.

Unlike conventional video-on-demand (VoD) systems built around the well-understood client-server model, a server-less VoD system is built entirely from user hosts. Video data are distributed among these user hosts which then cooperate to stream video data to one another for playback. Lee and Leung [Lee and Leung, 2002a] showed that this server-less architecture is

easily scalable to hundreds of user hosts using off-the-shelf computers and network switches. Moreover, by incorporating data and capacity redundancy into the system, one can even achieve system-level reliability comparable to or even exceeding those of dedicated video servers [Lee and Leung, 2002b].

The study by Lee and Leung [Lee and Leung, 2002a] is focused on the scalability and feasibility of the server-less architecture. They did not, however, address the practical problem of system growth when new user hosts join the system. Specifically, as video data are distributed among user hosts, these data will need to be redistributed to newly joined hosts to utilize their storage and streaming capacity. This problem has been investigated by Ghandeharizadeh and Kim [Ghandeharizadeh and Kim, 1996], Goel et al. [Goel et al, 2002], and Ho and Lee [Ho and Lee, 2003] respectively. Nevertheless, all three studies are focused on the reorganization of the video data. The problem of updating redundant data that are themselves computed from the video data has not been addressed.

This work was supported in part by the Hong Kong Special Administrative Region Research Grant Council under a Direct Grant, Grant CUHK4211/03E, and the Area-of-Excellence in Information Technology.

In this study, we investigate the problem of efficient update of redundant data when video data are reorganized during the growth of a server-less VoD system. We found that updating redundant data can incur significantly more overhead than data reorganization. To tackle this problem, we are going to present a new redundant data update algorithm called Sequential Redundant Data Update that takes advantage of the structure of Reed-Solomon erasure codes [Plank, 1997] to reduce the redundant data update overhead by as much as 75%. Numerical results show that by further delaying the update of redundant data until adding multiple nodes, say 10, we can further reduce the redundancy update overhead by as much as 97%.

In the next section, we first briefly review the server-less VoD architecture and the previous works on data reorganization. We formulate the data reorganization problem in Section 3. The redundant data regeneration and proposed update algorithm are presented in Section 4 and 5 respectively; Section 6 gives the performance evaluation and Section 7 concludes the paper.

2. BACKGROUND

In this section, we first give a brief overview of the server-less VoD architecture [Lee and Leung, 2002a] and then review the existing works on data reorganization.

2.1 Server-less VoD Architecture

A server-less VoD system comprises a pool of fully connected user hosts, or called nodes in this paper. Inside each node is a system software that can stream a portion of each video title to as well as playback video received from other nodes in the system. Unlike conventional video server, this system software serves a much lower aggregate bandwidth and thus can readily be implemented in today's set-top boxes (STBs) and PCs. For large systems, the nodes can be further divided into clusters where each cluster forms an autonomous system that is independent from other clusters.

For data placement, a video title is first divided into fixed-size blocks and then equally distributed to all nodes in the cluster. This node-level striping scheme avoids data replication while at the same time share the storage and streaming requirement equally among all nodes in the cluster.

To initiate a video streaming session, a receiver node will first locate the set of sender nodes carrying blocks of the desired video title, the placement of the data blocks and other parameters (format, bitrate, etc.) through the directory service. These sender nodes will then be notified to start streaming the video blocks to the receiver node for playback.

Let N be the number of nodes in the cluster and assume all video titles are constant-bit-rate (CBR) encoded at the same bitrate R_v . A sender node in a cluster may have to retrieve video data for up to N video streams, of which $N - 1$ of them are transmitted while the remaining one played back locally. Note that as a video stream is served by N nodes concurrently, each node only needs to serve a bitrate of R_v/N for each video stream. With a round-based transmission scheduler, a sender node simply transmits one block of video data to each receiver node in each round. Interested readers are referred to the study by Lee and Leung [Lee and Leung, 2002a; Lee and Leung, 2002a] for more details.

2.2 Related Works

The problem of data reorganization has been studied in the context of disk arrays [Ghandeharizadeh and Kim, 1996; Goel et al, 2002]. The study by Ghandeharizadeh and Kim [Ghandeharizadeh and Kim, 1996] is the earliest study on data reorganization known to the authors. They investigated the data reorganization problem in the context of adding disks to a continuous media server. They employed round-robin data striping common in disk arrays and investigated and analyzed techniques to perform data reorganization online, i.e., without disrupting on-going video streams. However, their study assumed there is no data redundancy in the disk array and thus did not address the redundancy update problem.

In another study by Goel et al. [Goel et al, 2002], a pseudo-random algorithm called SCADDAR for data placement and data reorganization was proposed for use in disk arrays. In this algorithm, each data block is initially randomly distributed to the disks with equal probabilities. When a new disk is added to the disk array, each block will obtain a new sequence number according to their randomized SCADDAR algorithm. If the remainder of this number is equal to the disk number of the newly added disk, the corresponding block will be moved to this new disk. Otherwise, the block will reside at the original disk.

In a recent study [Ho and Lee, 2003], Ho and Lee proposed a more efficient data reorganization

algorithm called Row-Permutated Data Reorganization that can achieve lower data reorganization overhead and also allow controllable tradeoff between streaming load balance and data reorganization overhead.

While the previous pioneering studies have been successful in reducing the data reorganization overhead substantially, they did not yet address the issue of redundant data update. Given that a server-less VoD system is built from user hosts that are inherently less reliable than dedicated video servers, fault tolerant capability clearly becomes a necessity. To this end, one will need to incorporate data and capacity redundancies into the system and these redundant data will need to be updated whenever new nodes are added. To our knowledge this study is the first attempt at tackling this redundant data update challenge. Our study reveals that the overhead incurred in updating these redundant data far exceeds even the overhead in data reorganization.

3. OVERHEADS IN DATA REORGANIZATION

Based on the server-less VoD architecture presented in Section 2.1, we formulate the system model in this section and present the three types of overhead in reorganizing data to accommodate newly added nodes. Let B be the total number of fixed-size video data blocks in the system and v_j be the j^{th} block of the video title. For simplicity we consider only one video title although the results can be readily extended to multiple video titles.

Fig. 1 illustrates one possible placement of video data in a server-less VoD system. Each block in the figure represents either a Q -byte video data or a Q -byte redundant data block. Blocks under the same column are stored in the same node. The j^{th} redundant data block, denoted by c_{ij} , are computed from video data stripe i , comprising blocks $\{v_k, k=i(N-h), i(N-h)+1, \dots, (i+1)(N-h)-1\}$, using a systematic erasure-correction code such as the Reed-Solomon Erasure Correction (RSE) code [Plank, 1997]. Briefly speaking, with h redundant data blocks in a data stripe, the system will be able to sustain the failure of up to h nodes without losing any data. A previous study [Lee and Leung, 2002b] had shown that one can achieve system-level reliability comparable to high-end dedicated video server with redundancies of $h/(N-h) \approx 0.2$.

When one or more new nodes join the system, they will add both streaming load as well as capacity to the system. There are three types of overhead in

assimilating these new nodes into the system. First, to utilize their streaming and storage capacity, the system will need to redistribute portion of the video data to these new nodes. The system may also need to reorganize video data in the existing nodes to maintain streaming load balance [Ho and Lee, 2003]. This *data reorganization process* incurs overhead in the form of relocating data blocks within nodes in the system.

Second, as these redundant data are computed from the data stripe, relocation of the data blocks will require corresponding update to the redundant data blocks. This *redundant data update process* incurs overhead in transmitting data blocks to the nodes for regenerating the redundant data blocks.

Third, as the system grows larger with more nodes, the system reliability will decrease if the number of redundancies h is kept constant. To improve reliability, we will need to introduce new redundancies to the system (i.e., increasing h). This *redundant data addition process* incurs overhead in transmitting data blocks to the nodes for generating the new redundant data blocks.

To our knowledge, only the data reorganization process has been investigated [Ho and Lee, 2003; Ghandeharizadeh and Kim, 1996; Goel et al, 2002]. In this study, we investigate the redundant data update process and leave the redundant data addition process for future work. Common to all three processes, the goal is to minimize the overhead incurred when new nodes are assimilated into the system.

4. REDUNDANT DATA REGENERATION

For a general systematic erasure-correction code in a system with N nodes and h redundancies, we will need all $(N-h)$ data blocks in a stripe to compute the corresponding h redundant data blocks. As individual data and redundant blocks of a stripe are all stored in different nodes, the data blocks will all need to be transmitted to the redundant nodes (i.e., nodes storing the redundant data blocks) for regenerating the new redundant data blocks.

Therefore for a system with B data blocks, a total of B blocks will need to be transmitted to and received by the redundant node to support redundant data regeneration. Clearly this overhead is very significant and worst, increases with the system scale and level of redundancies.

On the other hand, if a central archive server storing all

video data is available in the system, then it can simply regenerate the new redundant data blocks locally and send them to the redundant nodes to replace the old redundant data blocks. In this case, the number of blocks sent will be reduced by a factor of $(N-h)$ to $(B/(N-h))$. Nevertheless maintaining this central archive server will incur its own costs, and depending on applications, may not be desirable or even feasible.

Reconsidering the generation of a redundant data block from a data stripe, we can observe that in most cases, the reorganized data stripe still comprises many data blocks from the old data stripe before reorganization. For example, in growing a system from N nodes to $N+1$ nodes, the first data stripe will be reorganized from the composition of $\{v_0, v_1, \dots, v_{N-h-1}\}$ to $\{v_0, v_1, \dots, v_{N-h-1}, v_{N-h}\}$, which differs by only one data block v_{N-h} . This motivates us to investigate techniques to reuse the old redundant block to compute the new redundant block such that only a portion of the data stripe will be needed.

5. SEQUENTIAL REDUNDANT DATA UPDATE

Among different erasure correction codes there is a class of codes called linear systematic block erasure correction codes, with the Reed-Solomon Erasure Correction code being one well-known example. One key property of linear systematic block codes is the use of strictly linear matrix multiplications in computing the redundant data, and this very property enables us to reuse original redundant data to compute the updated redundant data.

Specifically, let $(N-h)$ and h be the number of data nodes and redundant nodes in the system respectively. Assuming the number of redundant nodes in the system is fixed, then we can apply the (N, h) -RSE code to compute the h redundant data blocks from each stripe of $(N-h)$ data blocks using

$$\begin{aligned}
 F \cdot D &= \begin{bmatrix} f_{1,1} & f_{1,2} & f_{1,3} & \cdots & f_{1,N-h} \\ f_{2,1} & f_{2,2} & f_{2,3} & \cdots & f_{2,N-h} \\ \vdots & \vdots & \vdots & & \vdots \\ f_{h,1} & f_{h,2} & f_{h,3} & \cdots & f_{h,N-h} \end{bmatrix} \begin{bmatrix} d_{i,0} \\ d_{i,1} \\ \vdots \\ d_{i,N-h-1} \end{bmatrix} \\
 &= \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & 2 & 3 & \cdots & N-h \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & 2^{h-1} & 3^{h-1} & \cdots & (N-h)^{h-1} \end{bmatrix} \begin{bmatrix} d_{i,0} \\ d_{i,1} \\ \vdots \\ d_{i,N-h-1} \end{bmatrix} \\
 &= \begin{bmatrix} c_{i,0} \\ c_{i,1} \\ \vdots \\ c_{i,h-1} \end{bmatrix} = C
 \end{aligned} \tag{1}$$

where the F , D , and C are the Vandermonde matrix [Plank, 1997], the video data vector, and the redundant data vector respectively; and $d_{i,j}$, $c_{i,k}$ represent data block j ($j=0,1,\dots,N-h-1$) and redundant block k ($k=0,1,\dots,h-1$) of stripe i respectively. Elements in F is computed from $f_{i,j} = j^{i-1}$ and are constants. Note that the matrix multiplication in (1) is computed over Galois Fields of 2^w where $N < 2^w$. For example, by setting $w=16$ then the code can support up to 65,535 nodes.

In the following sections, we present a novel Sequential Redundant Data Update (SRDU) algorithm comprising three techniques, namely Reuse of Original Redundant Data, Parity Group Reshuffling, and Reuse of Transmitted Data, to substantially reduce the redundancy update overhead.

5.1 Reuse of Original Redundant Data

To illustrate how original redundant data can be reused, consider the examples in Fig. 1 and 2, which represent respectively the system configuration before and after the addition of one new node. In the original configuration in Fig. 1, there are 4 data nodes and 2 redundant nodes. Now the first two original redundant data in redundant node r_1 , denoted by $c_{0,1}$ and $c_{1,1}$, are computed from

$$c_{0,1} = \sum_{j=0}^3 f_{2,j+1} v_j \tag{2}$$

and

$$c_{1,1} = \sum_{j=4}^7 f_{2,j-4+1} v_j \tag{3}$$

according to (1).

After a new node is added, the system configuration will be changed to that in Fig. 2. Now the two new redundant data block, denoted by $c'_{0,1}$ and $c'_{1,1}$, are computed from

$$c'_{0,1} = \sum_{j=0}^4 f_{2,j+1} v_j \quad (4)$$

and

$$c'_{1,1} = \sum_{j=5}^9 f_{2,j-5+1} v_j \quad (5)$$

Comparing (4) with (2) we can observe that they share four common terms in $v_j - v_0, v_1, v_2, v_3$. Thus we can rewrite (4) as follows:

$$\begin{aligned} c'_{0,1} &= \sum_{j=0}^3 f_{2,j+1} v_j + f_{2,5} v_4 \\ &= c_{0,1} + f_{2,5} v_4 \end{aligned} \quad (6)$$

In other words, we can compute $c'_{0,1}$ using the original redundant data $c_{0,1}$ plus data block v_4 . Therefore instead of sending all five data blocks to redundant node r_1 , we now only need to send one data block, i.e., v_4 , thereby dramatically reducing the overheads in updating the redundant data $c'_{0,1}$.

5.2 Parity Group Reshuffling

In some cases, the previous straightforward reuse technique cannot be applied due to differences in the coefficients f_{ij} . For example, $c'_{1,1}$ is computed from v_5 to v_9 and share common terms in v_5, v_6 , and v_7 with $c_{1,1}$. Thus it may appear that we can reuse the common terms and send only v_8 and v_9 to r_1 to compute $c'_{1,1}$. However, analyzing the equation for $c'_{1,1}$ –

$$\begin{aligned} c'_{1,1} &= \sum_{j=5}^9 f_{2,j-5+1} v_j \\ &= (f_{2,1} v_5 + f_{2,2} v_6 + f_{2,3} v_7) + f_{2,4} v_8 + f_{2,5} v_9 \end{aligned} \quad (7)$$

and for $c_{1,1}$ –

$$\begin{aligned} c_{1,1} &= \sum_{j=4}^7 f_{2,j-4+1} v_j \\ &= f_{2,1} v_4 + (f_{2,2} v_5 + f_{2,3} v_6 + f_{2,4} v_7) \end{aligned} \quad (8)$$

we found that the common terms v_5, v_6 , and v_7 now have different coefficients f_{ij} (e.g., $f_{2,1} v_5$ versus $f_{2,2} v_5$). As a result, we cannot reuse $c_{1,1}$ in computing $c'_{1,1}$.

To tackle this problem, we propose to reshuffle the order of computations for $c'_{1,1}$ to

$$c'_{1,1} = f_{2,1} v_8 + (f_{2,2} v_5 + f_{2,3} v_6 + f_{2,4} v_7) + f_{2,5} v_9 \quad (9)$$

thus enabling us to reuse $c_{1,1}$ in the computation:

$$\begin{aligned} c'_{1,1} &= f_{2,1} v_8 + \left(\sum_{j=4}^7 f_{2,j-4+1} v_j - f_{2,1} v_4 \right) + f_{2,5} v_9 \\ &= f_{2,1} v_8 + (c_{1,1} - f_{2,1} v_4) + f_{2,5} v_9 \end{aligned} \quad (10)$$

This reduces the number of data block transmissions from 5 to 3. Note that the receiver node will also need to use the reshuffled order to correctly decode the parity group. This parity group order information can either be generated dynamically, or simply be sent along the video data blocks.

Interestingly, there may be more than one way to reuse redundant block in updating the redundant data, and possibly with different redundant update overhead. For example, consider the redundant generation function for $c_{2,1}$:

$$\begin{aligned} c_{2,1} &= \sum_{j=8}^{11} f_{2,j-8+1} v_j \\ &= (f_{2,1} v_8 + f_{2,2} v_9) + f_{2,3} v_{10} + f_{2,4} v_{11} \end{aligned} \quad (11)$$

If we reshuffle the order of computations for $c'_{1,1}$ to

$$c'_{1,1} = (f_{2,1} v_8 + f_{2,2} v_9) + f_{2,3} v_5 + f_{2,4} v_6 + f_{2,5} v_7 \quad (12)$$

then we can reuse $c_{2,1}$ in the computation:

$$\begin{aligned} c'_{1,1} &= \left(\sum_{j=8}^{11} f_{2,j-8+1} v_j - f_{2,3} v_{10} + f_{2,4} v_{11} \right) \\ &\quad + f_{2,3} v_5 + f_{2,4} v_6 + f_{2,5} v_7 \\ &= (c_{2,1} - f_{2,3} v_{10} + f_{2,4} v_{11}) \\ &\quad + f_{2,3} v_5 + f_{2,4} v_6 + f_{2,5} v_7 \end{aligned} \quad (13)$$

However, in this case the number of data block transmissions is 5, which is two blocks more than that of reusing $c_{1,1}$. Thus in the SRDU algorithm, the system will first compute the redundancy update overhead for all reusable redundant blocks and select the one with the lowest overhead for reuse.

5.3 Reuse of Transmitted Data

A third way to reduce overhead is to reuse data blocks already transmitted to a redundant node in computing another redundant data. Reconsidering the previous example in computing $c'_{1,1}$, the data blocks needed are v_4, v_8 , and v_9 . However, v_4 has already been sent to the redundant node when computing $c'_{0,1}$ (c.f. Equation (6)) and thus can simply be reused. As a redundant

block is computed from a stripe of $(N-h)$ data blocks, we need to cache at most $(N-h)$ data blocks from previous updates at the redundant node.

6. PERFORMANCE EVALUATION

In this section, we evaluate the Sequential Redundant Data Update algorithm using simulation. Beginning with a small system, we add new nodes to the system and then apply the SRDU algorithm to update the redundant data blocks. Performance is measured by the number of data blocks that need to be sent to the redundant nodes – or simply called redundancy update overhead. The total number of data blocks is 40,000 and is fixed throughout the simulation. For simplicity the redundancy update overhead for updating one redundant node is presented. For systems with more than one redundant node, the total overhead is simply multiplied by the number of redundant nodes.

6.1 Redundancy Update Overhead in Continuous System Growth

In the first experiment, we begin with a system of five data nodes and one redundant node. Then we add a new node to the system one by one, each time the redundant data blocks are completely updated using the SRDU algorithm. This continues until the system grows to 400 data nodes.

Fig. 3 plots the redundancy update overhead versus system size from 6 to 400. As expected, Redundant Data Regeneration performs the worst, essentially requiring all data blocks to be sent to the redundant node for regenerating the redundant data. On the other hand, regenerating redundant data using a centralized archive server incurs the least overhead, albeit at the expense of extra centralized facility.

Surprisingly, direct reuse of the original redundant data also performs very poorly. This is because the algorithm maintains the same data order within the parity group in computing the redundant data, and thus severely restricts the redundant data that can be reused. Once this restriction is relaxed by reshuffling the parity group, the overhead is reduced by half to around 20,000 blocks. Caching already transmitted data blocks further reduces the overhead by half to around 10,000 blocks. Thus with all three techniques combined, the SRDU algorithm can reduce the redundancy update overhead by as much as 75%.

6.2 Batched Redundancy Update

In the previous experiment, we always completely update all redundant data blocks before adding another new node. Clearly this is inefficient if new nodes are added frequently or added to the system in a batch. To address this issue, we conduct a second experiment where redundant data blocks are not updated until a fixed number of nodes, say W , are added – *batched redundancy update*. During this time, storage and streaming capacity in the new nodes are not utilized and thus this approach represents tradeoffs between redundancy update overhead and resource utilization. Fig. 4 plots the redundancy update overhead versus the batch size W for initial system size of 80 nodes. The key observation is that the normalized per-node redundancy update overhead decreases significantly with the batch size. A second observation is that the gain in caching transmitted data blocks reduces when the batch size increases. This is because the number of common term in reusable redundant blocks increases with larger batch size, and thus reducing the need for raw data blocks to update the new redundant data.

7. CONCLUSION AND FUTURE WORKS

This study is a first step in tackling the problem of redundant data update. As the results clearly showed, the redundancy update overhead is even more significant than data reorganization overhead and thus cannot be ignored. By taking advantage of the structure of RSE codes, we were able to substantially reduce the overhead by as much as 75%, and by performing update in a batch, we manage to reduce the overhead further by 97% for a batch size of 10. Nevertheless, batched redundancy update is not without tradeoffs. In particular, the storage and streaming capacity of the new nodes cannot be utilized until the system is reconfigured. Thus further investigations are warranted to quantify the tradeoffs to determine the optimal batch size that can balance between redundancy update overhead and resource utilization.

REFERENCES

[Lee and Leung, 2002a] Jack Y. B. Lee and W. T. Leung, “Study of a Server-less Architecture for Video-on-Demand Applications”. In *Proc. IEEE International Conference on Multimedia and Expo.*, August 2002.

[Lee and Leung, 2002b] Jack Y. B. Lee and W. T. Leung, “Design and Analysis of a Fault-Tolerant

Mechanism for a Server-Less Video-On-Demand System". In *Proc. 2002 International Conference on Parallel and Distributed Systems*, Taiwan, Dec 17-20, 2002.

[Ho and Lee, 2003] T. K. Ho and Jack Y. B. Lee, "A Row-Permutated Data Reorganization Algorithm for Growing Server-less Video-on-Demand Systems". In *Proc. International Symposium on Cluster Computing and the Grid 2003*, Tokyo, Japan.

[Ghandeharizadeh and Kim, 1996] S. Ghandeharizadeh and D. Kim, "On-line Reorganization of Data in Scalable Continuous Media Servers". In *Proc. 7th International Conference on Database and Expert Systems Applications*, September 1996.

[Goel et al, 2002] A. Goel, C. Shahabi, S.-Y. Yao, and R. Zimmerman, "SCADDAR: An Efficient Randomized Technique to Reorganize Continuous Media Blocks". In *Proc. International Conference on Data Engineering*, 2002.

[Plank, 1997] J. S. Plank, "A Tutorial on Reed-Solomon Coding for Fault-Tolerance in RAID-like Systems". *Software -- Practice and Experience*, vol. 27, no. 9, pp. 995-1012, September, 1997.

d_0	d_1	d_2	d_3	r_0	r_1
v_0	v_1	v_2	v_3	$c_{0,0}$	$c_{0,1}$
v_4	v_5	v_6	v_7	$c_{1,0}$	$c_{1,1}$
v_8	v_9	v_{10}	v_{11}	$c_{2,0}$	$c_{2,1}$
v_{12}	v_{13}	v_{14}	v_{15}	$c_{3,0}$	$c_{3,1}$
v_{16}	v_{17}	v_{18}	v_{19}	$c_{4,0}$	$c_{4,1}$

Fig. 1. Data placement before addition of nodes.

d_0	d_1	d_2	d_3	d_4	r_0	r_1
v_0	v_1	v_2	v_3	v_4	$c'_{0,0}$	$c'_{0,1}$
v_5	v_6	v_7	v_8	v_9	$c'_{1,0}$	$c'_{1,1}$
v_{10}	v_{11}	v_{12}	v_{13}	v_{14}	$c'_{2,0}$	$c'_{2,1}$
v_{15}	v_{16}	v_{17}	v_{18}	v_{19}	$c'_{3,0}$	$c'_{3,1}$

Fig. 2. Data placement after adding one data node.

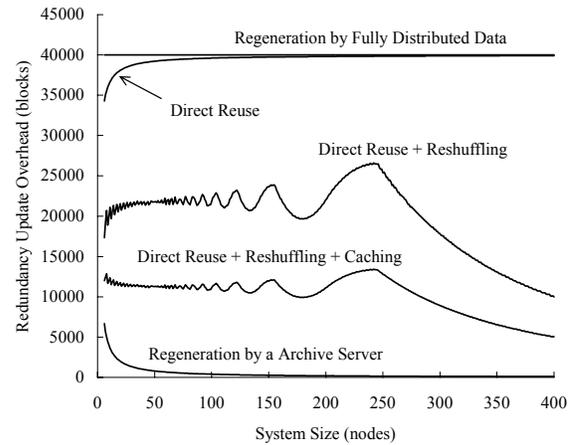


Fig. 3. Redundancy update overhead versus system size.

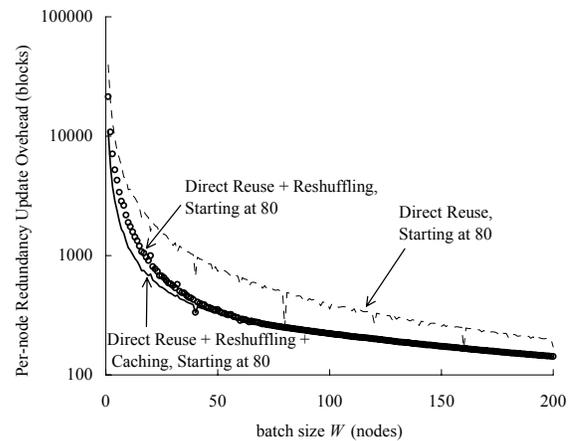


Fig. 4. Per-node redundancy update overhead versus batch size.



T.K. Ho received his BEng degree in Information Engineering from The Chinese University of Hong Kong in 2002. He is currently studying for his MPhil degree at the same department and participating in the Server-less Video Streaming Systems Project in the Multimedia Communications

Laboratory (<http://www.mcl.ie.cuhk.edu.hk>).

BANDWIDTH MANAGEMENT IN A CENTRALIZED LARGE SCALE DISSEMINATION NETWORK – A SIMULATION STUDY

KONSTANTINOS G. ZERFIRIDIS

*Department of Informatics
Aristotle University of Thessaloniki, 54124, Greece
zerf@csd.auth.gr*

HELEN D. KARATZA

*Department of Informatics
Aristotle University of Thessaloniki, 54124, Greece
karatza@csd.auth.gr*

Abstract: The evolution of the Internet gave rise to new applications. The need to disseminate high volumes of data to numerous users along with the evolution of Peer-to-Peer systems, introduced a new alternative to the traditional client-server paradigm. File sharing networks became the platform for thousands of users to share content. Users often turn to these networks to find highly anticipated, newly released software or video files which sometimes are of considerable size. However, increased mean response time or even network failures can be observed in such P2P systems, often caused because of uneven flow of data and intersperse congestion points. In this paper, the structure of Peercast, an agent based dissemination network, is presented. Several simulation experiments were conducted and their results are examined in order to determine how the network's bandwidth can be best utilized during the dissemination process.

keywords: peer-to-peer, network modeling, middleware, grid computing

1. INTRODUCTION

As bandwidth availability is increasing, users' demands change constantly. Today the internet is used to download music, software, video clips and other files of considerable size. This can saturate the network quickly, clogging the host computer. Such is the case for example when any highly anticipated software is released and several people are trying to download it at the same time. This became known as the middle night madness problem [Schooler and Gemmell, 1997]. Conventional FTP servers can no longer serve as a way of distributing large amounts of data. For example, modern Linux distributions can span more than one CD. Assuming that the server's bandwidth is 1 MBit/sec and the requested software is distributed in 2 ISO CD images, the server could only serve about 50 clients in a period of one week, even in the theoretical case that no errors occur. Mirroring the required content on several dispersed servers, cannot always compensate for the rapid traffic increase.

In such cases, traditional ways of making data available to the masses do not apply to modern demands. The main architecture used for casting data through the Internet is IP multicast, which mainly targets real-time non-reliable applications. It extends the IP architecture so that packets travel only once on the same parts of a network to reach multiple receivers. A transmitted packet is replicated only if it needs to, on network routers along the way to the receivers. Although it has been considered as the

foundation for Internet distribution and it is available in most routers and on most operating systems, IP multicast has not so far lived up to early expectations. Its fundamental problem is that it requires that all recipients receive the content at the same time. The most popular solution to this problem was to multicast the content multiple times until all of the recipients obtain it. Some of the other drawbacks of IP multicast include small address space (26-bit), need of large routing tables and lack of congestion control and reliable transfer control.

Several algorithms arise for membership management and packet replication to solve problems such as server implosion from client side NACKs (negative acknowledgments), server explosion from maintaining status of the download process for each client and managing downloads requests by users connected with different bandwidths. Forward Error Correction (FEC) has long been used for the dissemination of static data as it provides graceful degradation of performance in the presence of packet losses. Its greatest disadvantage is that it is very demanding on CPU and memory [Rizzo, 1997].

Although IP multicast might be considered ideal for applications that require relatively high and constant throughput but not much delay, it is not suitable for applications that may tolerate significant delays but no losses. This is the case with file distribution. These days, a new way of disseminating files emerged. File sharing networks [Parameswaran et al,

2001] are perhaps the most commonly used Peer-To-Peer applications. P2P systems existed since the birth of the Internet, but as bandwidth, computational power and great storage capacity became available, their popularity increased. Such systems have been used for diverse applications: combining the computational power of thousands of computers, forming collaborative communities, instant messaging, etc.

P2P file sharing networks' main purpose is to create a common pool of files where everybody can search and retrieve any shared files. Depending on the algorithm used, these sharing networks can be divided in two groups. Networks that maintain a single database of peers and their content references are known as centralized. Such file sharing networks [Shirky, 2001] have several advantages, such as easy control and maintenance, and some disadvantages as, for example, server overload. On the other hand, dynamically reorganizing networks such as Gnutella [Ripeanu, 2001], have a rather more elaborate service discovery mechanism, avoiding this way the use of a centralized server. Those kinds of networks are known as decentralized, and their main advantage is the absence of a single point of failure. However, the lack of a coordinating server may lead to inefficient use of the network's resources.

Along with the widespread use of those networks, several problems emerged. A study conducted at the Xerox Palo Alto Research Center showed that 70% of Gnutella users provided no files or resources to the system and that 1% of the users were providing half of the total system resources [Adar and Huberman, 2000]. This created network bottlenecks causing further inter-domain jamming. File sharing networks had never been designed for file dissemination. Nevertheless people turn to them to find highly anticipated files, when the official server stops responding due to high demand. Extensive research has been done about how existing P2P networks operate over time and how they can be optimized [Markatos, 2002; Ripeanu et al, 2002]. However, the dissemination process of highly anticipated files on P2P networks over unreliable network connections remains unexplored. Peercast, a P2P network first presented in [Zerfiridis and Karatza, 2003], is designed to assist the dissemination of a file in a heterogeneous network of clients. The purpose of this paper is to show how the Peercast performs under different bandwidth utilization scenarios using a simulated model of the network. The drawn conclusions can be used to optimize other P2P file sharing networks as well.

The structure of this paper is as follows. In section 2 PeerCaster, the agent based infrastructure used, is presented. Section 3 shows Peercast's structure, along with its latest extensions. Section 4 elaborates

on the network's simulation model and in section 5 the results and drawn conclusions are summarized. Finally, section 6 presents plans for further research.

2. THE INFRASTRUCTURE

Software agents are programs that act on behalf of clients. They are able to perform predefined tasks that are assigned to them. This is done either with or without the supervision of the user, depending on the given job. Mobile agents have an additional property [Chess et al, 1995]. The ability to transport themselves on different systems after being executed, carrying with them their program code, current state of execution and any data which was obtained. This gives them the unique capacity of living on a distributed network rather than on a distant stationary system, and to take advantage of the services that each host has to offer locally. Furthermore, mobile agents allow proprietary code to be used on the hosts, allowing complete customization of the retrieved results.

The unique properties of the mobile agents give them the edge in comparison to the traditional client-server paradigm. They have been used in the past instead of protocols [Joy, 2000], for file transfer [Spalink et al, 1999] and as a dynamic system for information discovery and retrieval. There are many applications that would benefit from the use of mobile agents as a vehicle for getting around bottlenecks. PeerCaster [Zerfiridis and Karatza 2002] is a platform implemented in Java that uses mobile agents as a vehicle delivering great amount of static data to users on a heterogeneous network. This is done by splitting the data into small packets, loading them onto mobile agents and releasing them to the peers where the payload is delivered and continue according to their itinerary. The coordination and communication overhead is acceptable considering the scalability that can be gained by the dynamic nature of the agents. As they can operate asynchronously and independently of the process that created them, they do not need to report back to the server. In this paper, PeerCaster was used as a mean of distributing high-demand files without clogging the host computer. This system could be integrated as part of a P2P file transfer network, or it could be used as an alternative to multicast for large files with great demand, such as the release of a new version of popular software as depicted in [Schooler and Gemmel, 1997].

3. THE NETWORK

When a file needs to be downloaded by more clients than the server can handle, alternative algorithms have to be utilized. The naive way of avoiding retransmissions is to pipeline the file through all the clients. But this is not a viable solution because clients might have to indefinitely wait to be served.

The proposed algorithm uses centralized approach in order to avoid uneven flow of data and intersperse congestion points which can compromise inter-domain quality of service. The server can upload the file to a certain number of clients simultaneously. When the server successfully uploads a file to a client, it keeps a reference of this client to a list. The server has the responsibility of maintaining a complete list of served clients that are currently on-line.

Although the server has a queue, most of the clients are expected to find this queue full. This is the case especially at the beginning of the dissemination process, as clients arrive more rapidly than the server can handle. When this happens, the server sends to the client a short (up to 100 entries) list of randomly selected peers that downloaded successfully the file, and are known to be on-line. This way, the new client can download the file from a peer that was already served, removing the congestion from the server. If the client cannot be served by any of those peers it requests another list of clients in order to continue searching for service. If the server is contacted more than 10 times, or the returned list is less than 100 entries long, the client waits for a certain period of time before it contacts the server again. If a client cannot contact a peer either because it is off-line or because it is unreachable due to network failure, it sends to the server a short message so that the server can update its database.

As it was mentioned earlier, when a client finishes the download it acts as a server for other clients. Similarly to the server, the clients have a short queue. If a client *A* requests the file from a client *B* that has it, and client *B* can not serve client *A* immediately, *A* is queued. If the queue is full, client *B* dismisses client *A*. When a client finishes the download, it sends a short report message to the server in order to include it in its list.

When a peer leaves the network, the list maintained at the server is left in an inconsistent state. In order to compensate for this, clients that are not able to contact other peers, report to the server that this peer is no longer reachable. If the server receives several such reports for the same peer, it removes its reference from the list.

In order to utilize all the available upload bandwidth, a single peer can serve several clients concurrently. Additionally, each client can initiate multiple concurrent download connections in order to utilize all the available download bandwidth. At the end of the transfer, the downloading client sends a message to the server in order to be included in the list.

Several issues arise about the performance of this algorithm under different network conditions in a heterogeneous network of clients. For example, what is the benefit of allowing several clients to download from a single peer? It will reduce the average waiting time, but what consequences will it have on the downloading speed and in the long run on the total number of served clients? On the other hand, if the clients are able to download from multiple peers simultaneously, how will it affect the system's dissemination process? This can in theory utilize all the download bandwidth of client and therefore, reducing the mean response time.

4. SIMULATION MODEL

In this section details are presented about the simulation model for the proposed network, and show how different strategies might affect the dissemination process. An object-oriented model of the network was used for the simulation. The programming language used was Java. The system was populated with clients arriving according to the exponential distribution. The simulation period was set to be 2 weeks (1209600 seconds). During the first week the mean interarrival time was incremented linearly from 5 to 20 sec in order to simulate demand on a highly anticipated file. For the second week the exponential distribution was used with 20 sec mean interarrival time. The file size was set to be 650MB (the size of a full CD).

All the clients that populated the system were set to have broadband connections to the Internet, resembling cable modems and DSL. This is done in order to use a realistic model. As in many cases, such connections have different download and upload speeds. Four different categories of users were used. The first category (10% of the clients) had download and upload speed of 256 Kbps, the second (40% of the clients) had 384 Kbps and 128 Kbps respectively, the third (20% of the clients) had 384 Kbps download and 384 Kbps upload speed, and the fourth (30% of the clients) had 1.5 Mbps and 384 Kbps respectively. This configuration is a theoretical model, and is used to compare how the same network performs under different conditions.

These kinds of clients are always on-line. However, they are not expected to share the file for ever. Therefore they were set to leave the dissemination network with exponential distribution and mean time of four days. The server was set to have 1.5 Mbps download / 384 Kbps upload connection (resembling a DSL user) to the net and never to go off-line. As the server is only uploading files, the simulation would have given the same results if the server had 384/384 connection to the net (third category). An additional difference between the server and the clients is that the server keeps a list of all the served

Table 1. Mean response time				
Concurrent download streams	Concurrent upload streams			
	1 slot	2 slots	4 slots	8 slots
1 slot	134785	164878	193065	283042
2 slots	118351	127631	157199	251143
4 slots	117844	122076	149080	241583
8 slots	122760	123780	145910	239038

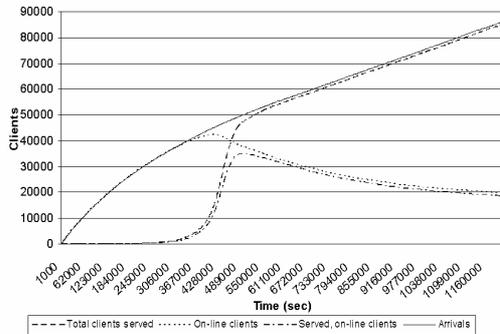


Figure 1. Network's status over time, 4 upload and 4 download streams

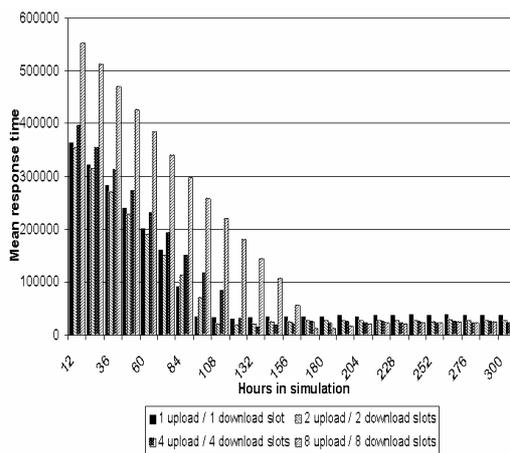


Figure 2. Mean response time in 12-hour intervals according to each client's arrival

clients that are currently on-line. This list is constantly updated.

The actual connection speed between two clients is calculated at the beginning of each session, taking into consideration the theoretical maximum speed they could achieve and an exponentially distributed surcharge, in order to simulate additional network traffic and sparse bottlenecks. If a new client cannot be served or queued immediately, it waits for 600 seconds and retries. In order to simulate peers that are not willing to assist in the dissemination process, 10% of the clients were set to go off-line immediately after they finish downloading the file. This is expected to significantly decrease the performance of the dissemination process. Nevertheless it is a behavior that can be expected.

If a client cannot contact another peer, it sends a message to the server that this peer is unreachable. When the server receives three such messages from different clients for the same peer, that peer is removed from the list. This is done to avoid removing a client from the list just because one connection could not be established. However, if a client that is participating in the dissemination process is not requested to serve another peer for over 1200 seconds, it contacts the server to verify that it is still included in the server's list. This is done as a countermeasure to accidental removals from the list.

As it was mentioned earlier, the behavior of this network can change significantly under certain conditions. The system's performance is investigated at the beginning (2 weeks) of the dissemination, under different conditions. Our focus is on how the system behaves under different bandwidth loads. More specifically, the simulations tested the system's performance when 1, 2, 4 and 8 concurrent upload streams were used. In each case, a serving-client was able to serve one or multiple peers at the same time by sharing the client's bandwidth. By sharing the bandwidth to multiple peers the full bandwidth is utilized, but the connection speed decreases. Additionally, the system's performance was tested with clients that were able to download from 1, 2, 4 and 8 serving-clients simultaneously. If a client can not use all its available download streams it retries to find an available serving-client after 600 seconds. With this approach, the client's download bandwidth can be utilized to the maximum. On the other hand, several serving-clients are occupied by serving one client, diminishing this way overall network performance.

5. SIMULATION RESULTS AND CONCLUSION

In total 16 simulations were done. Table 1 reveals significant differences between the tested scenarios. The increased mean response time in all cases can be explained as the clients that arrive early on the dissemination process have to wait for a long period of time to be served. When the rate of arrivals balances with the rate of clients being served, the mean response time stabilizes to lower levels. This balance occurs when a critical mass of serving-clients has been built. The critical mass is reached when the number of served clients in the system starts to decline (figure 1). Therefore, clients arriving later in the system benefit from a faster service. This is depicted in figure 2 where mean response time is shown in 12 hour intervals according to each client's arrival in the system.

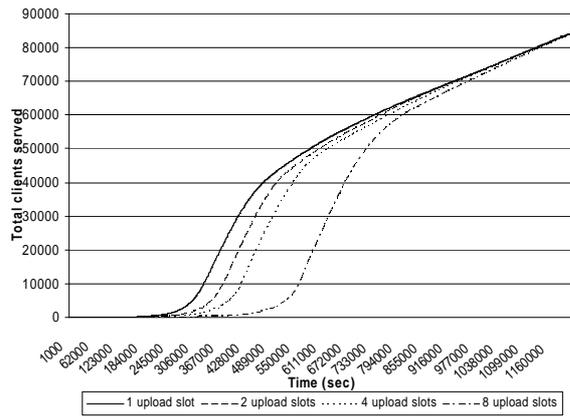


Figure 3. Total clients served over time (1 download stream)

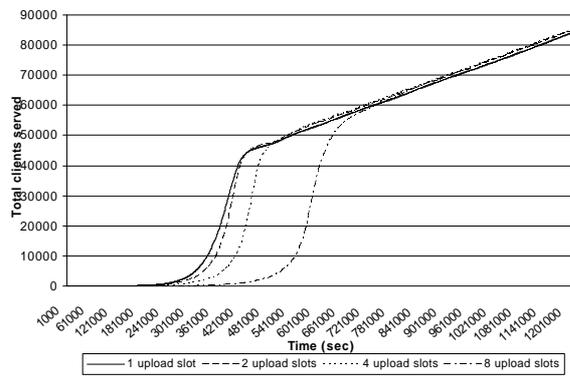


Figure 4. Total clients served over time (8 download streams)

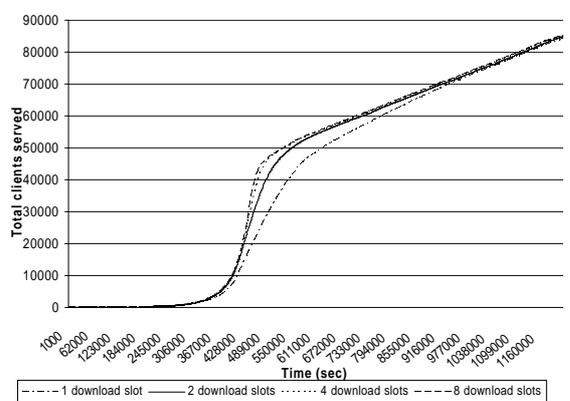


Figure 5. Total clients served over time (4 upload streams)

Table 1 shows that using 8 concurrent upload streams for each client increased dramatically the mean response time in all cases. Additionally, figure 2 shows that although the mean response time for the 8/8 case is increase at the beginning of the dissemination process, after the critical mass is reached it decreased dramatically, even to lower levels than those of the other cases,. The explanation for this is that when the critical mass has been built, there are enough serving-clients in the system to accommodate the rest of the peers and the new clients as they arrive. Therefore, the multiple upload streams utilize the client's upload bandwidth to the

maximum and assist the peers in finding service immediately as they arrive in the system.

On the other hand, multiple upload streams have the opposite affect at the beginning of the dissemination process. Sharing the serving-client's bandwidth to multiple peers reduces the downloading speed. This increases the response time, and therefore the critical mass is built much later in the dissemination process. This is shown in figures 3 and 4, where as the upload streams increase, the time period in which the system reaches the critical mass increases also. Comparing these two figures reveals also that the use of multiple download streams accelerates the build of the critical mass.

Figure 5 reveals that for the 4 upload streams case, the more download streams used, the sooner the critical mass is built. This can be seen in table 1 as well, where shorter mean response time is observed as the download streams increase. This is also the case for the 8 upload streams scenario. However, table 1 shows that this is not valid for the other two sets of tests. For example, the 1 download / 4 upload streams test produced reduced mean service time in comparison with the 1 download / 8 upload streams case. This shows that although multiple download streams have a positive affect on the utilization of the given bandwidth, they can also be accountable for the depletion of network resources.

Overall, the system's behavior can change dramatically by using different bandwidth utilization scenarios. Increased number of download streams helped in all the cases to the faster build of the critical mass. However in some cases this was the reason for an increase in the mean response time. On the other hand the use of multiple upload streams increased the mean response time before the build of the critical mass, but afterwards it decreased the mean response time. We propose the use of a dynamically changing number of upload and download streams as the dissemination process develops. The server can estimate when the critical mass is reached by the size of the list of serving-clients that it maintains. Before that point, using 2 upload and 4 download streams can speed up the build of the critical mass. After that point, by gradually increasing the upload and download streams to 8, is expected to decrease the mean response time. Simulation results of this scenario are under way.

6. FUTURE WORK

The use of a decentralized approach, as described in [Zerfiridis and Karatza, 2003], is also investigated in order to determine the best upload/download bandwidth utilization scenario in that case. Additional simulation experiments are under way, using distributions varying with time for more

realistic long-run simulations, as depicted in [Karatz, 2002]. Peercast is an evolving platform. For the current P2P network implementation we used a monolithic approach: all the data has to be sent to a client, before this client starts sending it to another peer. A new version that replicates groups of 256KB packets, to adjacent peers as they arrive, is under way. This is expected to alleviate the problems that are caused from peers that go off-line immediately or soon after they finish downloading the requested file. The synchronization between the peers is done in predetermined time intervals, called epochs [Karatz and Hilzer, 2001]. The peers are segmented in virtual groups according to their bandwidth and the epoch size depends on an estimation of the minimum bandwidth between the peers that form each dissemination group. Simulation results from this network are expected to show alleviation of several issues raised in this paper such as the increased mean response time at the beginning of the dissemination. An alternative way which we also investigate is to use prior knowledge of a peer's content to push newly arrived packets.

REFERENCES

Adar E. and Huberman B.A. 2000, "Free Riding on Gnutella", *Technical report*, Xerox Palo Alto Research Center.

Chess D.M., Grosf B., Harrison C.G., Levine D., Parris C. and Tsudik G. 1995, "Itinerant Agents for Mobile Computing", *Journal of Personal Communications*, IEEE Computer Society, Vol. 2 (5). Pp34-49.

Joy B. 2000, "Shift from Protocols to Agents", *Internet Computing*, IEEE Computer Society, Vol. 4 (1). Pp63-64.

Karatz H.D. 2002, "Task Scheduling Performance in Distributed Systems with Time Varying Workload", *Neural, Parallel & Scientific Computations*, Dynamic Publishers, Atlanta, Vol. 10. Pp325-338.

Karatz H.D. and Hilzer R.C. 2001, "Epoch Load Sharing in a Network of Workstations", *In Proc. 34th Annual Simulation Symposium*, IEEE Computer Society Press, SCS, Seattle, Washington. Pp36-42.

Markatos E.P. 2002, "Tracing a large-scale Peer to Peer System: an hour in the life of Gnutella", *In Proc. CCGrid 2002*, Second IEEE/ACM International Symposium on Cluster Computing and the Grid. Pp65-74.

Parameswaran M., Susarla A. and Whinston A.B. 2001, "P2P Networking: An Information Sharing Alternative", *Computer Journal*, IEEE Computer Society, Vol. 34. Pp31-38.

Ripeanu M., Foster I. and Iamnitchi A. 2002, "Mapping the Gnutella Network: Properties of large scale peer-to-peer systems and implications for system design", *Internet Computing Journal*, IEEE Computer Society. Pp50-57

Rizzo L. 1997, "On the feasibility of software FEC", *Technical report*, Univ. di Pisa, Italy.

Schooler E. and Gemmel J. 1997, "Using Multicast FEC to solve the Midnight Madness Problem", *Technical Report*, Microsoft research.

Shirky C. 2001, *Peer-to-Peer: Harnessing the Benefits of a Disruptive Technology / Listening to Napster*, ed. I.A. Oram, O'Reilly & Associates.

Spalink T., Hartman J.H. and Gibson G. 1999, "The Effects of a Mobile Agent on File Service", *In Proc. First International Symposium on Agent Systems and Applications*, Third International Symposium on Mobile Agents (ASA/MA '99), Palm Springs, California, IEEE Computer Society. Pp42-49.

Zerfiridis K.G. and Karatz H.D. 2002, "Mobile Agents as a Middleware for Data Dissemination", *Neural, Parallel & Scientific Computations*, Dynamic Publishers, Atlanta, Vol. 10. Pp313-323.

Zerfiridis K.G. and Karatz H.D. 2003, "Large Scale Dissemination using a Peer-to-Peer Network". To appear in the *Proceedings of the 3rd International Workshop on Global and Peer-to-Peer Computing on Large Scale Distributed Systems*, IEEE/ACM International Symposium on Cluster Computing and the Grid 2003, Tokyo.



KONSTANTINOS G. ZERFIRIDIS received his Diploma degree in Mathematics in June 1998 at the Aristotle University of Thessaloniki. In 1999 he received his M.Sc. degree in computer science from the University of Edinburgh. He is currently a researcher and working towards a Ph.D. at the Aristotle University of Thessaloniki. His research interests are mobile computing, mobile agents, distributed and Peer-to-Peer systems.



HELEN D. KARATZA is an Associate Professor in the Department of Informatics at the Aristotle University of Thessaloniki, Greece. Her research interests mainly include Performance Evaluation of Parallel and Distributed Systems, Multiprocessor Scheduling, Mobile Agents, Mobile Computing, and Simulation. Dr. Karatza is a member of the Editorial Board of the International Journal of Simulation: Systems, Science & Technology (the UK Simulation Society), Associate Editor of the Journal Simulation: Transactions of the Society for Modeling and Simulation International and area Editor for computer systems of the Journal of Systems and Software (Elsevier).

A THEORETICAL FRAMEWORK FOR MODELLING AND SIMULATING SECURITY PROTOCOLS

FRANTZ O. IWU and RICHARD N. ZOBEL

*Department of Computer Science
University of Manchester
Oxford Road, Manchester, M13 9PL
United Kingdom
E-mail: {iwuo, rzobel}@cs.man.ac.uk*

Abstract:

The aim of this paper is to present an approach to describe cryptographic protocols using agent-based simulation. This provides a framework to understand and model protocol behaviour and interaction in a simulation environment. Simulation techniques in the past have proven to be useful especially in areas where it is critical for testing to be carried out. This allows the designer to determine the correctness and efficiency of a design before the real system is constructed and deployed. Hence, an attempt to use this approach in testing the correctness of cryptographic protocols is promising.

Keywords:

Security, Protocols, Agent-Based Simulation

1. INTRODUCTION

Cryptographic protocols are designed to provide security services. Research has shown that a good number of these protocols are flawed. One reason for these failures, is primarily the lack of proper universally accepted technique and methodology for describing and analysing these protocols. Several successful attacks against cryptographic protocols, which exist in academic literatures show that weaknesses are not due to the underlying cryptographic algorithms but are as a result of logical errors. To deal with these problems, several methods have been proposed. These include methods based on specification languages and verification tools [Varadharajan, 1990], modal logic, expert systems, algebraic reasoning, and model-based approaches [Nieh, 1992; Gong, 1990; Burrows, 1990].

In this paper an approach to reasoning about the security of a protocol, which involves the use of agent models to characterise how principals interact is described. Furthermore, it describes how messages are sent and received, what messages a particular agent can assemble and transmit, the actions an agent can perform at a particular time and the use of simulation framework in modelling the activities of these agents. These characterisations form the bases for asking security related questions such as: what are the possibilities, given all possible situation and interactions, of security compromises. First, a conceptual model of the system needs to be designed, which describes how agents may communicate within a simulation environment and formalised using the Discrete Event Simulation (DEVS) formalism [Zeigler, 2000]. DEVS describes the autonomous and

dynamic behaviour of agents and how agents react and generate input and output events at the atomic and coupling levels. Second, there is a need to design a simulation model, which enables agents to react and respond to events such as an intruder activity. Finally, the Needham-Schroeder and DSE protocols are considered using this approach.

2. AGENT-BASED SYSTEMS

Research and development of agent-based systems as a solution for various problem domains are rapidly increasing. Agents are in fact a key contributing technology for the Internet and World Wide Web and can be classified in several dimensions. The concept of deliberative agents was derived from the deliberative thinking paradigm in which agents hold an internal reasoning model from which it can make decisions to meet set goals. These kinds of agents are found in the area of artificial intelligence, psychology, cognitive sciences where agents have been modelled with personality traits and passion for decision-making [Baillie, 2002; Schmidt, 2002]. Conversely, reactionary agents do not have any internal symbolic model but make decisions based on stimulus or reaction from its environment. Agents can be classified according to their attributes such as autonomy, learning ability, interaction and cooperation. An important attribute of an agent is its ability to take initiatives and learn from past experience as it reacts and/or interacts within or outside its environment.

3. AGENT FRAMEWORK

A cryptographic protocol is considered to include a set of agents and channels of communication.

These agents interact with each other according to some predefined rules and processing messages sent and received via the communication channels. A channel is an abstraction of the communication facility that has certain constraints. Each agent is an autonomous and reactionary entity capable of performing a sequence of operations (events) on messages. The agent formalism is characterised by the tuple modelled at the atomic level.

$$\Sigma_{\text{Agent}} = (X, S, Y, \delta_{\text{int}}, \delta_{\text{ext}}, \lambda, \text{ta})$$

$X = \{x_1, x_2, \dots, x_n\}$ is a non empty set of input events.
 $S = \{s_1, s_2, \dots, s_n\}$ is a non empty set of allowable states.

$Y = \{y_1, y_2, \dots, y_n\}$ is a non empty set of output events.
 $\delta_{\text{int}} : S \rightarrow S$ An internal state transition function describing the behaviour of a Finite State Automaton.

$\delta_{\text{ext}} : Q * X \rightarrow S$: An external state transition function describing reaction of the agent to external events, where $Q = \{(s, e) \mid s \in S, 0 \leq e \leq \text{ta}(s)\}$.

$\lambda : S \rightarrow Y$: An output function which maps the internal agent state to the output set. Output events can only be generated at the time of internal transition.

$\text{ta} : S \rightarrow \text{Time}$: This represents the time the agent stays in a particular state before transiting to the next sequential state.

Cryptographic protocols are designed to establish and authenticate communication between entities. Entities in cryptographic protocols are formally called principals and are assumed to have unique identities. A good number of cryptographic protocols require an authentication server or a certification authority to provide keys and certificates to enable secure communication between two or more principals. These properties and more are clearly identifiable in agents and to describe these properties in DEVS, the formalism for a coupled model needs to be introduced. The coupled model describes how to integrate all three agents as identified in the DSE protocol (discussed in subsequent sections) forming a larger model as shown below.

$$\Pi_{\text{AgentS}} = (X, Y, M, Y_{\text{eic}}, Y_{\text{eoc}}, \alpha, \text{select})$$

$X = \{x_1, x_2, \dots, x_n\}$ is a non empty set of inputs to the coupled model agent S.

$Y = \{y_1, y_2, \dots, y_n\}$ is a non empty set of outputs to the coupled model agent S.

$M = \{m_1, m_2, \dots, m_n\}$ is a non empty set of unique component references.

$Y_{\text{eic}} \subseteq \Pi_{\text{AgentS}}.\text{input} * \Sigma_{\text{Agent}}.\text{input}$: An external input coupling relation.

$Y_{\text{eoc}} \subseteq \Sigma_{\text{Agent}}.\text{output} * \Pi_{\text{AgentS}}.\text{output}$: An external output coupling relation.

$\alpha \subseteq \Sigma_{\text{Agent}}.\text{output} * \Sigma_{\text{Agent}}.\text{input}$: An internal coupling relation.

$\text{select} : 2^M \rightarrow M$: Tie breaking selector.

A multiple state transition is one of the problems associated with coupling concurrent and sequential components in discrete simulation systems. This may lead to instability if it occurs at the same simulation time. In order to deal with this problem a selection criteria is defined, which determines the component's transition that is priority. This is also applicable to agents described in the simulation model where select represents a tiebreak. Select chooses a unique agent from any non-empty subset E of M where E corresponds to the set of all agents having simultaneous state transitions.

4. THE SIMULATION MODEL

In order to explain the development of the model using the agent definition described above, a simple cryptographic protocol simulation model is presented. The model is intended to convey the session key K_{ab} and data, from agent A to agent B whilst keeping it secret from other agents on the network.

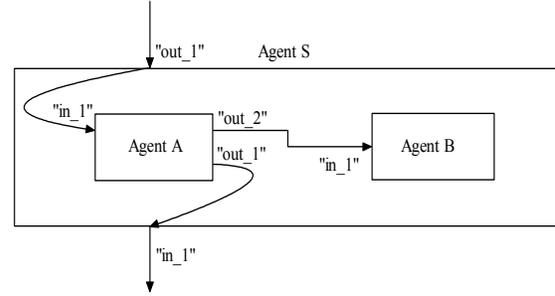


Figure 1: A Simple Cryptographic Protocol Model

Agent A makes contact with agent S, who provides A with the session key K_{ab} and a secret containing the session key K_{ab} but encrypted with B's key. Agent A then sends the secret to agent B who then decrypts the secret and stores the session key. Figure 1 shows the input and output ports of agent A. The input port "in_1" of agent A is for receiving messages from agent S. The output port "out_2" is used for sending the messages containing the session key K_{ab} to agent B and output port "out_1" of agent A is for making initial contact with agent S. A formal description of the model is specified using the atomic DEVS as shown below.

$$\Sigma_{\text{AgentA}} = (X, S, Y, \delta_{\text{int}}, \delta_{\text{ext}}, \lambda, \text{ta})$$

$X = \{\text{"in_1"}\}$

$Y = \{\text{"out_1"}, \text{"out_2"}\}$

$S = \{\text{idle, sent, recvd, acct}\}$.

$\delta_{\text{int}}(\text{idle}) = (\text{make_INI_REQ, sent})$

$\lambda(\text{idle}) = \text{"out_1"} = \text{msg}$

$\delta_{\text{ext}}((-,-), \text{"in_1"}) = (\text{recv_MSG}, \text{recvd})$
 $\delta_{\text{int}}(\text{cond} \neq \text{False}, \text{accpt}) = (\text{send_MSG}, \text{accpt})$
 $\lambda(\text{cond} \neq \text{False}, \text{accpt}) = \text{"out_2"} = \text{msg}$
 $\text{ta}(S) = \text{time value}$

The notion of time cannot be effectively predicted due to factors such as network latency, bandwidth, encryption/decryption algorithms etc. The variable msg is an address location used to store received messages from agent S and cond is a conditional variable, which indicates success or failure of processed messages. Messages are transmitted either as plain text or cipher text depending on the protocol being simulated.

In figure 2, a state trajectory is given for the agent A model. It shows that the agent made an internal transition from idle to sent state. The agent remains in the idle state until the time $\text{ta}(\text{idle})$ elapses. When this occurs an output $y_1 = \lambda(\text{idle})$ is generated and the state is transitioned to the sent state. The agent remains in this autonomous mode until it receives an external event. This event does not give rise to an output.

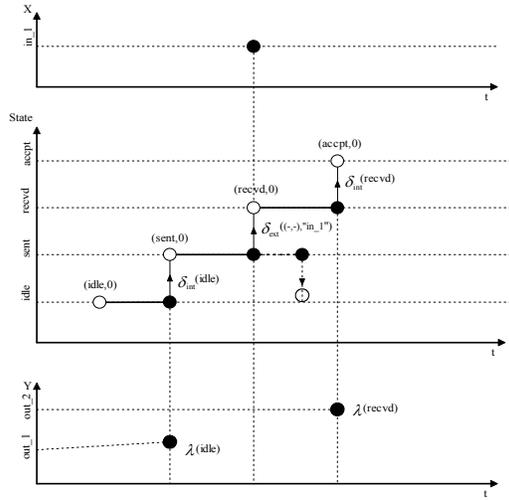


Figure 2: State Trajectory of Agent A

Once the agent A and agent B have been developed, the coupled agent S can be specified defining the required coupling relationship between all the agents in the model. The agent S model is formalised as follows.

$$\Pi_{\text{AgentS}} = (X, Y, M, \Upsilon_{\text{eic}}, \Upsilon_{\text{eoc}}, \alpha, \text{select})$$

$X = \{\text{"in_1"}\}$
 $Y = \{\text{"out_1"}, \text{"out_2"}\}$
 $M = \{\text{Agent A}, \text{Agent B}\}$
 $\Upsilon_{\text{eic}} = \{(\text{Agent S.out_1}, \text{Agent A.in_1})\}$
 $\Upsilon_{\text{eoc}} = \{\text{Agent A.out_1}, \text{Agent S.in_1}\}$

$\alpha = \{\text{AgentA.out_2}, \text{AgentB.in_1}\}$
 $\text{select} : (\text{Agent A}, \text{Agent B}) \rightarrow \text{Agent A}$

5. NEEDHAM-SCHROEDER PROTOCOL

This protocol is the basis of many existing protocol designs today. It implements a symmetric mechanism and shares the common problem of key distribution. Here the client A makes the initial contact with the server S by sending a message consisting of its identity, the identity of client B and a randomly generated number N. The server S randomly generates a session key, which is shared between clients A and B and then encrypts a message containing the shared session key, the identity of client A with the session key it shares with client B.

Following from that the server encrypts another message containing the shared session key, the identity of client B, the random number generated by client A and an embedded encrypted message. The server eventually sends the encrypted message to client A, who with the knowledge of the server's shared key is able to decrypt the message. Client A subsequently sends the embedded encrypted message to client B, who is also able to decrypt it. Figure 3 illustrates the simulation model in the context of agent based framework. $N_b - 1$ in message 5 implies that the message is from A and not from B [Burrows et al, 1990].

message 1 A \rightarrow S: A, B, N_a
 message 2 S \rightarrow A: $\{N_a, B, K_{ab}, \{K_{ab}, A\}_{K_{bs}}\}_{K_{as}}$
 message 3 A \rightarrow B: $\{K_{ab}, A\}_{K_{bs}}$
 message 4 B \rightarrow A: $\{N_b\}_{K_{ab}}$
 message 5 A \rightarrow B: $\{N_b - 1\}_{K_{ab}}$

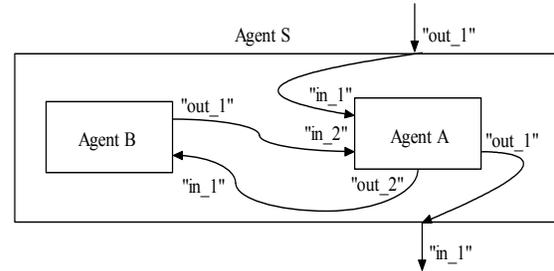


Figure 3: Needham-Schroeder Protocol Model

$$\Pi_{\text{AgentS}} = (X, Y, M, \Upsilon_{\text{eic}}, \Upsilon_{\text{eoc}}, \alpha, \text{select})$$

$X = \{\text{"in_1"}, \text{"in_2"}\}$
 $Y = \{\text{"out_1"}, \text{"out_2"}\}$
 $M = \{\text{Agent A}, \text{Agent B}\}$
 $\Upsilon_{\text{eic}} = \{(\text{Agent S.out_1}, \text{Agent A.in_1})\}$
 $\Upsilon_{\text{eoc}} = \{\text{Agent A.out_1}, \text{Agent S.in_1}\}$
 $\alpha = \{\text{AgentA.out_2}, \text{AgentB.in_1}\}$
 $\alpha = \{\text{AgentB.out_1}, \text{AgentA.in_2}\}$
 $\text{select} : (\text{Agent A}, \text{Agent B}) \rightarrow \text{Agent A}$

6. DSE PROTOCOL

The DSE protocol is based on shared and public key cryptography and is suitable for authenticating federates in HLA coupled distributed synthetic environment. Any number of federates can join or resign from the federation securely, hopefully, affecting the performance of the scheme minimally. The protocol is based on the plug and adaptor concept where plugs are attached to each federate model and the adaptor is attached to the RTI. This provides a platform for coordinated and secure communication between federates participating in the federation exercise. The adaptor S serves as an authentication server and a certificate authority providing various services to each federate model. Amongst other strengths, the protocol is designed to guard against a replay attack using synchronised clocks and randomly generated numbers.

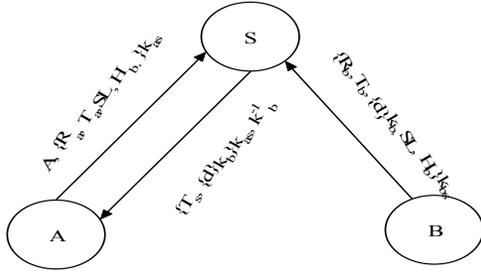


Figure 4: The DSE Authentication Protocol Structure

The description of the protocol is given below, with two federates attached to Plugs A and B as shown in figure 4. K_{as} and K_{bs} represent the shared keys for Plugs A and B, K_b^{-1} and K_b represent public and private keys for Plug B. R_a and R_b are random numbers generated by federates A and B. T_s , T_a and T_b are timestamps of Plugs A, B and adaptor S. SL represents the security level and finally d is the data in the form of updates. For the sake of clarity, the federate attached to Plug A will be referred to as federate A and the federate attached to Plug B as federate B. Federate B sends an encrypted message consisting of R_b , T_a and the object handle of the instance whose attributes need to be updated with the new attribute value d . If federate A subscribed to this attribute, once it has been registered and updated by B, it sends its identity along with an encrypted message consisting of a randomly generated number R_a , a timestamp T_a , and the object handle of the instance whose attributes have been updated to S. S confirms that the message is timely, and R_a checked against existing random generated numbers. S then generates a timestamp T_s , and forwards both the encrypted attribute values (data) and the certificate of B containing the public key of B to A who then extracts and verifies the public key. If the process is successful, the message

is decrypted and the data is reflected and updated as shown in figure 4.

message 1 $B \rightarrow S : B, \{R_b, T_b, \{d\}_{k_b}, SL, H_b\}_{k_{bs}}$
 message 2 $A \rightarrow S : A, \{R_a, T_a, SL, H_b\}_{k_{as}}$
 message 3 $S \rightarrow A : \{T_s, \{d\}_{k_b}\}_{k_{as}}, K_b^{-1}$

6.1 DSE Protocol Simulation Model

In this protocol agent B makes contact with agent S, if successful data is transferred. Agent A wishes to have access to the data transferred by agent B. To do so agent A must make contact with agent S passing on its identity and authentication details, which are verified. If successful agent S responds with the requested data encrypted with the shared key. The input and output ports of the agents are shown in figure 5 and the formal description given below.

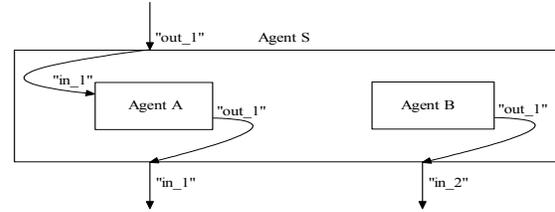


Figure 5: DSE Simulation Protocol Model

$$\Sigma_{AgentA} = (X, S, Y, \delta_{int}, \delta_{ext}, \lambda, ta)$$

$X = \{\text{"in_1"}\}$
 $Y = \{\text{"out_1"}\}$
 $S = \{\text{idle, sent, recvd, acpt}\}$
 $\delta_{int}(\text{idle}) = (\text{make_INI_REQ}, \text{sent})$
 $\lambda(\text{idle}) = \text{"out_1"} = \text{msg}$
 $\delta_{ext}((-, -), \text{"in_1"}) = (\text{recv_MSG}, \text{recvd})$
 $ta(S) = \text{time value}$

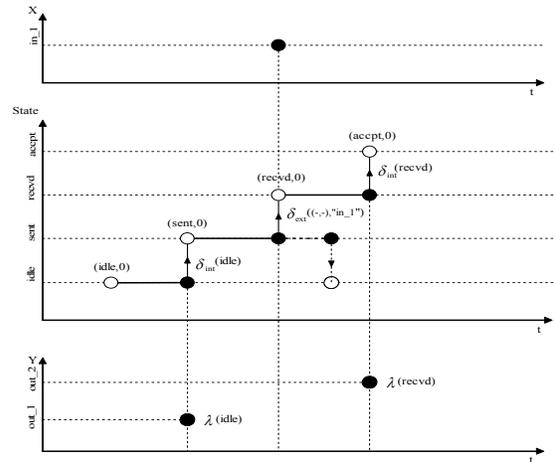


Figure 6: State Trajectory of Agent A in the DSE Model

The trajectory of agent A is shown in figure 6. Similarly, agent B can be developed and shown trivially. Once agent A and agent B have been

developed, the coupled model agent S can be specified. The agent S model formalisation is shown below.

$$\prod_{\text{AgentS}} = (X, Y, M, Y_{\text{eic}}, Y_{\text{eoc}}, \alpha, \text{select})$$

$X = \{\text{"in_1"}, \text{"in_2"}\}$
 $Y = \{\text{"out_1"}\}$
 $M = \{\text{Agent A, Agent B}\}$
 $Y_{\text{eic}} = \{(\text{Agent S.out_1, Agent A.in_1})\}$
 $Y_{\text{eoc}} = \{\text{Agent A.out_1, Agent S.in_1}\}$
 $\alpha = \{\text{AgentB.out_1, AgentS.in_2}\}$

6.2 Prototyping the Agent Model

The Simplex3 simulation system introduced by Schmidt [Schmidt, 2001] has been used to achieve the dynamics of the DEVS models. Each agent comprises various parts. The name, declaration and the dynamic part all make up the agent composition. Firstly, agents are identified by their names. The quantities and their properties are defined within the declaration part and the behaviour of the agent is described in the dynamic part. Agent behaviour could be modelled using differential equations, events or algebraic equations. However, in this case the behaviour of the agent is modelled using events.

6.3 High Level Component Agent_SIM

Three high level components have been defined, two for the atomic DEVS and the other for the coupled DEVS. Within the Simplex3 system, components can be composed of subcomponents and linked via connectors. Each agent is represented as an independent basic component and linked together in a high-level component Agent SIM and described using the Simplex3 model description language. The Agent_SIM model describes a cycle in which agents can send and receive messages via input and output channels. On the start of the simulation, agent B creates and sends a message carrying data updates to agent S who then verifies and accepts the update.

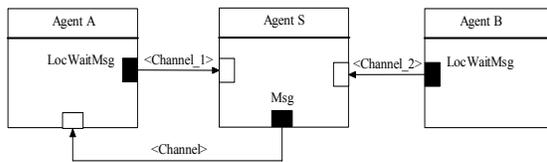


Figure 7: Relationship between Basic Agent Components

Agent A subscribes to data by sending a message comprising security information to agent S and if successful the data requested is sent to agent A. The dynamic behaviour of each agent describes the various events which can take place during the simulation run. Figure 7 shows the basic component agent A, agent B and agent S all linked together using component connections. Figure 8

shows the high level component connection representation of the agents. Messages are modelled using a mobile message component, as shown in figure 9, where a number of attributes are declared.

```

1: HIGH LEVEL COMPONENT Agent_SIM
2: SUBCOMPONENTS
3:   Agent_A,
4:   Agent_B,
5:   Agent_S
6: COMPONENT CONNECTIONS
7:   Agent_B.LocWaitMsg --> Agent_S.Channel_2;
8:   Agent_A.LocWaitMsg --> Agent_S.Channel_1;
9:   Agent_S.Msg --> Agent_A.Channel;
10: END OF Agent_SIM

```

Figure 8: High Level Component Agent_SIM

```

1: MOBILE COMPONENT Message
2: DECLARATION OF ELEMENTS
3: STATE VARIABLES
4: DISCRETE
5:   timestamp (REAL) :=0, # timestamp
6:   randomnum (REAL) :=0, # random generated number
7:   data (REAL) :=0, # data updates
8:   handle (INTEGER) :=0, # data handle
9:   identity (INTEGER) :=0 # federate identity
10: END OF Message

```

Figure 9: Mobile Component Message

7. EVALUATING THE PROTOCOL DESIGN

When examining the security of protocols, it is assumed that the underlying cryptographic mechanisms are secure. In evaluating the protocol an intruder does not necessarily have to attack the underlying mechanism directly but rather attempt to subvert the protocol's objective by defeating the manner in which such mechanisms are combined. Agents could be simulated with this capability and other known security breaches. This could provide for an effective security assessment of the protocol simulated.

The aim of the attack model is to provide some form of benchmark for testing the security of a protocol with the hope of uncovering any potential failures in the design. It is intended that the attack model is capable of performing a number of attack scenarios which include, first, the cases where the attack model is able to send messages but not read messages that are not addressed to it. Second, the attack model is able to send and read messages but not block messages and the last but not the least, the attack model could send, read and block messages with other messages etc. Other capabilities that could be included in the attack model include the ability to break certain classes of cryptosystems. The overall attack model would provide the basis for simulating various attack scenarios and thus could reveal the requirements for a more secure protocol design.

7.1 Attack Model Description

In this section, an abstract simulator is considered. It describes some of the capabilities of the attack model for example impersonation and message interception where the attack agent attempts to play the role of the sender or the receiver as well as modify message content. Attack actions could be defined as events engaged by an attacker, which affect messages sent and received by legitimate participants in the protocol. Hence, an attack capability could be defined as a set of actions an attacker is able to perform. These actions and events are described in the model. The simulation starts when the model receives the SIMULATE message shown below, and stops when $t_{\text{Attack}_{\text{agent}}} = \infty$

```
when receive (SIMULATE, t)
send (START, t) to agentattack
while ( $t_{\text{Attack}_{\text{agent}}} \neq \infty$ ) do
  "Intercept messages transmitted"
  send (MSG,  $t_{\text{Attack}_{\text{agent}}}$ ) to agentattack
  "Message modification"
  send (MSG,  $t_{N_{\text{child}}}$ ) to agents
endWhile
```

The attack simulator is necessary to drive the model. The variables t_L and t_N hold the time of last time, and the time for the next transition. This method sets the partial state to $s\{o\}$ and the value $e\{o\}$ is interpreted as the time elapsed in the current state.

```
when receive (START, t)
   $t_L \leftarrow t \leftarrow e\{o\}$ 
   $s \leftarrow t \leftarrow s\{o\}$ 
   $t_N \leftarrow t_L \leftarrow ta\{s\}$ 
end
when receive (MSG, t)
  if  $t \neq t_N$  then return endif
  intercepted  $\leftarrow$  message t  $\leftarrow$  MSG
   $s \leftarrow \delta(s)$ ,  $s \leftarrow t \leftarrow s\{o\}$ 
   $t_L \leftarrow t$ ,  $t_N \leftarrow t_L \leftarrow ta\{s\}$ 
end
when receive (MSG, t)
  if  $t \neq t_N$  then return endif
  message t  $\leftarrow$  MSG
   $s \leftarrow \delta(s)$ ,  $s \leftarrow t \leftarrow s\{o\}$ 
   $t_L \leftarrow t$ ,  $t_N \leftarrow t_L \leftarrow ta\{s\}$ 
end
```

8. SUMMARY

A possible approach, which utilises the merits of both agent-based and simulation technologies for analysing cryptographic protocols has been proposed. This approach is based on simulating an environment appropriate for describing the DSE protocol as well as other known protocols. The environment allows agents to interact amongst themselves and also react to external activities such

as an intruder attack. In addition, an attack model was described and introduced, with a number of attack capabilities, in the simulation environment to provide a test bed for examining security flaws in the protocol simulation.

9. REFERENCES

- Baillie P. 2002. "An Agent with a Passion for Decision Making." In Proc. of Agents in Simulation Workshop III Passau, Germany, University of Passau, April.
- Burrows M, Abadi M, and Needham R. 1990. "A Logic for Authentication" SCR Research Report 39, Digital Equipment Corporation, February.
- Gong L, Needham R and Yaholm R. 1990. "Reasoning About Belief in Cryptographic Protocols." In Proceeding of IEEE Symposium on Research in Security and Privacy, Pages 234-248, Oakland California.
- Iwu, F.O. and Zobel R.N. 2002. "Network Attack Profiling: Using Agents-Based Simulation to Gather Forensic Information" In Proc. of Agents in Simulation Workshop III Passau, Germany, University of Passau, April.
- Nieh B and Tavares S. 1992. "Modelling and Analysing Cryptographic Protocols using Petri Nets." In Proceedings of AUSCRYPT.
- Schmidt B. 2002. "How to give Agents a Personality." In Proc. Of Agents in Simulation Workshop III Passau, Germany, University of Passau, Passau.
- Schmidt B. 2001. "The Art of Modelling and Simulation: Introduction to Simulation Systems Simplex3." Book, SCS-European Publishing House Erlanger.
- Varadarajan V and Shankaran R. 1990. "The use of Formal Description Technique in the Specification of Authentication Protocols." In Proceedings of Computer Standards and Interfaces.
- Zeigler B. and et al. 2000. "Theory of Modelling and Simulation." Book, Academic Press, Inc. January.

10. BIOGRAPHY

FRANTZ IWU is a research associate at the University of York. He obtained his MSc in Advanced Computer Science and has recently completed his PhD studies at Manchester University. He is a member of the British Computer Society.

RICHARD ZOBEL He is a former Chairman of the United Kingdom Simulation Society (UKSim), Former Secretary of the European Federation of Simulation Societies (EUROSIM), and is a European Director of SCSi, the Society for Computer Simulation International. His current research work concerns distributed simulation for non-military applications, issues of verification and validation of re-useable simulation models and security for distributed simulation under commercial network protocols. He is now semi retired, but remains very active.

A TIME SLICING APPROACH TO EXTERNAL WORKLOAD MANAGEMENT ON BSP TIME WARP

MALCOLM YOKE HEAN LOW
Singapore Institute of Manufacturing Technology
71 Nanyang Drive, Singapore 638075
E-mail: yhlow@SIMTech.a-star.edu.sg

ABSTRACT

The performance of a BSP Time Warp parallel simulation system on a large-scale cluster of workstations can be severely affected due to the presence of external workload on individual machine in the cluster. This paper describes a new approach to managing external workload for BSP Time Warp parallel simulation on a cluster of workstations using the approach of time slicing. Experimental results comparing the performance of this new approach and the one proposed previously show that the new approach is resilient to interruption from external workload on multiple computing nodes in the cluster of workstations.

1 INTRODUCTION

Parallel simulation is an emerging technology that enables the execution of large-scale simulation model in a shorter timeframe compared to its sequential counterpart. Many of the existing parallel simulation protocols are developed with the assumption that the underlying parallel computing platform is dedicated and thus most do not consider the factor of variation in system load of the computing platform due to interruption from external workload.

The increasing popularity of large-scale cluster of workstations as the execution platform for parallel simulation requires a new approach in the design of parallel simulation protocols. Very often computing resources in these clusters are not dedicated and are usually shared among multiple users. The workload on each computing node in the cluster can fluctuate widely due to the presence of jobs from other users.

In [6, 7], we have reported our initial effort in designing a dynamic load-balancing (DLB) algorithm for the Bulk Synchronous Parallel Time Warp (BSP-TW) with external load-management capability. The external load-management module described in [7] uses the approach of evicting simulation workload from a processing node whenever the system load of the processing node exceeds a load threshold parameter. The drawback of this approach is that the performance of the system deteriorates rapidly in the presence of external workload on multiple nodes of the cluster of workstations. It is also difficult to determine the optimal values for the load threshold parameter.

In this paper, we present a new approach to external workload management using time slicing. Our experimental results show that the new approach is able to maintain high

performance in the presence of external workload on multiple nodes of the cluster of workstations without the need to use the load threshold parameter.

The rest of this paper is organized as follows. Section 2 describes the BSP model and the BSP Time Warp optimistic protocol. In section 3, the BSP-TW DLB_{ccl} algorithm proposed in [7] is described. We then describe the new BSP-TW DLB_{ccls} algorithm which uses a time slicing approach to external workload management in section 4. Section 5 presents experimental results comparing the new DLB algorithm with the existing one on a manufacturing simulation model. Some related work are described in section 6. Section 7 summarizes the paper and outlines future research directions.

2 BSP TIME WARP

The BSP model first proposed in [9] is designed to be a general purpose approach to parallel computing that allows the separation of concerns between computation, synchronization and communication costs. It has a simple cost model for predicting the performance of BSP algorithms on different parallel platforms. A BSP programming model consists of P processors linked by an inter-connecting network and each with its own pool of memory.

A BSP algorithm consists of a set of processors each executing a series of supersteps. Each superstep consists of three ordered phases: 1) a local computation phase, where each processor can perform computation using local data and issue communication requests; 2) a global communication phase, where data is exchanged between processors according to the requests made during the local computation phase; and 3) a barrier synchronization, which waits for all data transfers to complete and makes the transferred data available to the processors for use in the next superstep.

The BSP-TW algorithm [8] shown in Figure 1 is designed to be an efficient realization of an optimistic synchronization protocol ([3], [4]) on the BSP model. Each processor manages a group of logical processes (LPs) in the system. In BSP-TW, LPs are also referred to as simulation objects and the two terms are used interchangeably in this paper. LPs in the same processor share a common event-list. A series of supersteps are executed by each processor as indicated by the outer `while` loop and the `bsp_sync()` statement at the end of the loop.

The global virtual time (GVT) measures the progress of a simulation run. An estimate of GVT is computed after every

```

bsp_begin();
[A] Initialization;
while GVT < SimEndTime do
  [B] Receive external events and process rollback;
  [C] Compute new GVT, perform fossil collection and
      compute new event limit  $n_e$  every  $n_g$  supersteps;
  [D] Execute  $n_e$  events;
  bsp_sync();
endwhile
bsp_end();

```

Figure 1. Algorithm for BSP Time Warp.

n_g supersteps; n_g is also known as the GVT update interval. The body of the loop terminates when the GVT value is greater than the simulation end time.

The algorithm provides an automatic means of throttling the number of events, n_e , being simulated per superstep based on statistics from fossil collected events. The BSP cost model for a BSP-TW algorithm S can be expressed as

$$\text{cost}(S) = \sum_{i=1}^{n_s} (w(i) + gh(i) + L) \quad (1)$$

where n_s is the total number of supersteps; $w(i)$ is the computation cost for superstep i ; and $h(i)$ is the maximum number of messages sent or received respectively by any processor in superstep i . The architecture dependent parameters g and L represent the communication and synchronization costs respectively.

From the BSP cost model, we can see that the performance of a BSP-TW algorithm relies on three factors: a) computation balance; b) communication balance; and c) n_s , the total number of supersteps. Computation and communication imbalance can result from the dynamic changing nature of the workload of the simulation model and interruption from external workload. The total number of supersteps required to complete the simulation depends on the lookaheads on the links between LPs on different processors. Lookahead is defined as the minimum simulation time interval between event arrival, from the source to a destination LP. A dynamic load-balancing algorithm can reduce both computation and communication load-imbalance, as well as optimize lookaheads by migrating simulation objects between processors.

3 MANAGING EXTERNAL WORKLOAD BY EVICTING PROCESSORS

The BSP-TW DLB_{ccl} algorithm first described in [6] has facilities to dynamically balance computation and communication load-imbalance, as well as optimize lookaheads between processors. However, the algorithm does not take into account interruption from external workload. In [7], we proposed an extension to the BSP-TW DLB_{ccl} algorithm, referred to as BSP-TW DLB_{ccl_e} algorithm, to allow external workload management.

```

bsp_begin();
[A] Initialization;
while GVT < SimEndTime do
  [B] Receive external events and process rollback;
  [C] Compute new GVT, perform fossil collection and
      compute new event limit  $n_e$  every  $n_g$  supersteps;
  [D] After each  $\lambda$  GVT computation:
    [D0] balance_extLoad();
    [D1] balance_computation();
    [D2] balance_communication();
    [D3] optimize_lookahead();
  [E] Execute  $n_e$  events;
  bsp_sync();
endwhile
bsp_end();

```

Figure 2. Algorithm for BSP-TW DLB_{ccl_e}.

3.1 BSP-TW DLB_{ccl_e} Algorithm

Figure 2 shows the pseudo-code for the BSP-TW DLB_{ccl_e} algorithm. The BSP-TW DLB_{ccl_e} algorithm consists of four modules and is executed at each migration point, which occurs every λn_g supersteps ($\lambda \geq 1$). We also refer to the λn_g supersteps between two migration points as a migration interval. The pseudo-code for the BSP-TW DLB_{ccl} algorithm is not shown here as it is essentially BSP-TW DLB_{ccl_e} without module D0.

At each migration point, one of the four modules will be activated based on factors such as the amount of external workload, computation imbalance and communication imbalance.

The computation load-balancing in module D1 is carried out by transferring simulation objects from processors with high computation workload to processors with low computation workload. For module D2, communication load-balancing is carried out by exchanging simulation objects between processors. The module uses load exchange, rather than load transfer to preserve the computation balance achieved in module D1, at the same time improving the balance in communication workload. The lookaheads optimization in module D3 is carried out by merging simulation objects with small lookaheads into the same processor. For more detailed explanation of these three modules, readers are referred to [6].

The BSP-TW DLB_{ccl} algorithm described in [6] is enhanced with module D0 in order to handle computation and communication load-imbalance due to the presence of external workload. The pseudo-code for module D0 is shown in Figure 3.

The state variable $P_i.la$ is used to track the average system load of processor P_i . We classify the set of processors with average load greater than the processor load threshold parameter, θ , as heavily loaded. The average load of a processor is obtained by a UNIX system call `getloadavg()`. This system call returns the number of processes in the system run

```

balance_extload()
  foreach processor  $P_i$  do
    if  $P_i.loadavg > \theta$  then
       $migrate\_all(P_i)$ ;
      set  $P_i$  as inactive;
    else if  $P_i.la < \frac{\theta}{2}$  then
      set  $P_i$  as active;
    endif
  endfor

```

Figure 3. Algorithm for Balancing External Workload.

queue averaged over various periods of time. The one minute sample returned by the system call is used in the experiments.

At each migration point, the BSP-TW DLB_{ccl}e algorithm attempts to evict all the simulation objects out of these heavily loaded processors. The method $migrate_all(P_i)$ evicts all the simulation objects in processor P_i to other processors with normal workload in a round-robin fashion. The status of processor P_i is then set to inactive. As the dynamic load-balancing modules D1 to D3 only consider the set of active processors, simulation objects will not be migrated back to the processors that are still heavily loaded with external workload. When a previously heavily loaded processor's average system load drops below $\frac{\theta}{2}$, the status of the processor is reset to active. This causes the computation and communication load-balancing modules to detect the idle processor and allows simulation objects to be moved back to it.

4 MANAGING EXTERNAL WORKLOAD BY TIME SLICING

Although the BSP-TW DLB_{ccl}e protocol does solve the problem of external workload interruption, it sacrifices the complete use of a processor whenever it is loaded with external workload, regardless of the amount of external workload in the processor. Also, the performance of BSP-TW DLB_{ccl}e depends largely on how θ is set. If the value of θ is set too low, many processors may be evicted due to the presence of very small external workload. If the value of θ is set too high, the BSP-TW DLB_{ccl}e algorithm may not react effectively to the presence of external workload.

In this section, we consider another approach to managing external workload by considering the available time slice for the BSP process on the heavily loaded processors, rather than leaving the processors completely out of the parallel computation.

4.1 Example of External Workload Management using Time Slicing

We first illustrate our approach using the example shown in Figure 4. The figure shows the computation workload of a superstep for eight processors. Processors P0 to P3 are each loaded with two external workloads, indicated by the

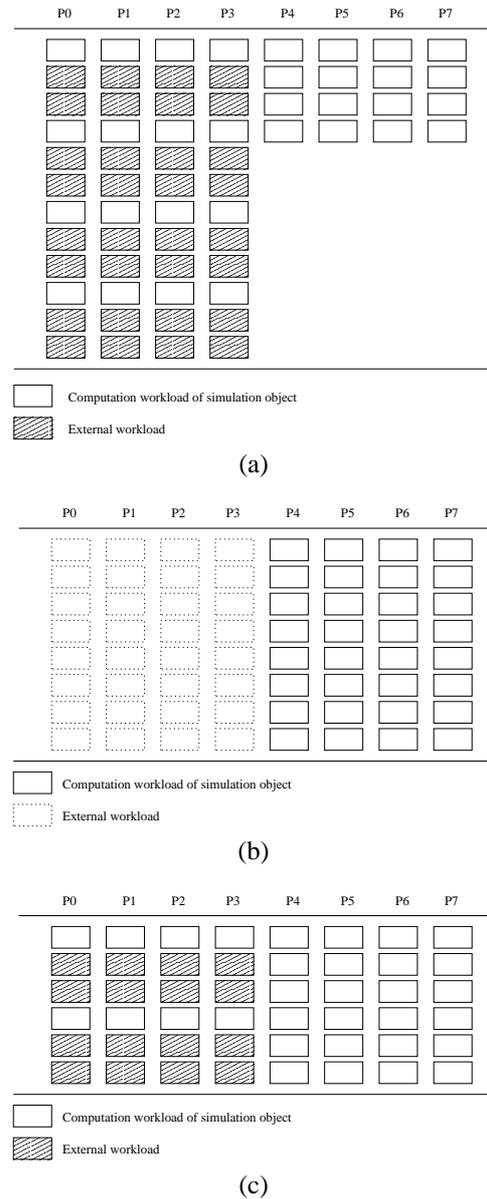


Figure 4. An Example of External Workload Management using Time Slicing.

shaded boxes. The computation workload of simulation objects on all the eight processors in the superstep are the same, as shown by the white boxes. Each white box can be considered the computation workload of a simulation object.

Due to the presence of external workload, the superstep on processors P0 to P3 takes three times the amount of time to complete, as compared to those on processors P4 to P7. We can also say that the simulation workload is only given one-third slice of the CPU processing time. If we assume that each box (white or shaded) consumes one unit of CPU processing time, the superstep takes 12 units of CPU processing time.

Figure 4b shows the workload configuration using the BSP-TW DLB_{ccl}e algorithm. All the simulation objects are evicted from the four loaded processors and distributed to processors P4 to P7. The resulting configuration is such that

processors P0 to P3 each can complete the superstep with minimum delay while processors P4 to P7 now have twice the amount of workload to process. The CPU processing time for this superstep is reduced to 8 time units.

Another approach to managing the workload is to consider the fact that the BSP workload on the heavily loaded processors still have access to one-third slice of the CPU processing time. We can migrate parts of the simulation objects out of these processors so that the overall workload for all processors (taking into account the external workload) after the migration is still balanced.

Figure 4c shows an example of how this is done. Two simulation objects are migrated out of each processor loaded with external workload. The resulting workload configuration is balanced across all processors. The superstep now requires only 6 units of CPU processing time.

The reason for the improvement over that using BSP-TW DLB_{ccl} is due to the use of the remaining one-third slice of CPU processing time on those heavily loaded processors to process part of the simulation objects' workload. The increased in workload of those processors not affected by external workload is reduced compared to that using BSP-TW DLB_{ccl} .

4.2 BSP-TW DLB_{ccls} Algorithm

We now describe the BSP-TW DLB_{ccls} algorithm that provides an alternative solution to managing external workload by considering the allocated CPU time slice for the computation workload in each processor in the system. The outline of the BSP-TW DLB_{ccls} algorithm is essentially the same as the BSP-TW DLB_{ccl} algorithm. Unlike the BSP-TW DLB_{ccl} algorithm which adds another module to the BSP-TW algorithm, the BSP-TW DLB_{ccls} algorithm works within the `balance_computation()` module. Also, it should be noted that the BSP-TW DLB_{ccls} algorithm does not require the use of the processor load threshold parameter θ .

Before we describe the modification to the module for balancing computation workload, we first need to resolve the condition for detecting imbalance in computation workload. For example, we would want to consider the workload configuration in Figure 4a as unbalanced while the configuration in Figure 4c as well-balanced. As the time taken to complete a superstep in each processor is computed by summing up the time taken for executing each event in the superstep, the `balance_computation()` module will instead consider the configuration in Figure 4a as well-balanced while the configuration in Figure 4c as unbalanced.

To resolve this problem, we can make use of the additional knowledge of the average system load of each processor ($P_i.la$) to work out a better approximation of the workload on each processor. We first scale the computation workload of each processor ($P_i.wl$) by its corresponding average system load as follows:

$$P_i.wl := P_i.wl * P_i.la . \quad (2)$$

```

balance_computation()
  while  $WB > \epsilon$  do
    let  $P_{max}$  be the processor with the max. computation workload;
    let  $P_{min}$  be the processor that yield the min. average workload;
    when paired with  $P_{max}$ 
       $x := \frac{P_{max}.wl * P_{max}.la - P_{min}.wl * P_{min}.la}{P_{max}.la + P_{min}.la}$ 
    computation_migrate( $x, P_{max}, P_{min}$ );
     $P_{max}.wl := P_{max}.wl - x$ ;
     $P_{min}.wl := P_{max}.wl$ 
    compute  $WB$ ;
  endwhile
  return  $flag$ ;

```

Figure 5. Algorithm for Determining Amount of Computation Workload to Migrate taking into account System Workload. ϵ is the Load-imbalance Threshold Parameter.

Note that for those processors with average system load less than 1.0, $P_i.la$ will be set to 1.0.

The calculation of the computation imbalance, WB , of the system is shown in equation (3).

$$WB = \frac{\max(P_i.wl) - \text{mean}(P_i.wl)}{\text{mean}(P_i.wl)}. \quad (3)$$

Using this formula, the load imbalance for the superstep shown in Figure 4a will be 0.5 while the superstep in Figure 4c will be treated as having perfectly balanced workload.

The BSP-TW DLB_{ccls} algorithm has exactly the same structure as that of BSP-TW DLB_{ccl} . The difference lies in the `balance_computation()` module, which now needs to take into consideration the system load of those processors involved in the load transfer process.

Figure 5 shows the pseudo-code for the `balance_computation()` module. Note that the computation workload used in the module have all been scaled by the average system load of individual processors. The processor P_{min} is not taken to be the one with the lowest computation workload, but rather the processor that will yield the lowest average workload when selected to engage in the load transfer process with the processor P_{max} that has the heaviest computation workload.

Figure 6 shows an example to illustrate why the processor that has the lowest computation workload is not chosen to be processor P_{min} . The example shows three processors and their respective computation workload for a superstep. We see that processor P0 is loaded with one external workload and processor P1 is loaded with four external workload. Although processor P2 is free from any external workload, its overall computation workload is still higher than processor P1.

Suppose processor P1 is now chosen to engage in the load transfer process with processor P0. One unit of computation workload will be migrated from processor P0 to processor

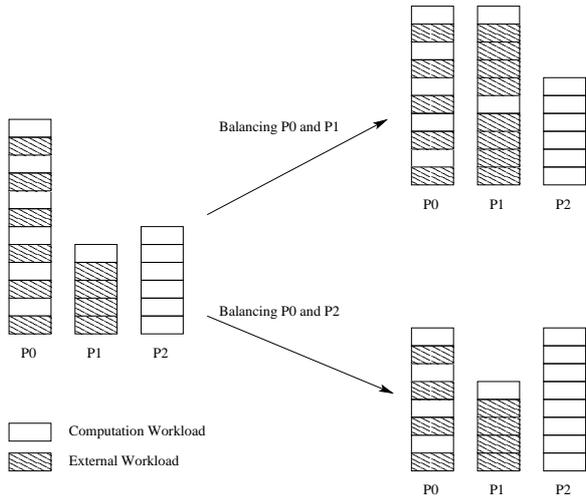


Figure 6. An Example to Illustrate the Selection of Processor P_{min} .

P1. This results in a net decrease of two units of computation workload in P0 and a corresponding five-unit increase in computation workload in P1. The overall improvement in the maximum computation workload is two units.

However, if processor P2 is chosen instead, two units of computation workload will be migrated from P0 to P2. This results in a net decrease of four units in computation workload in P0 and a corresponding two-unit increase in computation workload in P2. The overall improvement in the maximum computation workload in this case is four units. Although P2 is not the processor having the lowest computation workload, selecting it for the balancing process yields better performance compared to selecting P1, which has the lowest computation workload.

The presence of external workload on the individual processor and the scaling of computation workload for each processor requires some modifications to the formula used to compute the amount of workload to be transferred from processor P_{max} to P_{min} . The following formula computes the amount of computation workload, x , that needs to be migrated from processor P_{max} to processor P_{min} so that the resulting workload, y , on both processors after the migration is equal.

$$x = \frac{P_{max}.la(P_{max}.wl - P_{min}.wl)}{P_{max}.la + P_{min}.la} \quad (4)$$

$$y = P_{max}.wl - x \quad (5)$$

5 EXPERIMENTS WITH MANUFACTURING SIMULATION MODEL

In this section, we describe a set of experiments to compare the performance of BSP-TW DLB_{ccl}, BSP-TW DLB_{ccls}, and the BSP-TW DLB_{ccl} algorithms.

5.1 Simulation Model

The experiments are carried out using a manufacturing simulation model similar to that used in [5] to study different

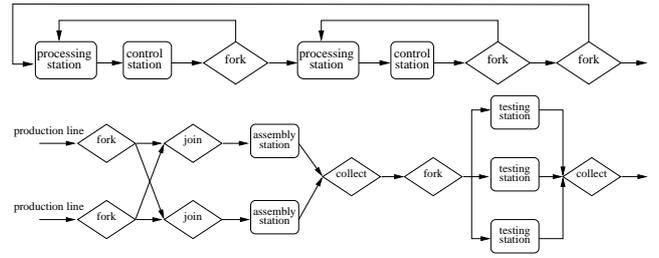


Figure 7. Layout of a Production Line and an Assembly and Test Facility.

runtime systems for a conservative simulation protocol. The manufacturing model consists of different entities of a typical production line with an assembly and test facility. Figure 7 shows the layout of a production line and an assembly and test facility. The configuration of the manufacturing model consists of a total of seven production lines. Each production line consists of 100 production stages. The assembly and test facility consists of 100 assembly stations and 100 testing stations. There are a total of 2417 simulation objects in this model.

This manufacturing model is a challenging model for optimistic simulation protocol such as BSP-TW due to the presence of many zero lookahead links on the `fork` and `join` nodes. Lookahead is crucial to the performance of the BSP-TW protocol since processors with many incoming communication links with small or zero lookaheads are likely to suffer from high event rollback rates.

For all the experiments, the GVT computation interval n_g is fixed at 50 supersteps. The migration interval λ is set to 5. A processor load threshold parameter of $\theta=1.5$ is used for the runs with BSP-TW DLB_{ccl}. The experiments are conducted on a cluster of eight 350MHz Sun UltraSparc workstations connected via a 100Mbps TCP/IP network. All execution times shown are the average of 10 runs. The simulation run length of all experiments is 10^4 time units. A block partition strategy is used to assign consecutive block of 25 simulation objects onto the same processor. The experiments are carried out by loading different number ($K=1, 2, 4$ and 6) of processors with different number ($N=1, 2, 3, 4$) of external workload. The external workload is introduced from the start of the simulation and lasts through the entire simulation duration.

5.2 Experimental Results

Table 1 shows the execution times using the three different protocols on the manufacturing model. The column under BSP-TW DLB_{ccls}* is executed using a modified version of BSP-TW DLB_{ccls}. This version uses a modified average system load for each processor, which is shown below:

$$P_i.la := (P_i.la)^2. \quad (6)$$

The modified system load of individual processor is then applied to the scaling of the computation workload in equation (2). This modification has no effect on those processors

		BSP-TW			
K	N	DLB _{ccl}	DLB _{ccl^e}	DLB _{ccl^s}	DLB _{ccl^s} *
1	1	602.0	489.4	551.2	520.6
	2	789.8	517.8	559.3	558.0
	3	844.3	566.1	617.9	569.6
	4	993.3	592.2	787.8	606.1
2	1	718.9	596.8	672.9	613.5
	2	1170.5	634.1	705.1	615.1
	3	1318.3	635.9	732.7	651.2
	4	1667.6	693.5	958.8	690.7
4	1	826.2	944.0	768.6	747.8
	2	1502.5	954.7	981.9	749.0
	3	1697.1	987.7	1055.7	767.6
	4	2329.2	1021.6	1275.6	840.5
6	1	918.9	1462.8	934.0	867.8
	2	1704.6	1736.5	1350.0	1038.4
	3	2090.9	1843.2	1449.8	1124.8
	4	2971.3	1870.6	1816.9	1128.7

Table 1. Execution Times (sec.) using BSP-TW DLB_{ccl^s} with N Number of Processors Loaded with K Number of External Workload.

with no external workload since $P_i.la$ will still be equal to 1.0. For those processors with average system load greater than 1.0, this change has the effect of encouraging the BSP-TW DLB_{ccl^s} to migrate more simulation objects out of them. Similarly, it also discourages the load-balancing algorithm from migrating simulation objects back into them.

Table 1 shows that for the cases with $N = 1$, the performance of BSP-TW DLB_{ccl^e} drops below BSP-TW DLB_{ccl} as the number of processors loaded with external workload is increased to six. The BSP-TW DLB_{ccl^e} algorithm is discarding six out of eight processors even though each of the six processors is only loaded with one external workload.

By not discarding completely those processors with external workload, the BSP-TW DLB_{ccl^s} protocol is able to achieve better performance than the BSP-TW DLB_{ccl^e} protocol for the cases with $N = 1$ and $K = 4$ and 6. For $K = 6$, the BSP-TW DLB_{ccl^s} algorithm outperforms BSP-TW DLB_{ccl^e} for all values of N .

However, as the number of processors loaded with external workload is reduced, the performance of BSP-TW DLB_{ccl^s} drops below that of BSP-TW DLB_{ccl^e}. This drop in performance in BSP-TW DLB_{ccl^s} can be attributed to two factors: 1) insufficient simulation objects are migrated out of those heavily loaded processors as the number of external workload on these processors is increased; and 2) side effects from the lookahead optimization module.

In order to verify the first hypothesis, we carried out the runs with BSP-TW DLB_{ccl^s}* to test if better performance can be achieved by encouraging more simulation objects to be migrated out of the heavily loaded processors. In a way, the squaring of the average system load of individual processor in equation (6) serves to exaggerate the load situation of those heavily loaded processors such that more simulation objects can be migrated out of them.

Table 1 shows that this approach does significantly improve the performance of the BSP-TW DLB_{ccl^s} algorithm. For the cases with $K = 4$ and 6, the performance of BSP-TW DLB_{ccl^s}* drops gradually with increasing external workload. This shows that the dismal performance of BSP-TW DLB_{ccl^s} is indeed due to insufficient simulation objects being migrated out of those heavily loaded processors.

However, for the runs with only one processor being loaded with external workload, the performance by either BSP-TW DLB_{ccl^s} or BSP-TW DLB_{ccl^s}* is still slightly worse than that using BSP-TW DLB_{ccl^e}. This performance drop can be attributed to the side effect of lookahead optimization.

Figure 8 shows a breakdown of the computation workload as well as the number of simulation objects on each processor for a run executed using BSP-TW DLB_{ccl^s}*. In this run, processor P1 is loaded with one external workload. We see that at superstep 1250, the balancing module is activated and the number of simulation objects on processor P1 drops from 307 to 170. Correspondingly, the computation workload for processor P1 decreases from a high level of 17.5 to 6.4.

However, at superstep 1500, an optimization of lookahead is carried out by the BSP-TW DLB_{ccl^s}* algorithm. This resulted in 20 simulation objects being migrated back into processor P1. The computation workload of processor P1 is increased to 11.5 in superstep 1750. At this point, the pattern repeats itself with the computation balancing module migrating simulation objects out of processor P1 and the lookahead optimization module migrating simulation objects back into processor P1. The main problem here is that the lookahead optimization module uses a migration threshold $\eta=0.5$ which allows up to 50% of the simulation objects to be migrated during each round of lookahead optimization and this tends to disrupt the load-balance achieved in the computation balancing process.

A possible solution to resolving this issue might be to use a value of η smaller than 0.5. While this will reduce the amount of simulation workload that can be migrated back into the heavily loaded processors, it will also slow down the lookahead optimization process on other processors, causing the performance to drop. An effective solution might require the value of η to be set differently for different processors with different load configurations. Further work will need to be carried out to explore this possibility.

6 RELATED WORK

Past studies of DLB algorithm for optimistic parallel simulation protocols have typically focused on which metrics to use to measure the actual workload of the system. In this paper, the metrics used are the computation workload together with the average system load of the individual nodes in the cluster of workstations.

In the study reported in [1], Carothers and Fujimoto presented an approach for background execution of Time Warp. The scheme allows a Time Warp system to execute in background and consume unused CPU cycles across a collection

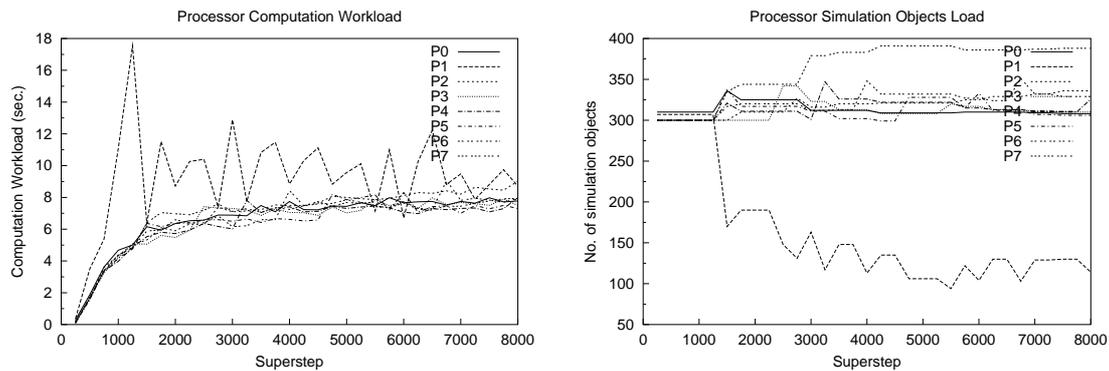


Figure 8. Processors Computation Workload and Number of Simulation Objects using BSP-TW DLB*_{ccls}.

of heterogeneous machines. The metric used is “Processor Advance Time” (PAT), which reflects the amount of real time needed to advance the virtual time of a logical process by one unit. A personal communication service network model is used in this study. The experimental results showed an improvement of up to 45% in the presence of external workload.

Glazer and Tropper also described a metric based on time slices [2]. They defined a time slice to be a metric proportional to the ratio of the amount of computation time required by a process over the advance of its simulation time. They presented speedup improvement ranging from 12% to 49% using this approach to balance simulation workload for three different simulation models running on a simulation multiprocessor environment. However, their experiments do not take into consideration interruption from external workload.

7 CONCLUSION

In this paper, we have described a new time slicing approach to external workload management for the BSP-TW parallel simulation protocol. Our experimental results comparing the performance of the BSP-TW DLB_{ccl}, BSP-TW DLB_{ccls} and BSP-TW DLB_{ccl} show that BSP-TW DLB_{ccls} protocol is able to achieve better performance over BSP-TW DLB_{ccl} when a high proportion of processors in the BSP-TW computation are burdened with external workload. However, the performance of BSP-TW DLB_{ccls} drops rapidly with increasing system workload on those heavily loaded processors. By amplifying the processor system workload to exaggerate the load-imbalance of the system, we show that the BSP-TW DLB_{ccls} protocol can indeed achieve significant performance improvement over both the BSP-TW DLB_{ccl} and BSP-TW DLB_{ccl} protocols.

ACKNOWLEDGEMENT

This work was carried out when the author was with the Oxford University Computing Laboratory. The work was supported by a postgraduate scholarship from the Singapore Institute of Manufacturing Technology (SIMTech).

REFERENCES

- [1] C.D. Carothers and R.M. Fujimoto. Background Execution of Time Warp Programs. In *Proceedings of the 10th Workshop on Parallel and Distributed Simulation (PADS'96)*, pages 12–19, Philadelphia, Pennsylvania, USA, May 1996.
- [2] D.W. Glazer and C. Tropper. On Process Migration and Load Balancing in Time Warp. *IEEE Transactions on Parallel and Distributed Systems*, 4(3):318–327, 1993.
- [3] D. Jefferson. Virtual Time. In *ACM TOPLAS*, volume 7, pages 404–425, 1985.
- [4] D. Jefferson and H. Sowizral. Fast Concurrent Simulation Using Time Warp Mechanism. In *Distributed Simulation 1995*, pages 63–69, La Jolla, California, USA, 1985. SCS-The Society for Computer Simulation, Simulation Councils, Inc.
- [5] C.-C. Lim, Y-H. Low, W. Cai, W.-J. Hsu, S.-Y. Huang, and S.J. Turner. An Empirical Comparison of Runtime Systems for Conservative Parallel Simulation. In *2nd Workshop on Runtime Systems for Parallel Programming (RTSPP 1998)*, pages 123–134, Orlando, Florida, USA, 30 March 1998.
- [6] M.Y.H. Low. Dynamic Load-Balancing for BSP Time Warp. In *Proceedings of the 35th Annual Simulation Symposium*, pages 267–274, San Diego, California, USA, 14-18 April 2002.
- [7] M.Y.H. Low. Managing External Workload with BSP Time Warp. In *Proceedings of the 2002 Winter Simulation Conference*, pages 704–711, San Diego, California, USA, 8-11 December 2002.
- [8] M. Marín. *Discrete-Event Simulation on the Bulk-Synchronous Parallel Model*. PhD thesis, Oxford University, November 1998.
- [9] L.G. Valiant. A Bridging Model for Parallel Computation. *Communications of the ACM*, 33:103–111, August 1990.

CENTRAL ISSUES AND CLASSIFICATIONS OF LOCATION MANAGEMENT TECHNIQUES IN WIRELESS AND MOBILE COMPUTING SYSTEMS

SEUNG-YUN KIM WALEED W. SMARI

*Department of Electrical and Computer Engineering
University of Dayton
300 College Park*

Dayton, OH USA 45469-0226

E-mail: kimseung@flyernet.udayton.edu, waleed.smari@notes.udayton.edu

Abstract: The rapid growth in mobile and wireless computing technology continues to present new challenges. Mobile users access information, independent of their location, through wireless and wired networks. In mobile computing, location management is introduced whenever users move from one place to another. In order to track a mobile user, the system must store information about his current location and report new locations to a home base station. Several techniques have been proposed to optimally manage the location of mobile hosts. In this paper, we present an overview of some principal issues, concepts and definitions used, and techniques proposed and developed for location management in mobile computing systems. The performance of these techniques is dependent on several parameters, such as the Call-to-Mobility Ratio (CMR). It also depends on update and routing costs, calls, and moves. We discuss different approaches introduced and assess their effectiveness. This will be followed by a survey of existing classifications of location management solutions. Finally, we introduce an alternative way of classifying these techniques in light of the central issues identified and in order to facilitate the development and design of a framework for these systems.

Keywords: Wireless and Mobile Computing, Location Management Techniques, CMR, Classification of LMTs.

1. INTRODUCTION

In mobile and wireless computing environments, mobile hosts may relocate from one cell location to another. In order to keep track of mobile hosts, the system must record and know the information about mobile hosts' current location. In recent years, several location management algorithms have been proposed to reduce the update and lookup costs accruing as a result of maintaining host information. An update occurs when a mobile host sends a message to update its stored location. A lookup occurs when it is required to locate a user each time a call is placed to that user or when a message is sent to her. The location updates and lookups are evaluated in terms of the number of messages sent, the size of messages, the distance the message needs to travel, the bandwidth consumed, the processing overhead and the delay incurred in answering locations queries. The main criterion used for efficient update is low signaling cost incurred by relocation of hosts between cells. The cost should be kept small enough not to affect network performance. This mobile communication network technology is expected to further develop to smaller cells for greater bandwidth sharing and reuse. The signaling load for location updates will be higher due to more frequent relocation for small cells [Hacacute and Liu, 1998]. A good location management scheme should attempt to optimize all of these parameters. Tracking mobile hosts and establishing efficient routing are basic functions of a mobile computing system. The system

needs to be updated and provided information about the location of mobile hosts regularly. Typically, the location area structure has several cells in it, and a mobile computing environment structure consists of several location areas, which may overlap with each other. Depending on the algorithm employed, an update will occur when the mobile host moves from one cell to another or from one location area to another location area.

A general mobile system consists of mobile hosts (MHs) that interact with a static network through fixed hosts, known as mobile base stations (MBSs). MBSs are augmented with a wireless interface, and they provide a gateway for communication between the wireless network and the static network. A mobile host can communicate with a mobile base station within a limited region around it. This region is referred to as a mobile base station's cell. Cells can have different sizes, and the average size of a cell is typically around 1 to 2 miles in diameter. A mobile host communicates with one mobile base station at any given time. An MBS is responsible for forwarding data between the mobile host and the static network. Due to mobility, a mobile host may cross the boundary between two cells while being active. Thus, the task of forwarding data between the static network and the mobile host must be transferred to the new cell's mobile base station [Krisha et al, 1996]. This process, known as *handoff*, is transparent to the mobile user. Handoff takes place when a mobile host moves from one cell to another

during a communication session. The information transmitted to the original mobile base station is easily forwarded to the new mobile base station through their common link. Note that handoffs and location management serve different purposes. The former is not required unless a communication session is in progress while a mobile host moves from one cell to another, whereas the latter is always required [Doley et al, 1996].

2. CALL-TO-MOBILITY RATIO

In a mobile computing environment, there are two important factors to consider: calls and moves. During a given time period, the number of calls and the number of moves determine how many data packets a mobile host is sent and how many movements the host has made, respectively. Classes of users are characterized by their call-to-mobility ratio (CMR) [Jain and Lin, 1995]. CMR is the average number of calls to a user per unit time, divided by the average number of times the user changes registration areas per unit time. We also define a local CMR (LCMR), which is the average number of calls to a user from a given originating signal transfer point per unit time. The LCMR can be used to relate the hit ratio to users' calling and mobility patterns directly. To do so, we need to make some assumptions about the distributions of the user's calls and moves. Let the call arrivals from an MBS to a user be a Poisson process with arrival rate λ , and the time that the user resides in a registration area be $1/\mu$. Then, LCMR can be expressed as: $LCMR = \lambda / \mu$.

Since we are dealing with non-negative random variables, it is convenient to associate with a probability density function, $f(s)$, its Laplace transform, $f^*(s)$. Let the residual time of a user at an RA be a random variable with a general density function f_m . Then, its Laplace transform

is given by $f_m^*(s) = \int_{t=0}^{\infty} f_m(t)e^{-st}dt$. Let t be the time

interval between two consecutive calls from the MBS to the user, and t_1 be the time interval between the first call and the time when the user moves to a new RA. From the random observer property of the arrival call stream [Feller, 1968], if call arrivals are Poisson distributed and $F(t)$ is an exponential distribution, the hit ratio of a user

calling can be given by $p = \int_{t=0}^{\infty} \lambda e^{-\lambda t} \int_{t_1=t}^{\infty} f(t_1)dt_1dt$,

where $f(t_1)$ is exponentially distributed with parameter μ .

That is, $f(t_1) = \mu e^{-\mu t_1}$, and $F(x) = 1 - e^{-\mu x}$, $x \geq 0$. From these relationships, we can express the hit ratio as follows [Jain and Lin, 1995]: $p = \int_{t=0}^{\infty} \lambda e^{-\lambda t} \int_{t_1=t}^{\infty} \mu e^{-\mu t_1} dt_1 dt = \lambda / (\lambda + \mu)$.

Note that for different values of LCMR, there will be different values of hit ratio.

Several works have used the CMR to compare different algorithms and to show that it is one of the factors that could heavily affect the performance of these systems. For instance, the number of calls and moves generated by

a mobile host in a unit time can be modeled as a Poisson distributed random variables. Then, the time interval between successive moves or calls can be obtained from the product of CMR and the Poisson distributed variables [Cho, 1998]. In his simulation study, each mobile host component repeatedly fires the call or move event with the time interval computed, and runs a corresponding routine. The simulation results obtained in the study showed that the number of location updates was dependent on two simulation parameters: the CMR and the symmetric rate, which is defined as the ratio between the time a mobile host stays at its home MBS and the time it spends in other MBSs. One study, [Ho and Akyildiz, 1997], argued that in general, the relative cost increases with the CMR. When the CMR is low, the mobility rate is high and the cost for the location registration dominates. When the CMR is high, the mobility rate is low and the cost saving from location registration diminishes. Therefore, the cost reduction is most significant when the CMR is low and the cost for accessing the home base station is high. Interestingly, FBFind algorithm [Kim and Smari, 2003a] showed different results. In this work, LCMR was varied from 1 to 10 and the cost of the algorithm was measured. They found that the algorithm saved about 40% more in costs when the LCMR was 10 than when it was 1, i.e., the algorithm saved more in costs with higher LCMR. In another work on distance-based updating cost analysis [Kim and Smari 2003b], they modified the FBFind algorithm's system model and measured the cost of the algorithm. They found that they could save more than 50% of the costs when the LCMR is 10 than when it is 1.

3. LOCATION UPDATE & LOOKUP SCHEMES

Mobile hosts within a cell communicate with other hosts through a MBS which is installed within the cell, as illustrated in Figure 1. This MBS is connected to other MBS through an underlying wire line network. In order for the network to efficiently route incoming messages to a mobile host, each mobile host is required to report its location to the network. This reporting process is called location update. The purpose of location update is to reduce the cost for tracking down the mobile host. An effective location update policy should reduce the average cost as much as possible compared to the no-update policy. There are a number of ways to determine these location update points. The most commonly used scheme is to group the cells into location areas. A mobile host performs location update whenever it enters a new location area [Akyildiz and Ho, 1995].

Typically, location update is done in the following way. First, each MBS broadcasts the identity of its location area periodically. Second, the mobile hosts always listen to the network broadcast information and store the current location area identity sent. If the received location area identity number is different with its previously stored number, the mobile hosts trigger location update procedure.

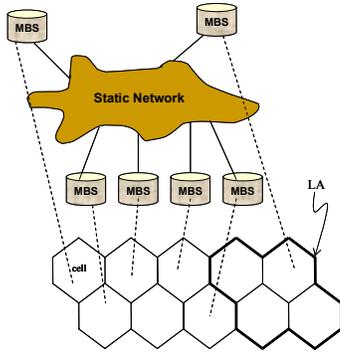


Figure 1 A General Mobile Computing System

In the case of a lookup scheme: when a call is placed to a mobile host or a message is sent to it, the system must be able to locate the host by tracking its movements. To make host tracking easy, the mobile network is partitioned into Location Areas, which are chunks of cells, as was mentioned before [Weng and Huang, 2000]. Now, to perform a lookup scheme, two steps of processing are taken. First, determine the LA where the mobile host is currently located at, and second, page the cells of this particular LA to determine the exact cell within where the MH is residing. There are several schemes introduced to reduce lookup time. Replication [Shivakumar and Widom 1997] and caching [Minh and Van As 2001] schemes are two well known techniques used for this purpose. Replication is different from caching in that it always keeps all copies up-to-date and there is no invalidation problem. But the associated costs of replication will increase rapidly, especially for frequently moving mobile hosts.

In addition to mobile host, mobility agents are very important entities for location and routing. A mobility agent provides the wireless interface between mobile hosts and the rest of the network. It maintains a set of mobility bindings—an association of the host's home identifier with a current locator for the hosts locally or remotely under its control. The agent works a router, and if necessary, forwards packets to a host's current location using the binding information it has. If a host is initially registered with this agent, the agent is called the home agent, otherwise, it is known as a foreign agent. When the host moves to another agent, the current agent becomes known as the previous agent [Cho, 1998].

Source messages intended for a mobile host can be routed in one of two ways: informed routing or triangle routing [Yates et al, 1996]. In informed routing, the source knows the direct route to the mobile host, and is informed of all location changes by the mobile host. In triangle routing, the source directs messages to a home agent that forwards messages to the mobile host.

Routing impacts update procedure performance directly. We can say that the efficiency of a location update

depends on how to distribute the location information through the entire network. Usually, the efficiency of location update is measured by the total cost of routing a packet to its destination mobile host. The total cost to route a packet to its destination mobile host is contributed by two parts, update cost and routing cost. Update cost consists of both registration cost and patron service update cost, while routing cost consists of search cost and hop cost between the node in which the mobile host's location binding is found and the destination mobile host [Hacacute and Huang, 2000].

4. GENERAL CLASSIFICATION OF LOCATION UPDATE SCHEMES

Classifying location management schemes can be considered from several perspectives. One way is to look at these schemes under the two main components of any management technique, namely, updates and lookups. Most of the literature has focused on the former. Due to space constraints, we will concentrate on it too in this work. Update schemes can be of three main types: dynamic [Akyildiz and Ho, 1995; Austin and Stuber, 1996; Bhattacharaya and Sajal, 2002; Chen, 2000; Cho, 1998; Doley et al, 1996; Hacacute and Huang, 2000; Ho and Akyildiz, 1997; Kim and Smari, 2003; Krishnamurthi et al, 1998; Lee et al, 2001; Lin, 1997; Liu and Maguire, 1996; Maass, 1998; Pissinou et al, 1999; Rocha et al, 1999; Scourias and Kunz, 1999; Suh et al, 2000; Wang and Huey, 1999]; static [Krisha et al, 1996]; and adaptive [Bharghavan, 1997; Yates et al, 1996]. Figure 2 shows this type of classification.

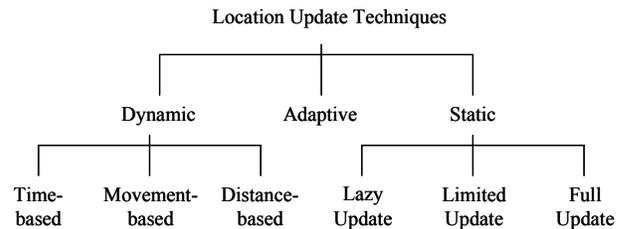


Figure 2 Traditional LUA Classification

A static algorithm may fall under one of three different types of updating methods: lazy update (LU), limited update (LMU), and full update (FU), depending on the update cost metric [Krisha et al, 1996]. Here, the updating occurs by querying from mobile base stations (either home MBS or visitor MBSs). The lazy update (LU) is the simplest update scheme, where an update occurs only at home MBS and the last visited MBS. In this case, the cost of update is zero because no update messages are sent to other visited MBSs. On the other hand, in the full update (FU) case, the update messages are sent to all visited MBSs through the last visited MBS. Hence, the cost of this update depends on the number of visited MBSs. Lastly, in the limited update (LMU) case, the update messages are sent to a specified number of visited MBSs.

Dynamic location update methods [Bhattacharaya and Sajal, 2002] can be classified into three categories too: time-based, movement-based and distance-based. Under strategies using the three categories, location updates are performed based on the time elapsed, the number of movements performed, and the distance traveled since the last location update, respectively.

An adaptive location management algorithm is sought when seamless location update is required across different mobile networks (i.e., between intranets). If a mobile host moves from one mobile network to another, the algorithm provides the necessary mechanisms to maintain service across the networks quickly and without losing connectivity. For example, mobility from indoor to outdoor mobile computing networks may result in a bandwidth decrease by two orders of magnitude. In order to provide a graceful degradation of the operating environment, mechanisms for system and application levels adaptation are necessary [Yates et al, 1996].

Table 1 summarizes algorithms that are referenced in this paper and indicates the respective category for each algorithm. The first column shows the reference cited in the article. The second column lists the name of the location update algorithm (LUA). The third column shows the type of policies used: static (S), dynamic (D), or adaptive (A) and their respective subcategories (e.g., time, movement or distance for dynamic, lazy, limited or full for static). The fourth column denotes the method used to model or solve the problem. (Markov) indicates that authors use a Markovian model to develop their solutions. Similarly, (Prob) is for a probabilistic mathematical approach, (Math) is for other mathematical techniques, such as Gaussian, Laplace transform, ratios, etc., (Pseudo) is for a pseudocode based algorithm. Last, (Simul) indicates that simulation was employed for the solution. The last column indicates whether the corresponding algorithm uses hierarchical (H) or nonhierarchical method (N), which will be discussed later, and the targeted application type (V for video and D for data) for the mobile system.

It is worth noting also that several studies have been carried out to compare many of the policies mentioned above [Hacacut and Liu 1998; Krishna et al, 1996; Siddiqi and Kunz 1999].

Table 1 The LUAs Categorization

Author year	Name of (LUA)	Policy	Method	Type /App
Akyil95	Dynamic mobile terminal LUA	D-t	Markov	N/V
Austin96	Direction biased handoff algorithm	D-m	Math.	N/V
Bharg97	PRAYER	A	Simul.	N/D
Bhatta02	LeZi LUA	D-m	Markov	H/V
Chen00	TLA, FRA	D-m	Markov	N/V
Chen98	FRA	A	Markov	H/V

Cho98	Route optimized LUA	D-t	Math.	N/D
Doley96	Modified tree method	D-t	Pseudo	H/D
Hacac00	LU routing schemes	D-m	Review	H/D
Ho97	Dynamic Hierarchical LUA	D-m	Prob.	H/V
Janni97	HiPER LUA	S-l	Simul.	H/D
Kim03 a	FBFind algorithm	D-d	Prob.	H/D
Kim03 b	DBLM algorithm	D-d	Prob.	H/D
Krishn98	Optimal LUA	D-d	Prob.	N/V
Lee01	LUA for frequently visited locations	D-m	Prob.	H/V
Lin97	Two Location Algorithm	D-m	Math.	N/V
Liu96	Prediction Algorithms	D-m	Markov	H/D
Maass98	Location aware mobile algorithms	D-m	Pseudo	H/D
Madh95	Dynamic programming method	D-d	Markov	N/V
Pissin99	Location and query management algo.	D-m	Pseudo	N/D
Rocha99	Mobile unit tracking algorithm	D-m	Pseudo	N/V
Scouri99	Activity based LUA	D-m	Simul.	N/V
Suh00	Hierarchical LUA	D-m	Prob.	H/D
Wang99	Distributed LUA	D-m	Markov	N/D
Yates96	Mobile assisted adaptive LUA	A	Prob.	N/D

5. A NEW CLASSIFICATION METHOD

We propose a new taxonomy for location management schemes based on the modeling technique employed. At this taxonomy's top level, we consider *hierarchical* [Bhattacharaya and Sajal, 2002; Chen, 98; Doley et al, 1996; Hacacut and Huang, 2000; Ho and Akyildiz, 1997; Jannink et al, 1997; Kim and Smari, 2003a and b; Krishna et al, 1996; Lee et al, 2001; Liu and Maguire, 1996; Maass, 1998; Suh et al, 2000] versus *non-hierarchical* modeling techniques [Akyildiz and Ho, 1995; Austin and Stuber, 1996; Bharghavan, 1997; Chen, 2000; Cho, 1998; Ho and Akyildiz, 1997; Krishnamurthi et al, 1998; Lin, 1997; Pissinou et al, 1999; Rocha et al, 1999; Scourias and Kunz, 1999; Siddiqi and Kunz, 1999; Yates et al, 1996]. The motivation for considering the hierarchy of models in classifying location management approaches of mobile systems is due to the fact that these systems, by design, are either hierarchical or nonhierarchical. As such, their models must be of corresponding nature. Hence, using this classification should prove useful in understanding, analyzing and designing these systems. Figure 3 shows the overall proposed classification.

A basic hierarchical model of a mobile system consists of multiple layers. Each layer stores databases that correspond to information about the lower layer to it. At the lowest layer, which is referred to as the MBS layer, the database information stored consists of information about all mobile hosts that visit the cell (or location area) serviced by that MBS. The second lowest layer stores database information about the MBS layer. At the highest level, i.e., the hierarchy's root, the database includes

information about all the children of that root. Typical information may include database ID, user ID, user profiles and pointers. In practice, this conceptual root may be “distributed” over several lower level “roots”, each of which can service its own lower levels. This way one database (at the actual root) need not store all users’ information nor service all root level queries and updates [Jannink et al, 1997]. When using a hierarchical technique, which employs a tree like modeling of the mobile system, there are two main subcategories to consider: threshold-based (using time, movement, and distance criteria) and non threshold-based (partitioning, grouping, and caching). These will be discussed further shortly.

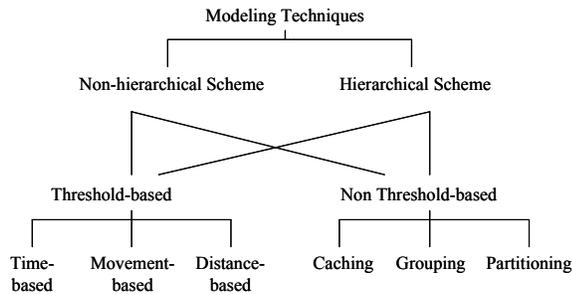


Figure 3 A New Classification of LM Algorithms

In a non-hierarchical technique, which Elnahas and Adly [Elnahas and Adly, 2000] referred to as a two-tier scheme, the current location of a mobile host is stored at two possible locations in the network. If the mobile host is at its home MBS, then its current location is maintained in that MBS. If, on the other hand, the mobile host is visiting another cell (or LA), then, its current location is maintained in its home MBS as well as in the MBS that services the cell it is visiting. The current location is updated at each move. Nonhierarchical schemes may also be of two subcategories: threshold- and non threshold-based. These will also be further discussed shortly.

5.1 Threshold-based Schemes

In a threshold-based scheme, a pre-specified value is used to trigger events of interest. This threshold value may relate to time, distance, or movement associated with the mobile host. For example, an update or lookup may occur if a prescribed time has elapsed since the last update or lookup. Threshold based schemes may be used to design update or lookup policies. In either case, there are three taxonomy subcategories that can be considered: time-, movement-, or distance-based.

In a time-based threshold scheme, the mobile host sends periodic updates (or lookups) to the system (a MBS). The period or time threshold T between updates can be programmed into the mobile hosts using timers. However, the cost due to redundant updates made by stationary mobile hosts has to be tolerated. Obviously, these stationary mobile hosts do not need to send updates which

represent no new data. Akyildiz and Ho [Akyildiz and Ho, 1995] address this problem by proposing the use of time points to check the mobile host location: if no movement is detected, then the MH need not send any update; otherwise, it will be allowed to. The method records the time spent by an MH in each location and uses these times to establish the MH movement time patterns. Then, it uses these patterns to determine the time for the next update (i.e., the MH did move).

In a movement-based threshold approach, the mobile host sends update messages to the system (a MBS) when it crosses a pre-specified number of cells. That means, the mobile host needs to count the number of cell boundary crossings and update when the count reaches certain threshold M . For efficiency purposes, cell sizes may differ between, for example, urban and rural areas. It is expected then that the number of crossings will change in these two cases. Applying a threshold policy means more updates will be made in the former case. Hence, one of the drawbacks of this scheme is increased signaling traffic due to different sizes of the cells. Scourias and Kunz [Scourias and Kunz, 1999] use the mobile host’s mobility patterns to reduce this signaling traffic. They develop a model that stores information about the mobile hosts’ daily movement patterns to minimize the traffic costs.

In a distance-based threshold solution, the mobile host is required to track the Euclidean distance from the location of the previous update and initiates a new update if the distance exceeds a specified threshold D . The distance could be specified in terms of the distance units used or the number of cells between the two positions. Madhow et al [Madhow et al, 1995] discuss finding the optimal value of D by using the expectation functions of the sum of update costs until the next update. They compare an iterative algorithm and a difference equation to find the optimal D and show that the proposed iterative algorithm works better.

5.2 Non-Threshold-based Schemes

In a non-threshold-based technique, no pre-specified value is used to trigger events of interest or to reduce the costs. Instead, these approaches use grouping methods, caching methods and so on. A well-known method under this category is location area partitioning. In this method, the service area under the static network is partitioned into location areas (LA) formed out of neighboring cells. The LAs could be overlapping. A mobile host must update whenever it crosses an LA boundary. Its location uncertainty is reduced by expanding the search space to the set of cells under the current LA rather than having per cell searches. All cells under an LA are paged simultaneously upon a call arrival, resulting in an assured success within a single step (i.e., an MH in that LA will connect). The MBSs must broadcast the LA-id (along with the cell-id) to help the MHs perform the update.

Weng and Huang [Weng and Huang, 2000] introduce a modified grouping method to achieve maximum cost savings in the network. They consider location areas with different sizes and calculate costs associated with each. The claim is that this method could yield a significant improvement over the conventional cell structure. However, there is a drawback: if a system has LAs of larger size, the technique increases the routing costs.

Bharghavan [Bharghavan,1997] discusses a data caching method. This method reduces access time but causes higher wireless traffic by reducing cache size. It also introduces a related problem: data consistency, since multiple copies of shared data are maintained. Thus, most approaches to caching ‘hoard’ data aggressively, and allow the mobile user to manipulate the cached copy at the portable when disconnected. The modified data is reintegrated with the server copy upon reconnection and update conflicts are typically reconciled by human intervention in the worst case.

5.3 The Taxonomy

Figure 4 shows the new classification of location management techniques. We can organize the known techniques of location management into this 144-cell taxonomy. For instances, the A algorithm represents a dynamic update technique under hierarchical model that deals with voice transmission application, using distance based threshold approach. Likewise, the B algorithm indicates a static lookup technique with non hierarchical partitioning model and voice transmission application, with non threshold approach. Hence, using this taxonomy, we can easily identify any LM problem at a glance.

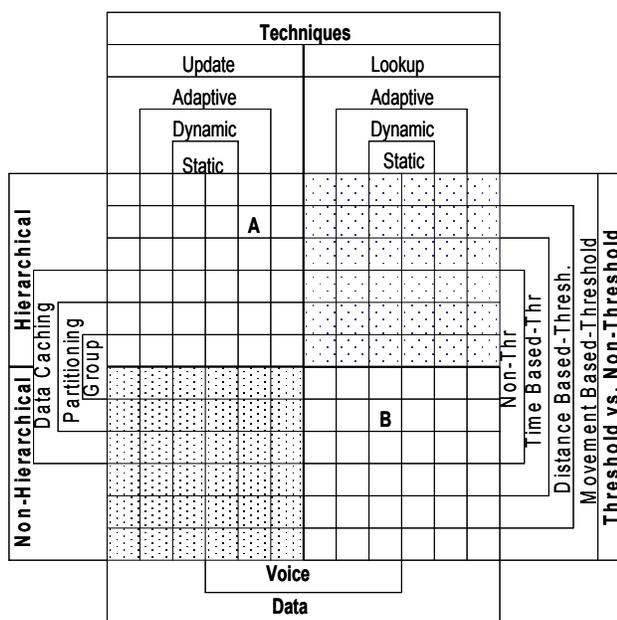


Figure 4 The New Classification of Location Management Techniques

6. CONCLUSIONS AND FUTURE WORK

In the past few years, location management issues and solutions have received a good deal of attention, both in literature and industry. Efficient location management techniques are an important aspect to consider in the design of future mobile and wireless environments, especially since the number of mobile hosts is poised to increase at a high rate. In this study, we defined the location management problem and its terminology, reviewed some of the main accomplishments achieved by researchers in the field, and established the fundamental issues that impact this problem. We also attempted to compare the solutions qualitatively according to their effectiveness for different types of updating techniques. That analysis led us to propose a new taxonomy for location management techniques in mobile and wireless environments based on several important factors. For future work, we plan to extend our analysis to other possible solutions of the location management problem and help devise more efficient and robust ones.

In closing, it is worth speculating on the long-term impact of location management issues on mobile and wireless environments and their design. The future for wireless and mobile computing is promising indeed, especially since technological advances continue to support more sophisticated applications for these environments.

REFERENCES

Akyildiz I. F. and Ho J. S. 1995, “Dynamic Mobile User Location Update for Wireless PCS Networks”, *Wireless Networks*, Vol. 1, Pp187 – 196.

Austin M. and Stuber G. 1996, “Direction Biased Handoff Algorithms for Urban Microcells”, *Wireless Personal Communications*, Vol. 3, Pp287 – 298.

Bharghavan V. 1997, “Challenges and Solutions to Adaptive Computing and Seamless Mobility over Heterogeneous Wireless Networks”, *Wireless Personal Communications*, Vol. 4, Pp217 – 236.

Bhattacharaya A. and Sajal K. D. 2002, “LeZi-Update: An information Theoretic Framework for Personal Mobility Tracking in PCS Networks”, *Wireless Networks*, Vol. 8, Pp121 – 135.

Chen I. R. 2000, “Analysis and Comparison of Location Strategies for Reducing Registration Cost in PCS Networks”, *Wireless Personal Communications*, Vol. 12, Pp117 – 136.

Chen I. R., Chen T. M. and Lee C. 1998, “Performance Evaluation of Forwarding Strategies for Location Management in Mobile Networks”, *The Computer Journal*, Vol. 41. No. 4, Pp243 – 253.

Cho G. 1998, “A Location Management Scheme Supporting Route Optimization for Mobile Hosts”, *Journal of Network and Systems Management*, Vol. 6, No. 1, Pp31 – 50.

Dolev S., Pradhan D. and Welch J. L. 1996, “Modified Tree Structure for Location Management in Mobile Environments”, *Computer Communications*, Vol. 19, Pp335 – 345.

- Elnahas A. and Adly N. 2000, "Location Management Techniques for Mobile Systems", Information Sciences, Vol. 130, Pp1 – 22.
- Feller W. 1968, An Introduction to Probability Theory and Its Applications, Wiley, New York
- Hacacute A. and Huang Y. 2000, "Location Update and Routing Scheme for a Mobile Computing Environment", International Journal of Network Management, Vol. 10, Pp191 – 214.
- Hacacute A. and Liu B., 1998, "Database and Location Management Schemes for Mobile Communications", IEEE ACM Transactions on Networking, Vol. 6, No. 6, Pp 851 – 865.
- Ho J. S. and Akyildiz I. F. 1997, "Dynamic Hierarchical Database Architecture for Location Management in PCS Networks", IEEE/ACM Trans. Networking, Vol. 5, No. 5, Pp646 – 660.
- Jain R. and Lin Y. 1995, "An Auxiliary User Location Strategy Employing Forwarding Pointers to Reduce Network Impacts of PCS", Proc. of the International Conference on Communications, Pp1 – 26.
- Jannink J., Lam D., Widom J. and Cox D. 1997, "Efficient and Flexible Location Management Techniques for Wireless Communication Systems", Wireless Networks, Vol. 3, Pp38 – 49.
- Kim S. Y. and Smari W. 2003, "A Frequency-Based Find Algorithm in Mobile Wireless Computing Systems", ISCA 18th International Conference on Computers and Their Applications, Pp25 – 31.
- Kim S. Y. and Smari W. 2003, "Distance-based Location Updating Cost Analysis in Mobile Wireless Environments", IEEE Semiannual Vehicular Technology Conference Fall 2003, Forthcoming.
- Krishna P., Vaidya N. and Pradhan D. 1996, "Static and Adaptive location Management in mobile Wireless Networks", Computer Communications, Vol. 19, Pp321–334.
- Krishnamurthi G., Azizoglu M. and Somani A. K. 1998, "Optimal Location Management Algorithms for Mobile Networks", The 4th Annual ACM/IEEE Intl Conf. on Mobile Comp. and Networking, Pp223 – 232.
- Lee C., Ke C. and Chen C. 2001, "Improving Location Management for Mobile Users with Frequently Visited Locations", Performance Evaluation, Vol. 43, Pp15 – 38.
- Lin Y. B. 1997, "Reducing Location Update Cost in a PCS Network", IEEE/ACM Transactions on Networking, Vol. 5, No. 1, Pp25 – 33.
- Liu G. and Maguire G. Jr. 1996, "A Class of Mobile Motion Prediction Algorithms for Wireless Mobile Computing and Communications", Mobile Networks and Applications, Vol. 1, Pp113 – 121.
- Maass H. 1998, "Location-aware Mobile Applications Based on Directory Services", Mobile Networks and Applications, Vol. 3, Pp157 – 173.
- Madhow U., Honig M. L. and Steiglitz K. 1995, "Optimization of Wireless Resources for Personal Communications Mobility Tracking", IEEE/ACM Transactions on Networking, Vol. 3, No. 6, Pp 698 – 707.
- Minh, H. N, and H. R. Van As, 2001, "User Profile Replication with Caching for Distributed Location Management in Mobile Communication Networks", Proceedings of the 2001 ACM Symposium on applied Computing, Pp. 381 – 386.
- Pissinou N., Makki K. and Campbell W. J. 1999, "On the Design of a Location and Query Management Strategy for Mobile and Wireless Environments", Computer Communications, Vol. 22, Pp651 – 666.
- Rocha M., Mateus G. and Silva S. 1999, "A Comparison Between Location Updates and Location Area Paging for Mobile Unit Tracking Simulation in Wireless Communication Systems", Proc. of the 3rd Intl. Workshop on Discrete Algorithms and Methods for Mobile computing and communications, Pp72 – 77.
- Scourias J. and Kunz T. 1999, "An activity-based Mobility Model and Location Management Simulation Framework", Proceedings of the 2nd ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems, Pp61 – 68.
- Shivakumar, n., and J. Widom, 1995, "User Profile Replication for Faster Location Lookup in Mobile Environments", International Conference on Mobile Computing and Networking, Pp. 161 -169.
- Siddiqi A. and Kunz T. 1999, "The Peril of Evaluation Location Management Proposals Through Simulations", Proc. of the 3rd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, Pp78 – 85.
- Suh B., Choi J. and Kim J. 2000, "Design and Performance Analysis of Hierarchical Location Management Strategies for Wireless Mobile Communication Systems", Computer Communications, Vol. 23, Pp550 – 560
- Wang K. and Huey J. 1999, "A Cost Effective Distributed Location Management Strategy for Wireless Networks", Wireless Networks, Vol. 5, Pp287 – 297.
- Weng C. M and Huang P.W. 2000, "Modified Group Method for Mobility Management", Computer Communications, Vol. 23, Pp115 – 122.
- Yates R., Rose C., Rajagopalan S. and Badrinath B. R. 1996, "Analysis of Mobile-assisted Adaptive Location Management Strategy", Mobile Networks and Applications, Vol. 1, Pp105 – 112.

BIOGRAPHY



Seung-yun Kim is a Ph.D. student in the Electrical and Computer Engineering Department at the University of Dayton, Dayton, Ohio, USA. His research interests are: Mobile and Wireless Computing, Distributed Processing, and Cluster Computing. Kim received a B.S. in Electrical Engineering from Parks College of Engineering and Aviation

of St. Louis University in St. Louis, Missouri in 1999 and a M.S. in Electrical Engineering from the University of Dayton. He is a member of the IEEE.

REAL TIME SYSTEMS FOR URBAN MODELLING

C. SWIFT^{1,2}, K. LEINEMANN¹, G. SCHAEFER²

¹*Institut für Angewandte Informatik, Forschungszentrum Karlsruhe, Germany*

²*School of Computing and Mathematics, The Nottingham Trent University, U.K.*

Abstract. This paper discusses how real time systems for urban modelling are becoming a realistic possibility. Portable computers bring increased flexibility for the urban modeller but at the same time challenges for the system designer. Portable computing also allows for real time on-site measurement and modelling. The work on real time modelling at the Forschungszentrum Karlsruhe is introduced and a flexible method for describing the geometry of roofs from angles and offsets is proposed. Furthermore, an overview of a prototype system developed is given.

Keywords: urban modelling, architecture, real time, portable computing, object orientated model combined systems, on site, total station

1. INTRODUCTION

The surveying of urban areas to create 2D maps for a variety of uses is now an established practice. However, using this survey data to create realistic 3D models is a technology that is still developing. This process is generally described as “urban modelling” and covers a variety of methods that produce models ranging from single buildings to fully modelled cities. The range of uses varies greatly; models may be required for statistical analysis, as production prototypes for models, to aid in planning decisions or for visual simulations such as virtual tourism.

Despite the large amount of current research in the area of urban modelling, the general opinion of the commercial value of urban modelling varies greatly. The current main attraction of the technology seems to be the visual effect, rather than as a way as presenting data for analysis and decision-making [1].

Despite this lack of a clear role for urban modelling the champions of the technology have identified many potential application areas. A selection of the most common is below:-

- Geometrical models for pollution mapping, wave propagation modelling.
- Urban planning models
- Base models for wooden or plastic architectural models by CNC machines.
- Modelling of cultural heritage monuments.
- Virtual models for tourism.

One of the problems with successfully developing urban models is that the diversity of the above areas

creates a wide range of requirements in accuracy, data, speed, and model realism [2]. This results in the dilemma for designers of creating expensive specific systems tailored to individual tasks or using the best compromise of current techniques.

The result of this is that few automated or semi-automated techniques exist. Automated and semi-automated techniques improve the speed of data capture, thus making the process cost effective. If cost effectiveness is achieved then the range of application areas is potentially huge. The large number of academic and research institutions that have projects in urban modelling reflects this potential.

2. PORTABLE SYSTEMS

Portable computers offer urban modellers the tantalising possibility of both measuring and modelling on site. Work done in this area suffers however from the comparatively decreased power of notebook and handheld computers over desktop machines. With the gain in increased portability come various interface problems for software designers. Generally portable computers differ from desktop machines in the following:-

- Portable computers have physically smaller displays that currently have a lower resolution than desktop displays.
- Due to size and power consumption the processing power, memory as well as graphics speed and storage lags behind desktop computer capabilities let alone those of a workstation.
- Input devices differ greatly from touch pads to touch sensitive screens.

- Connectivity with current trends, such as a wi-fi or bluetooth connections are as likely if not more so than a traditional RS232 port.

The most significant from the above differences for the user interface designer are the screen limitations and reduced graphics power. 3D modelling packages normally adopt a tri-view approach to modelling. The user interface has four 3D viewers showing an overhead view, two differing side views and a perspective view. For desktop machine with a 21-inch monitor capable of a native resolution of 1600 x 1200 or more, this presents no problem. However for a small device with a screen maybe a quarter the size and resolution this is impractical. Not only because of the lack of clarity created but also because when working in field conditions the views are too small to be manipulated accurately. A solution to this is to use a single 3D viewer that can switch between the four different viewpoints required. As well as being clearer this also requires less graphics power to run.

The small display also creates two further problems. Displaying the hierarchy and attributes of the project being worked on and giving the user access to the functions that are required to work on the project. The normal 3D CAD system solution is to divide the objects into layers to separate large 3D data sets into manageable chunks and then use toolbars and floating tool palettes to give users access to the functions of the program. The problem with this system is that urban models are large typically extending over kilometres and as such consist of a large number of objects that must be subdivided [1]. Using layers this creates either a very large list of layers that is difficult to navigate or a few layers with lots of objects that are hard to find and edit. Furthermore, one problem with toolbars and floating tool palettes is that whilst they are a good way of showing often used functions they also take up valuable screen space, which is important on a portable device.

Geometrically the relationship between differing sub-divisions in a city/urban area follows a tree shape. This is because they are artificial constructs designed and planned artificially [3]. They can be therefore subdivided into objects such as suburbs, streets, industrial areas, and buildings that can wholly contain other objects. Because of this relationship the whole project can be represented using tree lists. The advantages over layers include:-

- The user can decide which data to show by expanding and collapsing different branches of the tree.
- Single objects can be selectable and also part of higher-level groups.

- By engineering a basic tree object type that other objects are derived from, any object could be part of any branch, with the rules governed by the designer not the system.

Having decided on showing tree lists for the project structure the problem of cluttering the screen with toolbars and palettes is even more pressing. Putting all the functionality of the program in the main menu system presents the user with a bewildering array of options seemingly unconnected to the object or task selected. Another approach is therefore required.

Popup menus can be dependant upon cursor position and selection. This provides the possibility of having the selected object or task dictating which options are available. This can, not only save space but also help lead the user through the program. Using these popup menus to launch dialog boxes for user input and control removes the need for large floating tool palettes similar to those used in image processing packages

3. REAL-TIME SYSTEMS

Traditionally measurement and modelling were separate functions that were conducted by separate programs. The data was measured by the instrument, loaded onto the computer by an interface program and then loaded into a separate modelling package. This method created the limitation that it was not possible to have a real time system. However with the increased take up of portable computers for urban modelling real time systems are becoming a realistic possibility.

A real time system offers several possible advantages over the traditional method: -

- The model can be viewed and created on site and on the journey to and from the site increasing productivity and making recognition of errors easier. Once recognised, errors can be corrected on site. This reduces travel time to make corrections.
- Information can be entered directly onto the finished model during measurement making the need to sketch the survey with pen and paper redundant. (This electronic sketchbook method is introduced in the next section.)
- Traverse calculations, area calculations, and free stations can be included in the software providing quicker traversal, increased flexibility and if correctly implemented reduced complexity for the user.
- Coupling of measurement techniques to modelling allows semi automatic creation of models, including draft creation (sketching) and then measurement of all points within

objects or the use of attributes (angles distances etc) to define objects

Reflectorless and servo motorised total stations are commonly in use today. However the ability to control these instruments by computer is not commonly used. Computer control of these total stations can remove the need for post survey traversal calculations, improve the accuracy of measurement, and speed up the survey process. Because of this, total stations are normally the central instruments in any combined system.

4. OBJECT-ORIENTED MODELLING

As stated previously one of the primary aims of implementing the system in real time is to provide a method of reducing both errors on site and the amount of hand written data created.

To facilitate this a system of object orientated sketching is proposed. The objects that form the building (doors, roofs, ground plans, etc.) are sketched in the 3D viewer by the user before measurement. Rather than calculating the geometry separately after all sketching has taken place, the sketching, measurement systems and graphics architecture are integrated together to provide real time sketching, measurement and display.

This should be made possible by removing complex surface calculations required and by recalculating only entities affected by each measurement.

The objects to be sketched are implemented as objects within the system. This allows intelligent measuring and definition strategies to be used saving time and simplifying the measurement process.

5. MODELLING ROOFS

The most documented roof creation algorithm is the straight skeleton algorithm [4]. However this does not allow the creation of standard roof types from measurement offsets or angles.

Despite a lot of research no single algorithm can create all roof types. One of the problems is that roofs are simply so diverse that from one ground plan multiple roof shapes are possible. This diversity increases with the complexity of the underlying ground plan.

Due to this diversity any system wishing to comprehensively model roofs must provide several creation systems. The system should then suggest to the user the correct creation method depending on the type of roof to be modelled and the input parameters available. After creation or update using one of the algorithms the roof geometry can then be

passed to the standard roof object and additional attributes set and added as necessary.

6 TRESTLE DEFINITION METHOD

The system for roof modelling developed at the Forschungszentrum Karlsruhe is based on the premise that modelling a roof using its trestles, which define the roofline geometry, is a sensible abstraction since the structure of the roof can be defined by its trestles alone.

Figure 1 shows in blue the trestles that the system would use to define the roofline points of the example roof.

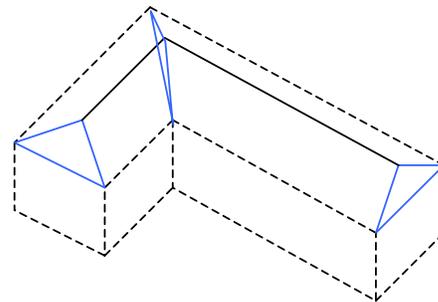


Figure 1 Trestles within a roof structure

6.1 Sketching

Before measurement or definition of the roof the user first sketches the roof object. To sketch a roof the user is required to select the points that form the roof base. The system presents the user with a list of possible roof types based on the number of points selected. This assists the user and ensures the system defines a correct roof type. This is shown in figure 2.

A roof generator object is created that encapsulates the required number of trestle objects to define the roof shape.

Roof types can be dynamically defined at runtime from data files. The definition of the roof types also includes default angles between the trestles and the base of the roof.

This enables the system to render the roof in 3D before a single measurement has been taken or parameter defined. Having sketched a roof it can then be precisely defined by measurement or definition from given parameters whenever this is required.

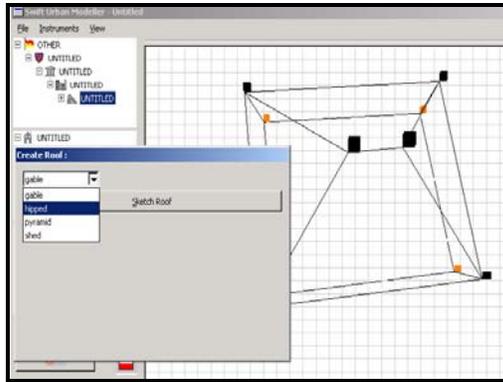


Figure 2 Roof sketching based on selection of roof base line points.

The following vernacular roof types are currently supported:-

- Gable
- Saltbox
- Hipped
- Pyramid
- Shed
- Cross-hipped
- Cross gable
- M-shaped

6.2 Measurement

The sketched roof can be defined by measuring the roofline points using the reflectorless mode on the total station. This requires the existence of a line of sight between the instrument and the points to be measured.

The quickest method is to measure a single point as the roofline point. This however is likely to be only accurate to within a few centimetres as the laser targeting spot diverges over distance. Work has been done on using several measurements to get a more exact definition of a target point [5]. However, this sacrifices speed for enhanced accuracy. The user should therefore pick the measurement type most appropriate to the application area of the urban model being generated.

6.3 Parametric Definition

There are several cases when taking a measurement of the roofline points is not desirable or possible. For example, it may not be possible to establish a direct line of sight between the instrument and the roofline points. The user may also be in possession of architectural plans detailing dimensions and/or angles of the structure. Using these to define the roof geometry may be clearly preferred. Another possibility would be that the user is defining a large urban area possibly from pre-measured ground plans mainly for visual purposes

and therefore millimetre accuracy would not be required.

To cope with these situations the system gives the user the option of defining each of the roofline points based on offset distances from the walls or using the angles made between the trestle and the roof base. Due to the simplicity of the calculations once a parameter is redefined the roof geometry can be updated in real time. Details on roof geometry algorithms and calculations are given in the Appendix.

The system resists using constraints based on the geometry of the roof type to reduce the number of measurements as this would also reduce the flexibility of the system and hence lead to undesired results for irregularly shaped ground plans. Flexibility is retained within the system by allowing the user to define each roofline point from measurement, angles or offsets. Having defined a point using one method it can then be redefined using another. This allows the methods to be mixed within a single roof construct.

7. REAL TIME SKETCHING SYSTEM

This section gives an overview of a prototype system that was implemented in (Visual) C++ using the Fox Toolkit and OpenGL to provide a GUI interface suitable for mobile computers. Its main aim is to prove some of the concepts outlined in this paper. In its current version it has the following capabilities: -

- Implementation of a real time sketching system allowing the user to sketch building floor plans, extrude the floor plan and then sketch a roof before measurement. Giving full 3D visualisation of the scene.
- Provision of a custom TCR Interface API to allow real-time measurement from a TCR30X series Leica total station.
- GUI allows visualisation of project hierarchy on a small (800 x 600) notebook screen without sacrificing the ability to view the scene as adjustments to the project are made.
- A roof generator allows creation of roofs as discussed in Section 5. These can then be adjusted by angles or by measuring the roofline points using the laser on the TCR.
- A 3D viewer supports plan, side and projection views with zoom in all three and translation in the plan and side views.

A screen shot of the prototype platform can be seen in Figure 3.

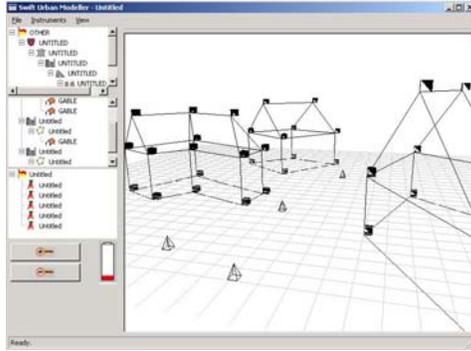


Figure 3 Urban modelling prototype system.

8. CONCLUSIONS

The development of real time portable urban modelling systems is now technically possible. However despite great potential, computer based urban modelling systems are still too much in theory and discussion and as such of no real interest to the software developers [1]. Standardisation and a target market must first be established before the concept meets its potential.

REFERENCES

- [1] BOURDAKIS V, 1998, On Developing Standards for the Creation of VR City Models, Laboratory of Environmental Communication and Audiovisual Documentation, University of Thessaly, Greece.
- [2] MUELLER H, Three-Dimensional Virtual Reconstruction of Buildings: Techniques and Applications. Institute for Spatial Information and Surveying Technology i3mainz, University of Applied Sciences at Mainz, Germany.
- [3] ALEXANDER C, April 1965, A City is not a Tree parts I, Architectural Forum, Volume 122, No 1, pp 58-62 (Part I),
- [4] AICHHOLZER O and AURENHAMMER. F, 1996, Straight skeletons for general polygonal figures in the plane. *Proc. 2nd Annual International Conference Computing and Combinatorics*, pp. 117-126. Lecture Notes in Computer Science 1090, Springer.
- [5] SCHERER, 2002, Advantages of the Integration of Image Processing and Direct Coordinate Measurement for Architectural Surveying – Development of the System Total -, XXII International Congress Washington DC. USA April 19-26-2002

APPENDIX

Basic trestle algorithm

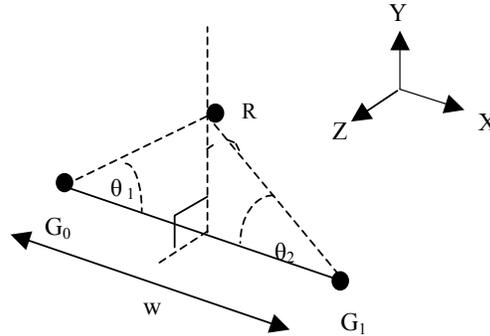


Figure 4 Roof with roofline point over G_0G_1

Figure 4 shows a roofline point R that lies directly above G_0G_1 . The point can be defined from the angles θ_1 and θ_2 .

First the distance G_0G_1 is calculated by Pythagoras's theorem.

$$w = \sqrt{(G_{1x} - G_{0x})^2 + (G_{1z} - G_{0z})^2}$$

Next we build two equations to describe the lines G_0R and G_1R . These are straight lines and therefore in the format $y = mx + c$. Firstly the values of m are calculated as

$$m_1 = \frac{\cos \theta_1}{\sin \theta_1} \quad m_2 = \frac{\cos \theta_2}{\sin \theta_2}$$

The intercept values of these simultaneous equations are C_1 and C_2 . C_1 is then equal to 0 and C_2 equal to w . The two equations are solved simultaneously to find the height h of the intersection and the distance along the line G_0G_1 is the solved value of c .

From this the x , y , and z coordinates of the roofline point can be calculated relative to G_0 .

$$R_x = G_{0x} + \left(\left(\frac{c}{w} \right) \times (G_{1x} - G_{0x}) \right)$$

$$R_y = G_{0y} + h$$

$$R_z = G_{0z} + \left(\left(\frac{c}{w} \right) \times (G_{1z} - G_{0z}) \right)$$

Offset trestle algorithm

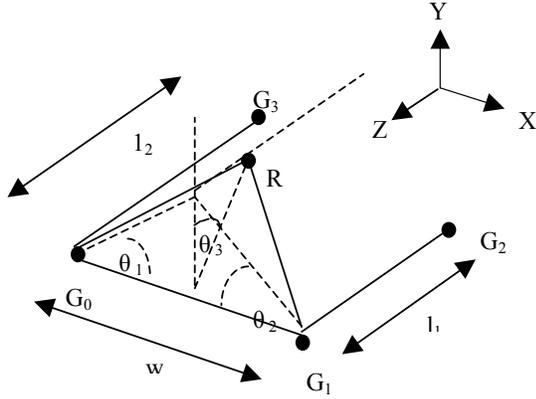


Figure 5 Roof with roofline point translated along $(G_1G_2 + G_0G_3)$

The previous solution is not going to be applicable to many roofline points. This is because the slope of many roofs is along the roofline as well as across it (shown as θ_3 in Figure 5). This occurs mostly at sections where the roof ends. Therefore it is necessary for the definition of this slope within our algorithm.

Steps 1 and 2 from the previous example are utilised and then an additional value l is calculated.

$$l = \sqrt{(G_{2x} - G_{1x})^2 + (G_{2z} - G_{1z})^2} + \sqrt{(G_{3x} - G_{0x})^2 + (G_{3z} - G_{0z})^2}$$

This is the length of the vectors G_1G_2 and G_0G_3 summed. This is because the roofline is defined as $G_1G_2 + G_0G_3$. Defining the roofline from the sum of two vectors takes into account the shape of the roof.

Then, the roofline point is calculated as

$$R_x = G_{0x} + ((c/w) \times (G_{1x} - G_{0x})) + \left(\left(y / \tan\left(\frac{\theta_3}{l}\right) \right) \times ((G_{2x} - G_{1x}) + (G_{3x} - G_{0x})) \right)$$

$$R_y = G_{0y} + h$$

$$R_z = G_{0z} + ((c/w) \times (G_{1z} - G_{0z})) + \left(\left(y / \tan\left(\frac{\theta_3}{l}\right) \right) \times ((G_{2z} - G_{1z}) + (G_{3z} - G_{0z})) \right)$$

Definition from Offsets

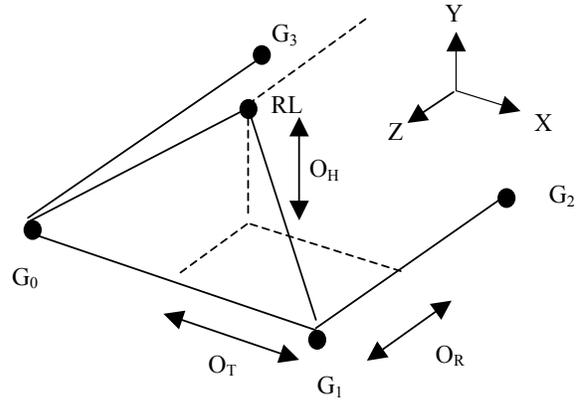


Figure 6 Roof with roofline point determined by offsets

Below is the definition of the roofline point R calculated from the 3 required offsets (see Figure 6). Again as with the offset trestle definition the roofline point is translated along $(G_1G_2 + G_0G_3)$.

$$R_x = G_{0x} + ((o_T / w) \times (G_{1x} - G_{0x})) + ((o_R / l) \times ((G_{2x} - G_{1x}) + (G_{3x} - G_{0x})))$$

$$R_y = G_{0y} + o_h$$

$$R_z = G_{0z} + ((o_T / w) \times (G_{1z} - G_{0z})) + ((o_R / l) \times ((G_{2z} - G_{1z}) + (G_{3z} - G_{0z})))$$

where o_h , o_T , and o_R are the height, trestle and roof point offsets respectively.

CLIENT SIDE SIMULATION TOOL JSSim

JAROSLAV SKLENAR

*Department of Statistics and Operations Research
University of Malta
Msida MSD 06, Malta
Web: <http://staff.um.edu.mt/jsk11/>
E-mail: jaroslav.sklenar@um.edu.mt*

Abstract: JavaScript is an interpreted language where the important techniques of Object Oriented Programming can be utilized. Some of them are not included directly, so they need additional support. For example inheritance has to be programmed explicitly. A JavaScript programmer is thus making use of a modern language that, together with HTML, supports creation of documents that can contain user-friendly input of validated data, any kind of data processing, and lucid presentation of results. Solutions based on JavaScript and HTML are typically placed on the web and made thus available literally to everybody who has a browser supporting particular versions of these two languages. These capabilities have been applied to create various web-hosted problem-solving tools. Such tools can contain simple and medium-scale simulation models. Several simulation models have already been implemented and placed on the web with very encouraging response. Routines used to create these models, including a simple event-oriented simulation engine together with a collection of classes for general use in discrete simulation, have been collected into a tool that we call JSSim. The paper describes the capabilities of this tool by using examples oriented to simulation of queueing systems. The tool also supports direct links between JavaScript objects and parts of the corresponding HTML documents in order to simplify programming as much as possible. A queueing network has been simulated to compare JSSim with ExtendTM and ArenaTM from several points of view.

keywords: Web based simulation, JavaScript, Discrete event simulation, Queueing models.

1 INTRODUCTION

The book [Flanagan, 1998] describes the JavaScript prototype oriented paradigm. The papers [Sklenar, 2001, 2002] explain how to use this paradigm in order to be able to utilize all important techniques of Object Oriented Programming (OOP) in JavaScript. Some new techniques not available in strongly typed compiled Object Oriented Languages (OOL) are also introduced. In particular the programmed inheritance described in the paper [Sklenar, 2001a] enables creation of “subclasses” that inherit only selected methods of the superclass. Thus we can create simplified versions of general superclasses. All these techniques can be used to create a reusable code open to future expansion and modification. In other words in an interpreted JavaScript environment we can use the techniques typical for classical compiled strongly typed OOLs like for example Simula or Java together with the flexibility and simplicity typical for interpreted languages with loose typing. All this of course can be done at the expense of security, but as JavaScript is not intended as a language for large software projects, it is not considered as a big problem. The paper [Sklenar, 2001b] deals with the implementation of a simulation engine that was written entirely in JavaScript and that together with appropriate HTML documents supports user-friendly development of web hosted tools that contain simple and medium

scale simulation models. The engine is based on the classical event-oriented approach with two primitives: *schedule an event at a certain time* and *cancel a scheduled event*. These primitives are implemented as calls to routines with appropriate parameters. Other simulation supporting facilities are also available, for example generation of random numbers, working with queues, and transparent collection and computation of statistics. All these facilities have now been collected into a tool called JSSim (JavaScript Simulation). The purpose of this paper is to describe the capabilities of this tool. Though JSSim is a general tool for event-oriented discrete simulation, examples oriented to simulation of queueing systems will be used.

2 SIMULATION FACILITIES OF JSSim

Facilities found in languages and tools for programming discrete simulation models can be classified into the following main groups:

- *Time control, synchronization and communication of processes*
- *Generation of random numbers*
- *Transparent collection of statistical data*
- *Statistical analysis*
- *Advanced data structures*
- *User-friendly Input and Output*

Next chapters will summarize the implementation of these facilities in JSSim.

2.1 Time Control

For time control we consider only the two commonly used approaches. While the process-oriented discrete simulation represents the most advanced way of modeling the dynamics of complex systems, the classical event-oriented approach is simpler and easier to learn and to implement. That's why it has been chosen for a JavaScript based tool that is not intended for large simulation studies. Assuming that the reader is familiar with the event-oriented principle, these are the basic facts: During (re)loading of the document the engine creates two global variables: the *time* and the empty *sequencing set* (SQS). Events are represented by *event notices* created by the user and stored in the SQS. Each event notice has the occurrence time of the event and any other user-defined data. The engine assigns the time when the event is scheduled. From the user's point of view, the SQS is a list of event notices ordered by the time of occurrence in increasing way. After activation, the engine repeatedly removes the first event notice from SQS, updates the model time, and activates a user routine that is given the reference to the event notice. Simulation ends by the empty SQS or by any user supplied condition. These are the engine routines that are called from the user's part of the simulation model:

`initialize_run()` is a routine that clears the SQS (the previous experiment may have finished with nonempty SQS) and sets the model time to zero. It should be called at the beginning of the model initialization.

`evnotice()` is the event notice constructor. It returns an object with the time property, that is used later by the engine and should not be accessed by the user. The user can add any other properties to distinguish between types of events and to store other model dependent data.

`schedule(event,t)` schedules the event whose notice is the first parameter at the time given by the second parameter.

`modeltime()` is the current time of the model. So scheduling an event *e* after a delay *d* is performed as follows:
`schedule(e,modeltime()+d).`

`cancel(event)` cancels a scheduled event. The function returns a boolean value that reports whether canceling was successful.

`simulation_run(stats,length)` starts the simulation experiment. This routine should be

called after the model initialization that has to schedule at least the first event. The two parameters just affect the progress reporting in the status bar. The routine ends by reaching the empty SQS or by the user supplied terminating condition - the user's routine `finish_run()`.

The above routines are common to all simulation models. Model specific behavior is implemented by two routines that have to be supplied together with the code (preferably also a routine) that starts the simulation. These are the routines (together with examples) that represent the user's part of the simulation control:

`finish_run()` tests whether simulation should be terminated. It is called by the engine after updating the model time just before activating the next user event. It can just test the time against the experiment duration or it can implement a more complicated terminating condition, like for example serving a given number of customers. The following is the function of a model where the experiment is finished by reaching its duration `runlength`:

```
function finish_run() {
    return (modeltime()>runlength)
};
```

`eventroutine(event)` is activated by the engine. The routine is given the reference to the event notice that has been removed from the SQS. The rest is the user's responsibility. Typically there will be some properties created by the user used to switch between various types of events. It might be a good idea to keep this routine short and simple and to write routines for various types of events similarly as they are written in event oriented simulation languages. The following is the function of a model with two types of events:

```
function eventroutine(event) {
    // The event routine switches
    // between types of events
    switch (event.eventtype) {
        case 1: next_arrival(); break;
        case 2:
            end_of_service(event.servnum);
            break;
        default:alert(
            "Wrong eventtype: "
            + event.eventtype);};
};
```

The start of simulation has also to be programmed. For example it can be a function activated by pressing a button "Run". This function is supposed to perform the following activities in this order:

- Initialization of the engine by `initialize_run()`
- Model specific initialization
- Starting simulation by `simulation_run()`
- Model specific experiment evaluation.

The following is an example of a function activated by pressing the button “Run” and its link to HTML. Some model specific tests have been removed.

```
<INPUT TYPE="button" VALUE="Run"
onClick="simulation()">

function simulation() {
  // Tests whether simulation can
  // start (not shown here)
  initialize_run();
  // This prepares the engine
  initialization();
  // Initiates model & statistics
  var ev = new evnotice();
  // Scheduling the first arrival
  ev.eventtype = 1;
  // User defined property
  var x = arrival.generate();
  // Generation of first interval
  intstat.update(x);
  // Interval statistics update
  schedule(ev,modeltime() + x);
  // Scheduling the first event
  simulation_run(showstatus,
    runlength);
  // This starts the experiment
  evaluation();
  // Experiment evaluation };
```

2.2 Generation of random numbers

JSSim contains a rather complex class used to generate instances (objects) that represent random numbers. These can have either a theoretical distribution (so far only few are available), but primarily they are supposed to contain tables used to generate values with a general (for example experimentally obtained) distribution. Methods are available for entering and editing such tables. Working with empirical tables is user-friendly; table entries can be modified, inserted and deleted. Large tables can be saved and loaded (provided cookies are enabled in the browser). Figure 1 shows a table created by HTML used to enter parameters of a random variable. The controls are self-explaining. Figure 1 shows the situation just before confirmation of an empirical CDF table by pressing the button “Check & Confirm”. Inversion is used for generation that can be either discrete or interpolated. The technique of restricted inheritance (simplification) mentioned earlier was used to declare a simplified version of this class for generation of discrete random numbers with

empirical distribution only. Its instances have been used for example to represent random movements of customers in queueing networks. Work with random numbers is very simple. Instances are first created by statements similar to the following one located typically in the so-called head code that is interpreted during loading of the document:

```
var arrival=new Distribution("a1");
```

The method `generate()` returns the random values, so during simulation statements similar to the next one are used:

```
var x = arrival.generate();
```

So far the standard JavaScript random generator `Math.random()` is used.

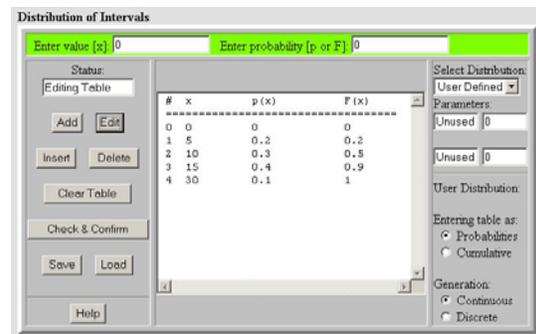


Figure 1: Entering parameters of a random variable

2.3 Transparent statistics

Transparent collection of statistical data and simple statistical analysis are implemented by the classes *Accumulator* and *Tally* (Simsript IITM terminology is used). They differ in time treatment. *Tally* ignores the time; the statistics is based on the collection of the assigned values only. *Accumulator* statistics is based on time integrals. Basically they are both real variables with transparent collection of statistics. The consequence for the user is the difference in the form of the assignment statement. The usual `a = x` has to be replaced by a method call `a.updateto(x)`. Simsript IITM calls this mechanism *left monitoring*. It is based on the idea suggested by [McNeley, 1968] who used the name *Store Association*. Both *Tally* and *Accumulator* objects have methods that return standard statistical figures like `a.average()` that are used during experiment evaluation without any further programming.

2.4 Advanced data structures

To implement the sequencing set that is conceptually an ordered list of event notices the *heap* class has

been implemented. Heap (not to be mixed with the dynamically allocated memory of some languages) is a perfectly balanced binary tree stored in an array with the following properties assuming ascending ordering of items by a certain key:

- The root with minimum key is at the position 1
- The two children (if any) of a node at the position i are at the positions $2i$ and $2i+1$
- Both children have bigger (or equal) key than the parent.

Heap supports two basic operations *adding an item* and *removing the first item*. These operations have the performance of $O(\log_2 n)$ where n is the number of items in the heap. Heap is also intended to be used as a priority queue. For more details see [Sklenar, 2001b].

JSSim also contains classes that implement a linked list and a generic statistically monitored queue without any specific ordering. Using these two classes as superclasses, programmed multiple inheritance was used to define classes for FIFO and LIFO queues. Due to the superclasses used in multiple inheritance, the queues can have practically unlimited length and methods are available that return typical statistical results like average, standard deviation, and maximum of the queue length. Methods that perform the basic operations have the same name. Loosely typed JavaScript is polymorphic, so the same code is used to work with various types of queues.

2.5 User-friendly Input and Output

Validated input is easily implemented by JavaScript code associated with text areas in the HTML document. JSSim contains various validation routines to check for example that the user has entered a syntactically correct non-negative number. The technique is known to everyone who has filled in any on-line form. In addition to validation it is also possible to update model parameters accordingly. This can simplify model initialization when simulation is started. The following HTML fragment together with the associated page contents represents a validated input of a probability value. The routine checks non-negativity and whether the value is not bigger than 1. Note that 0 is restored in case of wrong input.

```
Enter probability [p or F]:
<INPUT TYPE="text" NAME="GIpx"
SIZE=15 VALUE="0.0" ONCHANGE="if
(!testNonnegLE1Value(GIpx.value))
{GIpx.value = 0}">
```

Enter probability [p or F]:

Model parameters can be updated directly when the user enters the values or alternatively it is possible to link objects to HTML text fields and to write methods that read the validated data before simulation starts. This direct link has so far been utilized for outputs. The idea is as follows. The link is done by common names. So assume that a queue instance has been created by calling its constructor:

```
queue = new FifoQueue("Q1");
```

The constructor creates and initializes the queue, the name is stored to the property `qname`. The instance has two output methods inherited from statistically monitored queue. The first method is used to update the contents of the host HTML document:

```
StatQueue.prototype.scrupdate =
function(dname) { with (this) {
eval(dname + qname +
"av.value = average()");
eval(dname + qname +
"ma.value = maxqlength");
eval(dname + qname +
"sd.value = stdDev()");
}};
```

Note that the method `scrupdate()` updates three text fields (typically in a table with results) that would contain the average length, the maximum length, and the standard deviation of the length of the queue. Assume that the method is called as follows:

```
queue.scrupdate("document.form1.");
```

So for the average and with respect to the above example the procedure `eval` is given and evaluates the parameter:

```
document.form1.Q1av.value=average()
```

This updates the text field called `Q1av` on the screen. The following is the HTML fragment together with the associated page contents:

```
<TH> Average </TH>
<TD><INPUT TYPE="text" NAME="Q1av"
SIZE=25></TD>
```

Average	0.6672205803062008
---------	--------------------

So far it is the user's responsibility to keep the compatibility of names. Here it is the name of the JavaScript object `Q1` that is linked to the HTML text field called `Q1av`. This can be achieved by using standard HTML templates processed by the "Replace All" operation available in practically all

text editors. In this case a template displaying typical queue statistics would be used. Another method `winupdate()` generates an HTML fragment that displays four lines with results:

```
StatQueue.prototype.winupdate =
function(stitle,w) { with (this) {
  w.writeln(stitle +
    " length statistics:" +
    "<BR><UL>");
  w.writeln("<LI> Average: " +
    average());
  w.writeln("<LI> Maximum: " +
    maxqlength);
  w.writeln("<LI> Std Dev: " +
    stdDev() + "</UL>");
}};
```

The method `winupdate()` is used for generation of results in textual format in a separate window. Assuming that there is an open window `resw` the method is activated as follows:

```
var d = resw.document; ...
queue.winupdate("Queue",d);
```

The generated output can then be copied and pasted into other documents as it has been done here:

Queue length statistics:

- Average: 0.6672205803062008
- Maximum: 10
- Std Dev: 1.4089986068350546

3 EXAMPLE SIMULATION

There are several simulation models created by using JSSim that are available on the web. One of them is a general simulator of queuing networks whose last version is available at: <http://staff.um.edu.my/jskl1/simweb/net2/netmain.html>. This model has been used to compare capabilities of JSSim with two professional simulators that both can be characterized as Visual Interactive Modelling Systems (VIMS) [Pidd, 1998]. Academic version of Arena™ (Rockwell Software Inc.) is distributed with the book [Kelton et al., 2002]. It is a general discrete simulation tool oriented to simulation of queuing systems. Extend™ (Imagine That Inc.) is a general tool for both continuous and discrete simulation. Its demo version can be downloaded from <http://www.imagethatinc.com/>.

The simulated system is a network made of two generators of customers and four network stations. Figure 2 made of self-explaining blocks is a network created by Extend, Arena chart is similar. The network works as follows: after generation the customers enter randomly any of the four service stations, all with the same probability. After being served the customers either leave the network or

move to any of the four stations, all five options have the same probability.

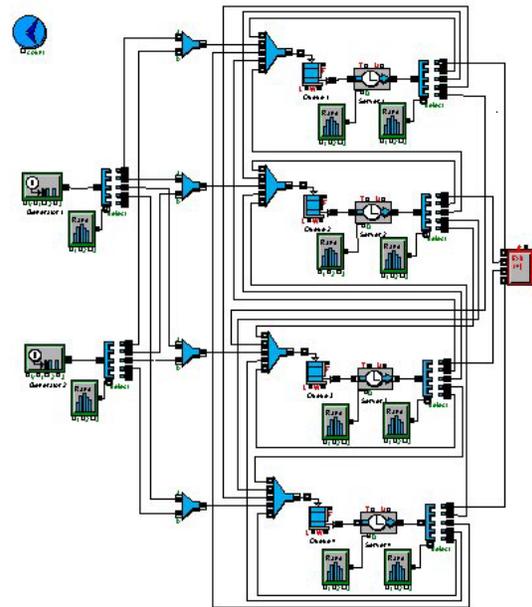


Figure 2: Example queuing network in Extend™

The two generators are for simplicity equal with exponential distribution of intervals with the mean value 10 time units (let's assume minutes). The service stations are also equal, all made of one FIFO queue of unlimited capacity and one server with exponential distribution with the mean value 2 minutes. All these assumptions can be easily modified. Having these input data, the first task was to create the models. In Arena and Extend it means drawing the networks and entering parameters of blocks. Describing all details is out of the scope of this paper, but the work is easy and can be done with just the basic training. In JSSim simulator there is no drawing. The distributions are entered in tables like the one in Figure 1. Server parameters are also entered in tables, this model uses defaults, so actually nothing is entered. The first criterion is the ease of creating the model. While drawing a network is certainly a good way for beginners and for education, it is a real nuisance in case of routine work with non-trivial models. Creating the model in JSSim simulator by entering data into tables was much faster than drawing a network like the one in Figure 2. Moreover equal distributions are entered only once, then saved to cookies, and loaded for other blocks – see the “Save” and “Load” buttons in Figure 1. JSSim simulator and Arena provide all typical results for the generators, the queues, the servers and the whole network as such, in particular the average time spent by a customer in the network. In Extend measuring this time has to be incorporated in the model, that is not included in Figure 2. The following text has been copied from a

report generated by the JSSim simulator in a separate window. These are system results for one particular 60000 minutes long experiment rounded to 3 decimal places.

Number of arrivals : 11787
 Number of lost customers : 0 (0%)
 Number of departures : 11778
 Average time in network : 19.855
 Network time standard deviation : 21.367
 Minimum time in network : 0.001
 Maximum time in network : 234.360

The following are the results for the first service station with the CDF table removed:

Server # 1

Exponential service duration, Mean=2
 Routing of departures: (removed)
 Number of channels : 1
 Unlimited queue, FIFO organization
 Number of arrivals : 14809
 Number of not waiting arrivals : 5948 (40.16%)
 Number of lost customers : 0 (0%)
 Number of services : 14808
 Average service duration : 1.997
 Minimum service duration : 0.00007
 Maximum service duration : 19.646
 Average waiting time : 1.902
 Waiting time standard deviation : 3.352
 Average non zero waiting time : 3.178
 Maximum waiting time : 35.340
 Average time in server : 3.899
 Time in server standard deviation : 3.890
 Minimum time in server : 0.00007
 Maximum time in server : 38.558
 Average queue length : 0.469
 Queue length standard deviation : 1.065
 Maximum queue length : 13
 Utilization of server(s) : 0.493

Results from Extend and Arena models are very similar with variations given by different sequences of random numbers. The second and the most important criterion is the speed. Table 1 shows the typical duration of an experiment of the length 60000 minutes for various simulators on the same computer (PII, 350MHz, 128MB, Windows/Me) in single task mode. JSSim engine measures the time exactly, duration for the other two was measured by stop watches. There is some variation, figures in Table 1 are averages taken from several runs. Though the precision of the measurements is not very high, for the comparison they are sufficient. It is no surprise that JSSim's interpreted code is slower than the other two compiled and optimized simulators. The speed of the JSSim simulator in Internet Explorer 6 is in fact a pleasant surprise. The results show clearly that using JSSim it is possible to

create at least medium size models that run fast enough to enable long experiments or repetitions to reduce the variance of the results.

Table 1: Speed of queueing networks simulators

Simulator	Duration [s]
Extend 4.1	32
Arena 5.0	55
JSSim (IE 6.0)	78
JSSim (NC 4.75)	135
JSSim (NC 7.02)	340

CONCLUSION

JSSim is a result of experimentation with concrete models. Its facilities were added gradually according to concrete problems that had to be solved. In addition to facilities listed so far there are also various utilities like displaying a help window, work with cookies, etc. Its next development will be oriented to enhancement of random numbers and especially to the definition of more complex standard classes like complete multichannel servers. The simulators implemented so far are used mainly for education, but they generated also interest from professional organizations. Simulators are freely available for direct use and for download at <http://staff.um.edu.mt/jskl1/simweb/>.

REFERENCES

Darnell R. et al. 1998, "HTML 4 Unleashed. Professional Reference Edition", Sams.net Publishing.
 Eckel B. 1998, "Thinking in Java", Prentice Hall. Inc.
 Flanagan D. 1998, "JavaScript - The Definitive Guide", O'Reilly & Associates, Inc.
 Kelton W.D. et al. 2002, "Simulation with Arena", McGraw-Hill.
 McNeley J.L. 1968, "Compound Declarations". In: *Proceedings of IFIP Working Conference on Simulation Languages*, Oslo, May 1967. North Holland, p.292-303.
 Pidd J. 1998, "Computer Simulation in Management Science", John Wiley & Sons.
 Sklenar J. 2001, "Interactive Simulators in JavaScript". In: *Proceedings of 15th European Simulation Multiconference ESM2001*, Prague, p.247-254.
 Sklenar J. 2001, "Client Side Web Simulation Engine". In: *Proceedings of 27th ASU Conference Model Oriented Programming and Simulation*, Rättvik, Sweden, p.1-13.
 Sklenar J. 2002, "Discrete Event Simulation in JavaScript". In: *Proceedings of 28th ASU Conference: The Simulation Languages*, Brno, Czech Republic, p.115-121.

PREDICTION OF LINK TRAVEL TIMES IN THE CONTEXT OF NOTTINGHAM'S URBAN ROAD NETWORK

JOANNA K. HARTLEY

*School of Computing and Mathematics, The Nottingham Trent University
Burton Street, Nottingham, NG1 4BU, U.K.
Tel. +44 (0) 115 848 6172, Fax. +44 (0) 115 848 6518
Email: Joanna.Hartley@ntu.ac.uk*

Abstract: Traffic congestion is becoming a serious environmental threat that must be resolved quickly. Traditionally, travel information systems have been specific to a particular mode of transport. For instance, traffic information (road conditions broadcast) has been directed at drivers. Instead, travel information systems are now being developed which incorporate route guidance systems to divert drivers away from the congested areas either by change of travel mode or travel route. The mobile travel information system developed at The Nottingham Trent University enables progression from a passive mode of interaction between traffic control systems and road-users (one-way flow of information) to an active mode. The integration of data concerning traffic flows and individual journey plans thus makes it possible to perform optimisation of travel. This paper focuses on the issue of provision of real-time information about urban travel and assistance with planning travel. Nottingham's SCOOT (Split Cycle Offset Optimisation Technique) traffic-light control system provides real-time information about the link travel times within certain areas of the city. However, rather than using link travel times at the time of the request, it is more effective to predict the link travel times for the time of travel along the particular links. The future link travel times depend upon the historical travel time of the link (for the specific time step in the day) as well as the current link travel time. Consequently, the link weights are a combination of real-time data, historical data and static data. The prediction method will be validated in the context of Nottingham's urban road network. The results will be presented at the conference.

Keywords: Transportation, Optimisation, Efficiency, Prediction methods

1. BACKGROUND

Traffic congestion is becoming a serious environmental threat that must be resolved quickly. Great Britain has become a role model in the battle against global pollution. The Prime Minister, Tony Blair, has acknowledged that a 20% reduction in carbon dioxide emissions in Great Britain is a credible target for the year 2010 [Brown, 1997]. However, significant measures are necessary to attain this target. Road vehicles and industry are the main sources of pollutant emissions. In the United Kingdom, road vehicles are responsible for over 50% of the emissions of nitrogen oxides and over 75% of carbon monoxide emissions [DETR, 1998]. Congestion is already a major problem in many areas and traffic volume is set to grow by 30% in the next 20 years [MacAskill, 1999].

Indeed, in 1996-1998, the average person in Great Britain made over 1000 journeys per annum. This is a substantial increase when compared with the figure of 742 journeys made per person per annum in the period 1992-1994. Although the average citizen is clearly becoming more mobile, the distance travelled by an average person in a year has not changed significantly. The greatest average distance travelled per person per annum is for the

purpose of commuting, closely followed by visiting friends at home [Keynote, 1999]. The largest increases in the number of journeys per person are connected with education. By optimising such journeys, the cumulative emission of vehicles will be curtailed.

This paper describes the infrastructure that is currently being developed at The Nottingham Trent University to facilitate multi-modal travel throughout the city of Nottingham. This paper focuses on the issue of provision of real-time information about urban travel and assistance with planning travel. This includes consideration of uncertainty about traffic delays, inconvenience of parking and the variability of travel time along urban links.

2. TRAVEL INFORMATION SYSTEMS

Traffic/travel information is inherently heterogeneous and is characterised by varying levels of information granularity: from detailed lane occupancy data through to video traffic surveillance to travel times of specific journeys using various modes of transport. Recent advances in computing and communications have made it

feasible to access all of these important sources of information. However, the integration of this information for the purpose of optimising urban travel with respect to various environmental, social and economical criteria remains an unanswered challenge.

There are a number of ways of informing travellers about the location of congestion areas – such as, radio or television broadcast and variable message signs. The growing body of opinion, that the traditional forms of supervisory control are both too expensive and inaccurate, prompts new development. The traditional forms require full involvement of a human operator. However they do not take into account the specific requirements of individual journeys. In particular, because of the protection of privacy, the crucial information about the intended destinations of individual vehicles is not normally available to these controllers and, even if it was, it could not be processed efficiently. On the other hand, an attempt to delegate the responsibility for journey optimisation to road users by informing them (through radio broadcasts or variable message signs) about the best routes, that are relevant to various journeys, is bound to be counterproductive because of the resulting information overload.

Traditionally, travel information systems have been specific to a particular mode of transport. For instance, traffic information (road conditions broadcast) has been directed at drivers. While fulfilling its intended objective, such an approach to urban travel support does little to encourage multi-modal travel in the cities.

To eliminate these constraints, this research project takes a fundamentally different approach and rather than aiming at maximising the efficiencies of the use of individual modes of transport taken in isolation, it considers a broader multi-modal travel framework. Travel requirements are defined in terms of journeys and the mode of travel is just one of the decision variables. An important feature of our approach is that it recognises the individual nature of journeys. The enquirers are able to select their journey according to their individual preferences, such as the importance of a short journey time or the importance of timely arrival at the destination. Although the answers may be highly subjective it must be remembered that it is precisely these preferences that make people opt for one mode of transport or another. In this sense, multi-modal travel optimisation offers good mapping onto human decision-making.

2.1 Conventional Systems

Travel information systems are now being developed which incorporate route guidance systems to divert drivers away from the congested areas either by change of travel mode or travel route. Dynamic guidance is of the greatest benefit to travellers, as new routes and travel modes can be suggested as conditions change [McDonald and Montgomery, 1996]. The system will aid the road users by providing access to information that is not readily observable from the current location of the traveller, yet is relevant because of the planned journey. The wide use of such a system will reduce the amount of congestion within the city, by the choice of departure time and shorter routes as suggested by the route guidance system. This will lead to expected reductions of pollutant emissions in currently congested and critical areas.

The system enables progression from a passive mode of interaction between traffic control systems and road-users (one-way flow of information) to an active mode. Within the active mode, the road users supply the information about their intended destination (without disclosing their identity) and, in response, receive customised traffic information that optimises their journeys.

Traveller information solutions aimed at Internet users have been developed. However the Internet's communication delays cause problems due to the dynamic nature of urban traffic and with access limited to the home/office or the few available travel information terminals, the accuracy and the relevance of the advice is limited. At the other end of the spectrum, travel information systems incorporated in top-of-the-range cars, offer excellent static travel information but are limited regarding the real-time traffic and public transport data provision and as such do not contribute directly to mode switching decisions by drivers [Bargiela et al., 1999].

2.2 Real-Time Systems for Mobile Users

The system must be capable of simultaneous data acquisition, processing and dissemination of the traffic/travel advice in real-time to a full spectrum of end users. Preston et al. [1993] proposed the use of in-vehicle telephones to communicate with a remote computer. The increasing use of mobile phones by the general public takes their suggestion one step further, opening access to the decision support system to many more users. The integration of data concerning traffic flows, public transport and individual journey plans thus makes it possible to perform multi-modal optimisation of travel.

So, there is a need for a hierarchical urban sustainability structure that would be specifically responsible for providing a global optimisation layer while relying on the local optimisations affected by the individual organisations responsible for urban transport [Peytchev and Bargiela, 1998; Pursula, 1998]. The development of such a structure should clearly rely to a maximum extent on the standard computer and public communication systems. However, the feasibility of such an undertaking has to be proven by detailed consideration of the technological constraints of the sub-systems that are to be integrated.

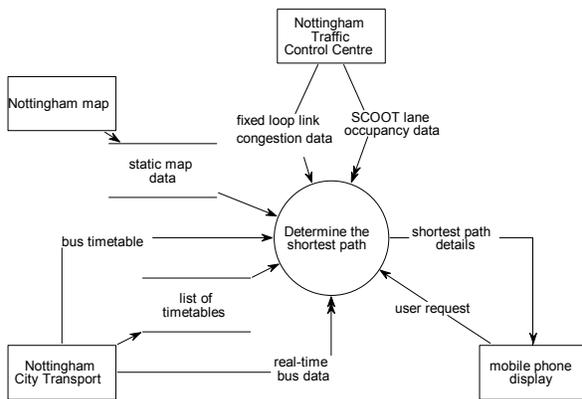


Figure 1. Determining the shortest path

The developed structure is a Distributed Memory Environment (DIME) [Peytchev and Bargiela, 1998] that manages the data from a number of sources (Traffic Control Centre, public transport company and the user) (Figure 1). Nottingham Traffic Control Centre continues its kind agreement of allowing the Intelligent Simulation and Modelling group to have access to its Traffic Control System (SCOOT and congestion data), providing the necessary current traffic information. Nottingham City Transport has obligingly approved the use of the necessary information concerning their bus timetables.

The user communicates with the DIME system via a mobile phone. The advantage of a mobile phone is that there do not exist the constraints of being part of a car's equipment or being deployed at a particular location. Along with the increasing use of mobile phones by the general public, this means that the system has a much broader user base. By implication, this will result in a much greater impact on travel mode switching decisions [Bargiela and Berry, 1999]. Also, the system is easy to use and not prohibitively expensive.

3. ROUTE GUIDANCE

There are many methods of path finding that are appropriate for use within the spectrum of route guidance. Some of these methods have been considered and evaluated in the context of multi-modal travel in Nottingham's urban network. The results are published in [Hartley and Bargiela, 2001; Hartley, 2003]. This paper concentrates on the delivery of timely route guidance given the available real-time traffic information.

3.1 Urban Network Information

The pre-requisite to determining any shortest path in an urban network is having information about the road link weights (Figure 4).

The necessary input data, mostly provided by the different organisations managing the urban network, comprises SCOOT (Split Cycle Offset Optimisation Technique) link data, congestion data for fixed loop links and static map data (Figure 1).

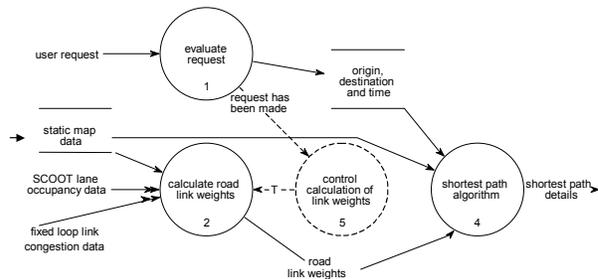


Figure 4. Calculation of travel along links

SCOOT is an intrinsic part of Nottingham's traffic-light control system comprising induction loops that detect the presence of vehicles in real-time. The SCOOT link data provide real-time information about the link travel times within certain areas of the city. Fixed loop congestion data again provide real-time information – these data are specific to certain junctions or roads (distinct from the SCOOT-managed areas). The static map data include information about the topology of the urban network and the length of roads. The integration of SCOOT lane occupancy data (leading to link times in SCOOT-managed areas), fixed loop congestion data (leading to link times in some non-SCOOT areas) and static map data (providing estimated static data of the link times for the remainder of the network) are manipulated into up-to-date, reliable information of alternative paths and adverse traffic conditions on appropriate links. This enables the derivation of the optimal route for travel by car.

3.2 Travel by Car

Dijkstra's algorithm [1959] has been used to determine the optimal route by car. Dijkstra's algorithm builds an expanding list of examined vertices and looks at paths through vertices on the list. The path with the smallest total of link weights is incrementally found.

Dijkstra's algorithm

```

pathlength(all links) = ∞
marked(all links) = false.
marked(origin) = true.
pathlength(origin) = 0
do for all links until marked(destination) = true.
{
  search for all pairs of nodes s.t.
  marked(node1) = true. & marked(node2) = false.
  then
    pathlength(node2) = min[pathlength(node2),
      pathlength(node1)+length(node1,node2)];
  from this set determine which node2 has minimum
  pathlength then
    marked(node2) = true.
}

```

Rather than using link travel times at the time of the request, it is more effective to predict the link travel times for the time of travel along the particular links. The method used was developed as part of the Ali-Scout project [Kotsopoulos and Xu, 1993]. The future link travel times depend upon the historical travel time of the link (for the specific time step in the day) as well as the current link travel time. The process is as follows:

$$D = \frac{T_h(l, n)}{T_{cur}(l, n)} \quad (1)$$

where $T_h(l, n)$ is the historical mean travel time of link l at time step n and $T_{cur}(l, n)$ is the current (time step n) travel time at link l .

$$T_p(l, m) = \frac{T_h(l, m)}{D} \quad (2)$$

where $T_p(l, m)$ is the predicted travel time of link l at future time step m .

As the state of the network can change extensively in a short period of time, a combination of real-time data and historical data should be used, with the proportions dependant on the expected time taken to arrive at the measured link [Kotsopoulos and Xu, 1993]. Kotsopoulos and Haiping [1993] propose information discounting:

$$T_p(l, m) = a * T_h(l, m) + (1 - a) * \frac{T_h(l, m)}{D} \quad (3)$$

where a is an increasing function of $(m-n)$, say $e^{(m-n)}$.

Consequently, the link weights are a combination of real-time data, historical data and static data (Figure 5).

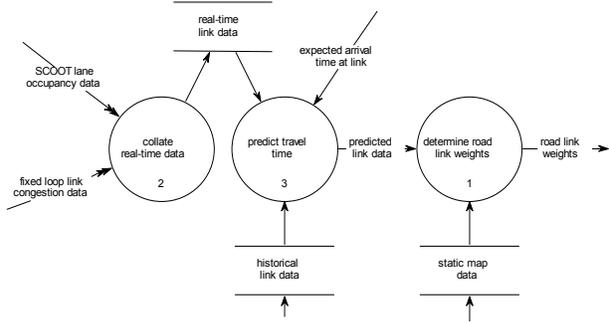


Figure 5. Combination of historical, current and static data

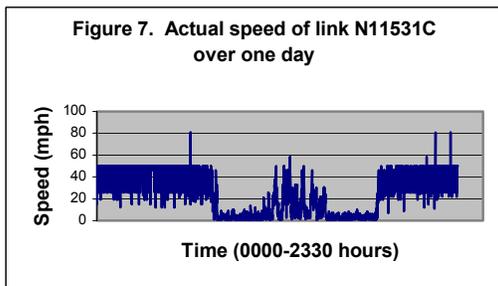
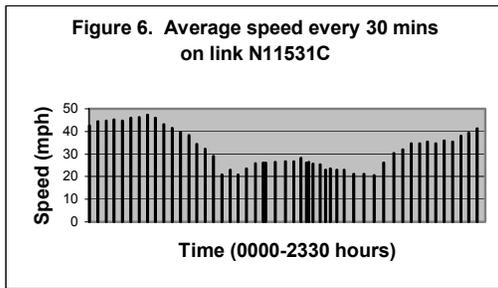
4. REAL URBAN TRAFFIC NETWORK APPLICATION

Currently, the available real-time information in the Nottingham urban network is collected from SCOOT detectors, which monitor highly traversed links within the city centre and the arterial routes into the city. 100 links out of 2018 are currently equipped with SCOOT inductive loop detectors. The SCOOT data consists of a large amount of traffic control information relayed in the form of messages [Siemens PLC, 1997] (which include information about flow, occupancy, delay and speed etc.). The U06 message provides information every 30 seconds about the average point-speed of a car travelling along a link (measured over the last 5 minutes). This speed and knowledge of the link length is used to estimate the current travel time of the link.

5. RESULTS

As some of the routes across Nottingham may take up to one hour to traverse, it is not sufficient to use the current travel time estimations. So, instead predictions of travel time will be used (as described in section 3.2). The available historical U06 messages will be used to determine the validity of the prediction method in the context of Nottingham's urban network. The results will also show how the incorporation of real-time information routes traffic away from congested areas. It will also be shown that the method is

capable of dealing with the transition between peak and off-peak conditions.



Figures 6 and 7 show that the historical speed cannot be relied upon, as the speeds fluctuate even on a single link on a single day.

The efficiency of the algorithm is of paramount importance, so that the information provided to the user is timely and thus relevant. For Nottingham's network of 597 nodes and 2018 links, Dijkstra's algorithm has a system run-time of 0.4 seconds. The execution time of the predictive route finder algorithm should not be greater than a few seconds.

All of the above results will be presented in detail at the conference.

6. DISCUSSION

It should be noted that minimisation of travel time by car does not necessarily produce the optimal route from 'door-to-door'. With the increasing ownership of cars, there is more demand on the limited number of parking spaces within any city. Consequently, the inconvenience of parking can make travel by car be less preferable especially for those travellers who are particularly adverse to travel time uncertainty. So users may ultimately be encouraged to travel by public transport instead. This becomes especially apparent when travel advice includes both car and public transport as modes of transport, as is the case in the developed real-time travel information system detailed in section 2.2.

7. CONCLUSIONS

This work is part of a study exploring the provision of traffic and travel information through mobile communications. This paper has clearly shown that the communications infrastructure has been successfully implemented.

This paper demonstrates the necessity of real-time information when providing traffic/travel information to the general public. The use of real-time data provides the user with information about the state of the network, not normally foreseeable by the traveller.

8. FUTURE WORK

The car guidance system should be tested on a suitable simulator before being implemented in the real world. The Simulation and Modelling Intelligence group at The Nottingham Trent University [Peytchev and Bargiela, 1994; Peytchev and Bargiela, 1995] has developed a suitable microscopic traffic flow simulator. Any private vehicle route guidance system must consider the inherent implications for the rest of the urban traffic network. It must be ensured that the movement of traffic from one area of the network does not result in congestion of another area of the network. Investigation into induced traffic has already shown that road improvements will normally result in a traffic growth of 10% in the short term and 20% in the longer term [Goodwin, 1996].

Studies have shown that the acceptance of route guidance is strongly correlated to any previous experience of the system. Simulation will be used to test how the use of the route guidance system will enhance the progression of the traveller [McDonald et al., 1995].

Due to the large amounts of static and real-time data that will be used by the path finding algorithm, there are a number of issues to investigate with regard to storing information. The appropriateness of storing set paths, or calculating paths on demand will be investigated. The pruning of the urban network will be necessary – this may be achieved in a pre-processing mode or as part of the algorithm. Also, further analysis of multiple users (of the order of 100) will need to be considered to continue to provide a viable service. Some possible long-term solutions are the use of more processors, parallel algorithms, or some form of artificial intelligence (such as neural networks).

REFERENCES:

Bargiela, A., Berry, R., 1999. "Every BIT counts", Traffic Technology International, Feb/Mar, pp 63-66.

Bargiela, A., Peytchev, E., Berry, R., 1999. "Experiences with a distributed traffic telematics environment – portable travel information system", Proc. of IEEE Africon '99 Conference, September, SPEC217.

Brown, P., 1997. "Britain's Green Lead at UN", The Guardian, 24 June 1997.

DETR, 1998. "Air Pollution – What it Means for Your Health", Air and Environmental Protection, <http://www.environment.detr.gov.uk/airq/aqinfo.htm>.

Dijkstra, E.W., 1959. "A Note on Two Problems in Connection with Graphs", Numerische Mathematik, 1, pp269-271.

Goodwin, P.B., 1996. "Empirical Evidence on Induced Traffic", Transportation, Vol. 23, pp. 35-54.

Hartley, J.K., 2003. "Efficiency vs. Correctness of a Travel Information System", *Proc. UKSim 2003*, April 2003, ISBN 1-84233-088-8, pp 214-219.

Hartley, J.K., Bargiela, A., 2001. "Decision Support for Planning Multi-Modal Urban Travel", Proc. of 13th European Simulation Symposium, Marseille, October 2001, ISBN: 90-77039-02-3, pp 387-391.

Keynote, 1999. "Passenger Travel in the UK", www.keynote.co.uk, Ed. Jane Griffiths, November 1999, ISBN: 1-84168-011-7.

Kotsopoulos, H.N., Haiping, X., 1993, "An Information Discounting Routing Strategy for Advanced Traveller Information Systems", Transportation Research Part C, Vol. 1, No. 3, pp 249-264.

MacAskill, E., 1999. "Promise of a better, faster, more reliable system", *The Guardian*, 14 December 1999.

McDonald, M., Hounsell, N.B., Njoze, S.R., 1995, "Strategies for Route Guidance Systems Taking Account of Driver Response", Pacific Rim Trans. Tech. Conf., 1995 Vehicle Navigation and Info. Systems Conf. Proc. 6th International VNIS, pp. 328-333.

McDonald, M., Montgomery, F.O., 1996. "Urban Traffic Control In Europe", Proc. Instn. Civ. Engrs. Transp., Vol. 117, February, pp 50-56.

Peytchev, E., Bargiela, A., 1994, "Micro Simulation of City Traffic Flows in Support of Predictive Operational Control", 10th Int. Conference on Systems Engineering, ICSE '94, Coventry.

Peytchev, E., Bargiela, A., 1995, "Parallel Simulation of City Traffic using PADSIM", Proceedings of Modelling and Simulation Conference ESM'95, Prague, Eds. Snorek, Suhansky, Verbraeck.

Peytchev, E., Bargiela, A., 1998. "Traffic Telematics Software Environment", *Proc. European Simulation Symposium*, Oct. 1998, ISBN 1-56555-147-8, pp 378-382.

Polenta, T., Hartley, J.K., 2003. "A comparative Study of Stochastic 'Least-Time' Path Algorithms in the Context of the Nottingham Urban Network", *Proc. UKSim 2003*, April 2003, ISBN 1-84233-088-8, pp 194-200.

Preston, J.M., May, A.D., Aldridge, D.M., 1993, "The Specification of Trip Planning Systems", IEE Colloquium on 'Electronics in Managing the Demand for Road Capacity', pp. 2/1-4.

Pursula, M., 1998. "Simulation of Traffic Systems - An Overview". Proc. 10th European Simulation Symposium, Nottingham Trent University, October 1998, pp 20-24.

Siemens PLC, 1997. 'SCOOT User Guide', Poole, Issue 17.

BIOGRAPHY:



JOANNA HARTLEY was awarded a BSc (Hons) degree in Mathematics at the University of Durham in 1991. In 1992, she became a research assistant in the Department of Computing at The Nottingham Trent University and was awarded a PhD in 1996. The title of her PhD is "Parallel Algorithms for Fuzzy Data Processing with Application to Water Systems". She is now a senior lecturer at The Nottingham Trent University and an active member of the Intelligent Simulation and Modelling group. She is a member of the UKSim committee and was an associate editor of UKSim 2003. Her current research interests include parallel processing, mathematical modeling and probabilistic state estimation relating to urban traffic networks and water distribution systems.

**OPERATIONAL
RESEARCH
&
V-V / METHODOLOGIES**

TOWARDS COMPOSABLE SIMULATION: SUPPORTING THE DESIGN OF ENGINE ASSEMBLY LINES

ANDREW WINNELL* and JOHN LADBROOK**

**Lanner Group.*

*The Oaks, Clews Road, Redditch, Worcestershire, B98 7ST. UK.
Tel: +44 (0) 1527 551327; e-mail: awinnell@lanner.co.uk*

***Ford Motor Company,*

*Dunton Engineering Centre, Basildon, Essex, SS15 6EE. UK.
Tel: +44 (0) 1268 401663; e-mail: jladbroo@ford.com*

Abstract: This paper introduces the latest collaborative project work being carried out by Consultants at Lanner Group and Manufacturing Engineers at Ford Motor Company's PowerTrain Operations. The work involves aiding the design and optimisation of planned Engine Assembly Lines, by generating and experimenting with a complete simulation model produced using a spreadsheet interface, re-using and adapting a constantly evolving portfolio of modelling components. Such a methodology can be used to enhance quality and consistency in a process of continuous model validation and verification by using tried and tested building blocks. The re-use of modelling components, or 'modules', also helps to control and reduce model build time, a factor increasing in importance as automotive manufacturers strive to constantly reduce overall lead time and cost-to-market. The ease of use of the spreadsheet interface, coupled with the enhanced efficiency inherent in a modular approach to simulation modelling, empowers specialists and non-specialists alike to meet targets when designing and implementing complex processes. Modular Simulation is contributing to improved Business Process Management at Ford PowerTrain – improvements that, at the time of writing, are being rolled out globally.

keywords: Automotive, Business Process Management, Composable Simulation, Methodology, Reuse.

1. INTRODUCTION

This paper introduces the latest collaborative project work being carried out by Lanner Group Consultants and Manufacturing Engineers at Ford's PowerTrain Operations (PTO). A unique feature of the on-going consulting relationship between Lanner Group and Ford PTO is the reuse of a constantly evolving suite of modelling components, or 'modules'. Simulations composed using this portfolio of modules via a spreadsheet based front-end, have the advantages of being quicker, less costly to construct and maintain and easier to validate with a greater degree of confidence, as well as being more accessible to the engineers employed in the design and implementation of assembly lines.

After first giving a brief introduction to simulation at Ford PTO and their relationship with Lanner Group, this paper will explore some of the arguments surrounding Composable Simulation that have been put forward in recent years. The methodology enabling successful implementation of Composable

Simulation will then be detailed. Finally, concluding remarks and potential future directions will be given.

1.1. Background

Ford PTO has used simulation for over 20 years. In that time, significant progress has been made, not only into the process design issues themselves but in the simulation methodology employed to make these improvements. Ford are using the latest technology developed by Lanner Group, a UK based specialist Simulation company. Their WITNESS simulation system is used by Ford throughout the world to model new and changing facilities in order to answer such questions as "What is the throughput achievable for a line?" and "How large should a buffer storage area be?"

The choice of tools, support and expertise deployed, have been the focus of a continuous drive to raise the awareness of, and thus utilization by, non-'simulationists'. [Ladbrook, 2001]. Another key area is that of increasing the availability of these resources to *all* at Ford PTO able to benefit from them. Several systems have been developed enabling simulation

models to be created automatically via spreadsheet entry by Ford engineers. This effectively makes the simulation model easier to construct for an engineer, by using an interface that explains the data required from them in the form that is most readily understood. The input of simple data, indicating operation dimensions, in the spreadsheet places the next operation in the assembly line relative to the current operation. An entire production loop is created automatically by input of the data into a WITNESS model shell, the whole process being controlled by Visual Basic (for Applications) and WITNESS' own command language; creating a model dynamically and visually in real-time.

1.2. A New Assembly Line

The latest project work undertaken by the Lanner Group in collaboration with Ford PTO concerns the design and implementation of a new engine assembly line. One of the chief concerns of the work was to drive down the time taken to achieve successive levels of model development. Ford PTO comprises a subset of the complex interlinked and *interdependent* processes of Ford's Supply Chain. Timely decision making is thus required by Ford PTO to ensure they meet their commitments to Ford as a whole, who in turn are constantly striving to reduce time-to-market. Some have referred to this high level of interdependence as a 'House of Cards'.

Many of the building blocks that comprise this new line, or rather, a potential model of this new line, already existed. Much of the project thus comprised the adaptation of these modules to incorporate new production philosophies, and to ensure their continued interoperability throughout. In the following section we will discuss Composable Simulation with particular reference to this project. We will then detail the methodology followed by developers at Lanner and at Ford, and the steps followed by manufacturing engineers at Ford when building subsequent models.

2. COMPOSABLE SIMULATION

Composable Simulation can be considered a subset of the wider field of software reuse, a field with some considerable effort devoted to it: *"The software community has struggled with the concept of reuse for many years. Components offer a useful mechanism to support reuse. But a number of questions are raised by them as well"*. [Page and Oppen, 1999]. Ray J. Paul, in his foreword to Ezran et al. (2002), gives his approach to the reuse of programming and modelling constructs in this wider

software/information systems context: *"To make the model provide future software reuse, sub-models of the organisation would have to be determined, made relatively self contained, represent a recognisable part of the organisation, and be likely to be required as part of some future unknown system. Quite a tall order."* This paper will go on to detail an ongoing string of projects satisfying this demanding brief.

So what exactly is Composable Simulation? *"Composability is still a frontier subject in Modelling and Simulation"* [Kasputis and Ng, 2000]. In such a new field it is hard to find succinct definitions (even more mature fields, such as OR as a whole, struggle with this!). At a high level it could be considered: *"...a system with which simulations are created at runtime to meet the specific requirements of that run. The user specifies his needs to a system that in real time builds a simulation..."* [Kasputis and Ng, 2000]. At a lower, software/model developer oriented level, Composable Simulation involves the selection of a series of existing modelling constructs, bringing them together in such a way as to model the real world situation at hand, in much the same way as existing modelling methodologies – but with much of the underlying coding already carried out. The literature in the area suggests that Composable Simulation represents something of a panacea. Why?

2.1. Time and Money Benefits

Much of the case for Composable Simulation is inherently intuitive, especially given the relative lack of published experience in the area: *"Intuitively, component-oriented design offers a reduction in the complexity of system construction by enabling the designer to reuse appropriate components without having to reinvent them."* [Page and Oppen, 1999]. This time saving, particularly in the commercial context, has clear cost implications by potentially reducing the effort required to reach a comparable level of development of a simulation model sooner: *"Such a system offers the potential for providing higher quality simulations in less time for lower costs."* [Kasputis and Ng, 2000]. So, we see that not all of this benefit need be taken in the form of reduced costs, nor perhaps should it be...

2.2. Quality Benefits

On the issue of *quality* in simulations, there is little disagreement over the need to conduct VV&T (Validation, Verification and Testing) throughout the life-cycle of a simulation project [Balci, 1994], [Robinson, 1999] but: *"Assessing credibility throughout the life-cycle of a simulation study is an*

onerous task” [Balci, 1994]. The case for improved quality is not equally valid [sic.] across the VV&T board though; where composable simulation really contributes is in the area of Verification (‘building the model right’) as opposed to Validation (‘building the right model’). This is achieved, effectively, by extending the testing phase of successive studies – reused modules have already undergone some verification and testing in their previous role. Some verification is still carried out of course, as the more tacit/emergent properties of a module, and its relationship to models built, are experienced.

2.3. Ease-of-Use and Accessibility.

Another benefit of Composable Simulation is that afforded by the potential level of abstraction by the end-user, from the underlying code generating the behaviour they (wish to) observe. The division of labour so prevalent in just about any efficiently carried out enterprise is promoted by this abstraction, allowing a greater degree of separation of programming, modelling and, in the case of Ford PTO, engineering expertise. This promotes the efficient resolution of the real-world problem and allows the benefits of simulation to be brought to a far wider range of real-world problems, *and people*.

2.4. Difficulties and Reservations

Of course, what we have discussed herein is the *potential* of Composable Simulation to aid and promote the application of simulation, not the *actual*: “*Current capability in composability is limited*” [Kasputis and Ng, 2000].

One of the key issues that may delay the adoption of component-based approaches is the potential complexity of their application: “*As the number of candidates for reuse (composition) becomes large, the benefits of reuse (composition) become negated by the costs of storage, organisation and retrieval of candidates.*” [Page and Oppen, 1999]. The authors go on to discuss the complexity of the selection of components. A non-technical summary is perhaps most succinctly achieved with the observation that for n components, there are in the order of 2^n possible models that can be constructed from them – any search for the ‘optimal’ combination could thus not be carried out in ‘reasonable’ (i.e. not increasing exponentially with n) time.

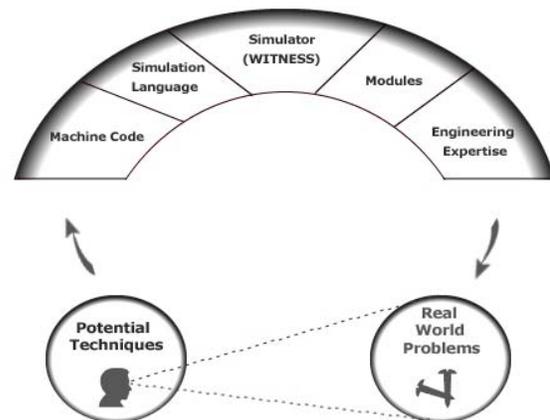
The issues facing the modeller, given a virtually unlimited choice of components, is similar to that faced by the researcher looking for information in the internet age. There is then a further complication

caused by the need to establish the suitability of candidates – a factor exacerbated by the emergent properties of combinations.

2.5. The Overall Case

Much of the argument in favour of Composable Simulation here is of course analogous to the creation and subsequent uptake of simulators (i.e. simulation software packages such as WITNESS), as distinct from simulation programming languages. In the Composable Simulation arena, however, it is possible for the Consultant/Model Developer to bridge the gap *even further* between real-world problems and the techniques brought to bear upon them – using Modules:

Fig 1. Problem and Solution; Bridging the Gap.



The Consultant/Model Developer must become adept not only in the relevant software and modelling process, but also in abstracting from their work that which can be more widely applied *in a formal manner* – that is, generating Modules. We thus have a strong case for the greater specialisation of simulation Consultants in specific industry areas.

2.6. Moving Forward

Although in the long term, the suite of Modules could support both a span of domains and a range of granularity, the difficulty of this task is widely acknowledged [Kasputis and Ng, 2000], [Page and Oppen, 1999]. Despite this, the desirability of this outcome is clear: “*In this envisioned future, simulation becomes ubiquitous.*” [Page and Oppen, 1999]. A pragmatic approach, suggesting how progress towards Composable Simulation may be made, has been given by Kasputis and Ng. (2000): “*Initial work... should deal with physical descriptions. Lessons learned in structures and*

processes can then be applied to the modelling of other aspects as they mature. It is also wise to limit the initial effort to one or a few domain areas or classes of applications."

Finally; is it not simply good general programming practice to make use of existing routines and modelling constructs, where they exist and are available to the current developer or modeller? The answer is, of course, "Yes" - The factor that separates Composable Simulation from simple good practice is that reuse of the components is a design influence from the start. It is not simply good fortune that pre-existing components exist, these constructs were developed with attention paid not only to their current purpose, but likely future use too. Indeed, the fact that modules are designed with the current purpose in mind at all is merely a result of the setting of the work - commercial necessity being "*the mother of invention*". The on-going Consulting relationship between Ford PTO and Lanner Group enables modelling to build upon previous effort, using a portfolio of modules and a tool to draw them together and construct the model. 'FAST' is just such a tool...

3. THE FORD ASSEMBLY SIMULATION TOOL (FAST)

Ford PTO are looking to apply a consistent global approach to Business Process design. The process about to be outlined, along with the technological tools and expertise required for its successful implementation, are being rolled out globally. Hand-crafting models from the start is difficult and time consuming, whereas reusing entire models is dangerous and unlikely to result in a valid model. FAST building however, takes seconds on a laptop computer, allowing the focus to remain on the issue of validity.

3.1. The Model Building Process

Developers, Consultants and Manufacturing Engineers all contribute to the finished model. Developers bridge the gap from programming to simulators. In the Composable Simulation case, Consultants/ Model Developers then build upon this by concentrating on building modules. Finally, the end-user inputs their requirements in a format tailored to their requirements, using the technology to solve the process design issues they are faced with.

3.1.1. The Developer

Developers provide the software, in this case WITNESS. Simulation packages, or simulators, require a difficult balance to be reached between

flexibility and ease-of-use; a balance for which WITNESS has been recognized as a class-leader, particularly where complex and large-scale modelling is required. [Hlupic and Paul, 1999]

3.1.2. The Consultant

Generally, it is the role of the consultant to create the model itself, delivering the finished product with appropriate documentation. In this case, to a certain extent, Consultants /Model Developers step back from this position, instead concentrating on work intended to bring the model building exercise within reach of a wide selection of users, e.g. Manufacturing Engineers. This comprises liaising with Ford PTO to establish required new functionality whilst ensuring inter-operability, and requires the consultant to take a more abstract approach than building the model directly - focusing efforts on the underlying modules, and the WITNESS code that brings them together to form the possible models to be built at run-time.

3.1.3. The Manufacturing Engineer.

It is at the Ford PTO end that requirements are formulated in terms of module and FAST shell capability - FAST is the WITNESS 'model' acting as canvas on which the model can automatically be built. Much of the VV&T effort is focussed here, with recent developments by consultants being put through their paces. A benchmark of expected overall line performance has built up over the years, and so relatively tight upper and lower bounds of expected performance can be used for black-box validation of any assembly line not already in service. At a lower level, white-box validation is carried out by those who are most familiar with the processes being simulated, although this is usually done by those also adept with the WITNESS software. Finally of course, building, running and experimenting are all carried out at run-time. Clearly, with such easily modified model structure, scenarios can be investigated with ease.

3.2. Challenges Faced.

The greater degree of abstraction from the finished simulation model presents the Consultant/Model Developer with particular challenges to overcome. Many of these were alluded to in the previous section, but are discussed here with particular relevance to the Engine Assembly Line project. In order to take account of new production procedures and philosophies, as well as build upon overall functionality, modules constantly need to be updated. The process of updating a module is itself relatively

easy, taking in the order of hours rather than days. These modules are a popular and often used feature of the WITNESS simulation software. This project however, also had to find ways of assembling these modules automatically at run-time, and in almost any conceivable permutation. The challenge here, then, is the assurance of inter-operability.

To take a very simple and frequently occurring example, modules, when loaded into the FAST shell, must be placed relative to the previous module. The varying ‘footprints’ of these modules must therefore be taken into account when assigning a location for the new module to be loaded onto. Ensuring the correct display of modules, given the number of module types, is not a trivial task. The precise position of a module is a relatively unimportant feature when it comes to the correct functioning of the model – it is more useful for subsequent analysis and communication. It is also easy to see when this form of interoperability has not been achieved – it is readily seen on the screen. Much of the rest of the functionality of the module is both more important, and less easily spotted. It is important then to take a highly incremental, methodical and organised approach, incorporating frequent testing, to building new functionality – especially where new features are replicated across the entire model.

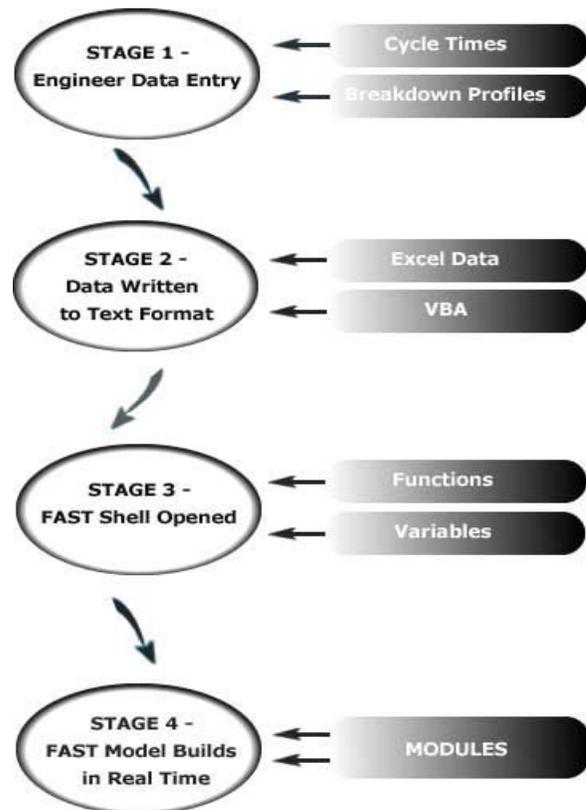
A specific new feature of the current suite of modules is the modelling of human behaviour. This is in its earliest stages, modelling certain features of human interaction with the production lines as a type of breakdown, according to an empirically defined schedule. Additionally, the model needs to keep pace with constant design changes in an iterative process of design → model → test → design. This is greatly assisted by the FAST build procedure; changes effected by Consultants/Model Developers at the modular level are automatically incorporated into all relevant parts of the model at run-time! Indeed, the potential for a Composable Simulation framework to accomplish this speed of development had already been identified: *“...there is a high probability that most or all of the representations needed for a new operation type would already exist. Therefore, simulations that operate within a composable framework would have the potential to adapt quickly to emergent operations.”* [Kasputis and Ng, 2000].

The development time for new simulations, though greatly reduced, is still significant. New functionality, as well as production philosophies and procedures, create a moving target to the modeller – flexibility creates speed, and speed enables the design process to go through more iterations!

4. CONCLUSIONS

In many ways the advances made by FAST are an extension of modern simulation packages, where users are presented with a palette of iconic components [Page and Oppen, 1999]. But FAST is a highly specialised extension of one of these packages, made possible by the high degree of flexibility in the WITNESS package. Because models can quickly be constructed and altered, building a model just prior to run-time to address specific issues, FAST *does* constitute a genuine move towards Composable Simulation. The status of FAST built models as Composable Simulations can be seen from the following diagram – modelling components required to create the model are not drawn together until run time, and this stage is automated according to the prior input of the Manufacturing Engineer:

Fig 2. Model Creation.



We saw in the previous section a number of reservations concerning the potential practical application of Composable Simulation. However, by restricting ourselves to a specific (engine assembly) domain, with a ‘natural’ and consistent level of modular detail, significant progress has been made through the control of combinatorial complexity. In this application we have therefore managed to

achieve many of the benefits associated with Composable Simulation, and learnt valuable lessons that will be required if we wish to extend our modelling scope or broaden the problem domain.

5. THE FUTURE

Another frontier area in simulation is the incorporation of Virtual Reality [Waller and Ladbrook, 2002]. This has the primary benefit of communicating simulation to the widest range of agents. However, VR currently requires highly intensive work, both in terms of computation and development effort. A composable approach allows the consolidation of work already carried out, again helping to control the new effort required in each new endeavour – enabling those involved to stay ahead of each new leap in computational speed.

Going back to Composable Simulation *per se*, Web-based simulation appears to mark the envisaged culmination of this work – where modelling constructs proliferate in the same way that information does today. [Page and Opper, 1999]. There are special challenges here though, as the user doesn't merely need to search for the right constructs, but must take account of the relationship of the emergent properties of these components with the modelling objectives they face – as we have had to do with FAST.

We will thus always need a modeller – but their role may become more abstract. Today's modellers need not have in depth programming knowledge (although many of them do!), and the number of layers between the underlying code and the finished model grows. This frees up resources and enables those engaged in the activity of simulation to concentrate on the *modelling* of process, rather than the *coding* required to do so.

There exists between the Lanner Group and Ford PTO a continuing commitment to develop improved tools and methodology. By operating at the frontiers of current simulation expertise, continuous improvements have been made in the design, implementation and overall management of new and existing Business Processes at Ford – improvements that at the time of writing are being rolled out globally.

6. REFERENCES

Balci O. (1994) “*Validation, Verification and Testing Techniques Throughout the Life Cycle of a Simulation*

Study.” in Balci O. (Ed.) (1994) *Annals of Operations Research* 53, pp 121 - 173.

Butler Group (2002) “*EAI and Web Services: Technology Evaluation and Comparison Report, Vol. 6.*” Butler Group Ltd, Hull.

Ezran, M. Morisio, M. and Tully, C. (2002) “*Practical Software Reuse*”. Springer-Verlag. Lond.

Hlupic, V. & Paul, R. J. (1999). “*Guidelines for Selection of Manufacturing Simulation Software.*” IIE Transactions Vol. 31 pp. 21-29.

Kasputis, S. & Ng, H.C. (2000) “*Composable Simulations.*” in Joines, J.A; Barton, R.R; Kang K. and Fishwick P.A. (Eds.), (2000) “*Proc's of the 2000 Winter Sim. Conference.*” Piscataway, NJ. IEEE. pp. 1577-1584.

Ladbrook, J. and Januszczak, A (2001) “*Fords PowerTrain Operations – Changing the Simulation Environment.*” UKSIM 2001 Conference of the United Kingdom Simulation Society pp.81-87

Page, E.H. & Opper, J.M. (1999) “*Observations on the Complexity of Composable Simulation.*” in P.A. Farrington, H.B. Nembhard, D.T. Sturrock, and G.W. Evans (Eds.) (1999) “*Proc's of the 1999 Winter Sim. Conference.*” Phoenix, AZ. IEEE. pp. 553-560.

Robinson, S. (1999) “*Simulation Verification, Validation and Confidence: A Tutorial.*” in Zeigler, B.P. (Ed.) Transactions of the Society for Computer Simulation International. Vol. 16. No. 2. pp. 63-69.

Waller, A. P. and Ladbrook, J. (2002) “*Experiencing Virtual Factories of the Future.*” in E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, (Eds.) (2002) “*Proc's of the 2002 Winter Sim. Conference.*” San Diego, CA. IEEE. pp 513-517.



ANDREW WINNELL received a B.Sc. (Dual Hons.) in Economics and Statistics from the University of Keele in 2001, and an M.Sc. in Management Science and Operational Research from the University of Warwick in 2003 after completing a dissertation on the selection of simulation software. He joined the Lanner Group as a Consultant in the latter half of 2002.

APPLYING NEW TECHNOLOGIES TO AUTOMATE AND SUPPORT COMPLEX SIMULATION MODELS FOR OIL DISTRIBUTION IN BRAZIL

BARBOSA, GUILHERME.J. & LIMOEIRO, CLÁUDIO.D.P.

E-mail: gjb@petrobras.com.br

*Petróleo Brasileiro S.A. (PETROBRAS)
Operations Research Team
Information Technology/Solution Provision*

*Av. República do Chile, 65 – Sala 1601
CEP: 20035-900 – Rio de Janeiro – Brasil
Phone: +55 21 2534-7469 Fax: +55 21 2534-3895*

Abstract: Distribution centers managed by BR, one of the Petrobras group companies, are the starting point from Gas stations and big fuel customers supply. These centers, intermediate links of the fuel supply chain, provide oil products to a very competitive market. To be efficient, the centers must have regulating stocks that use service level parameters for management purposes. Determining these parameters according to the number of centers and their localization around Brazil require plenty of simulation models, which have to be managed by an information system, so that they can be used in a friendly way. The focus of this paper is to describe the new technologies involved in supporting and automating the models created, the steps being taken for their implementation and the role of each software used.

keywords: rule-based systems, system integration with high performance.

1. INTRODUCTION

Petrobras is a large petroleum company, leader in Latin America, present in several countries and is considered one of the top energy companies in the world.

The Operations Research Team has a lot of experience in building decision support systems, mathematical programming, simulations and statistics. Our main applications are in oil selection and purchasing, investments, distribution, sales forecasting and resource allocation.

The Operations Research team has always developed projects using simulation. The team used FORTRAN, then moved to GPSS. Today, **PROMODEL** languages are used.

The most relevant simulation applications in Petrobras can be classified in two groups:

- Critical resources, drilling or maintenance rigs, the rental of specialized vessels (about US\$20,000 per day), port extensions, fleets and others.
- Determination of stock levels, LPG infrastructure (pipes and tanks) investments.

In the present case, a system was planned to manage all the process of simulation, starting from

the choice of the distribution center, the selection of a product handled at this center, organizing the creation of scenarios, initiating and monitoring the simulation process of the equivalent model chosen. Finally, it has also been projected to generate customized reports, some of them presenting graphical features.

The system has been developed in VISUAL BASIC, and it integrates PROMODEL, MS EXCEL and MS ACCESS to support all that functionality. Some difficulties appeared when starting to integrate all the software and making it work together. After that, other relevant aspects came round, like the time spent by the simulation models processing and the consumption of computer resources (memory), beside others.

Recently, the CITRIX solution has been tried out in order to solve some of these problems and then, after being approved, would be implemented.

2. THE PROBLEM (SIMULATION MODEL OBJECTIVE)

The project's objective was to define the amount of fuel that is stocked in the distribution centers, to set up service level goals and company management strategies, keeping costs at competitive levels.

The concept of safety stock inventory is widely used in petroleum companies, because mistaken sales forecasting and irregular supplies may cause loss of sales (stock-out) or increase stocking costs (surplus stock).

Sales variability and Irregular supplies (supply uncertainty) from distant centers might cause **stock-out** if there is a lack of product, possibly caused by a delay of the deliver, or **surplus stocks**, which involves higher costs, using ships or trains for storage if there is no room in the tanks at the centers.

Our first main difficulties were:

- Number of sites;
- Extensive supply lines all over Brazil;
- Almost all clients are demanding experts in logistics;
- Need to adapt results to each situation; in other words, we had to customize models for each center;
- Need for huge amounts of information;
- Need for a friendly IT solution, mainly because this solution is intended to be used intensively.

3. THE SOLUTION (SYSTEM STRUCTURE)

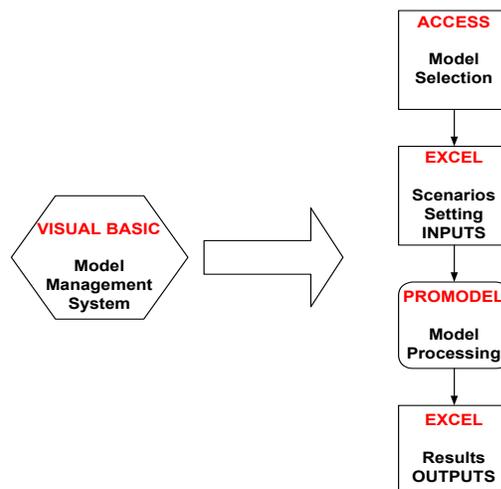
The solution was to develop a Decision Support System involving a large number of simulation models to inform the distribution sites stock management. It was developed in VISUAL BASIC and integrates PROMODEL, EXCEL and ACCESS to support all the following functionality:

- Determine, monitor and review target stocks, tank capacities and other parameters for each site;
- Support models for the calibration and maintenance of databases;
- Creating scenarios for studies;
- Manage all the process of simulation (initiating and monitoring the simulation process of the equivalent model chosen);
- Generate customized reports.

As we were working with about 50 models, each one containing different characteristics in terms of inputs and output format files, we need to develop a System to support and manage the process related to the simulation of the models, covering the distribution and the fuel security stocks from all the BR refineries and distribution centers (subsidiary), which are geographically spread all over Brazil.

The focus of this paper is to describe the appliance of new technology to manage and support the amount of models created and turning them easy to use (friendly)

Decision Support System Architecture:



Picture 1

The models has been developed in **PROMODEL**, which has got some built-in tank routines developed to make it easier to construct models with tank representations and its product operations. Besides that, it gives us much more flexibility to programme our fuel distributions process as close as possible to its real performance.

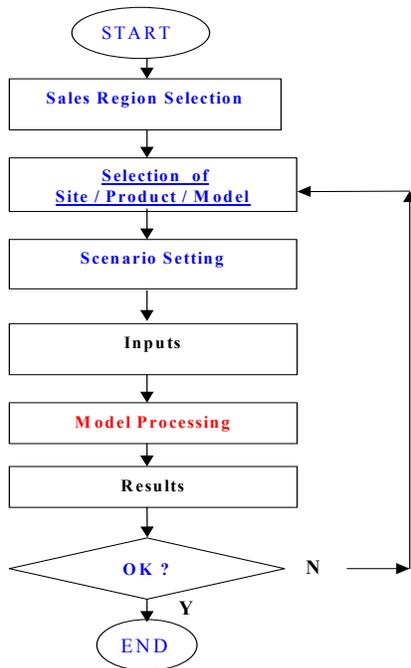
The model's structure is based on tridimensional arrays, importation and exportation files, intern tables and others. Some generalized subroutines were created, saving a lot of time in developing models for all the company's distribution centers, because they could be easily adapted for developing all the models, due to their similarities. The files used by the system throughout the process also use generalized codes which can be customized very fastly, thanks to their structure.

Because of the limited resources of the Simulation Softwares available, we've had to decentralize the model using **EXCEL** and **ACCESS** capabilities so that we could generate scenarios and turn it's usage much more easier for those who doesn't know how to operate simulation softwares.

Then, the next step was to develop the automation of the simulation software, and the **VISUAL BASIC** was chosen for that.

Structurally, the process can be presented as the following flowchart

Colors: ProModel - *red*; VisualBasic - *blue*;
 Excel - *black*; Access - (underlined).



Picture 2

The step of **Selection of Site/Product/Model** is supported by **ACCESS**. The database makes the relation among the Sales Region, the sites and the products available at that site, and also combine the selection with the appropriate equivalent model developed for that choice.

Due to the large amount of data, an **ACCESS** database was also needed to support the distribution defining process and their calibrations.

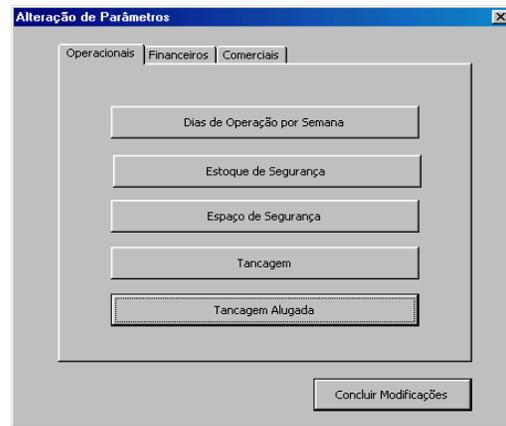
Like all random simulation models, these models are prepared to receive regular calibrations. However, any emergency calibration can be made when the need arises. This can be easily done because of the database.

The next picture shows the first screen displayed when the system is initialized

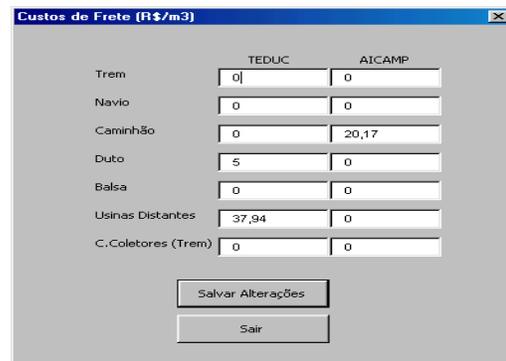


Picture 3

There is a large amount of **EXCEL** files manipulation, managed by the system, each one having a different **MACRO** developed to support the process and make it easier for the user to modify the **Inputs** when creating scenarios. Examples of input parameters screen (**Pictures 4 and 5**)

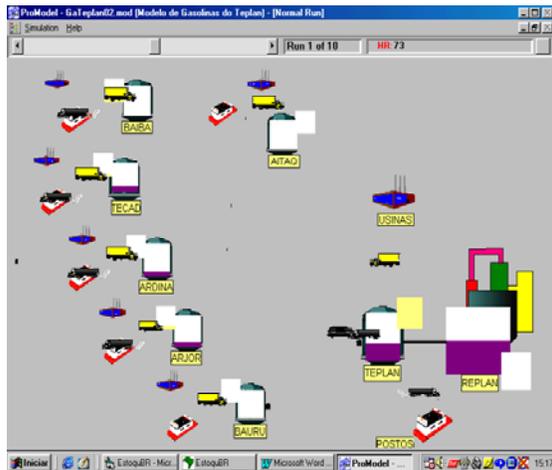


Picture 4



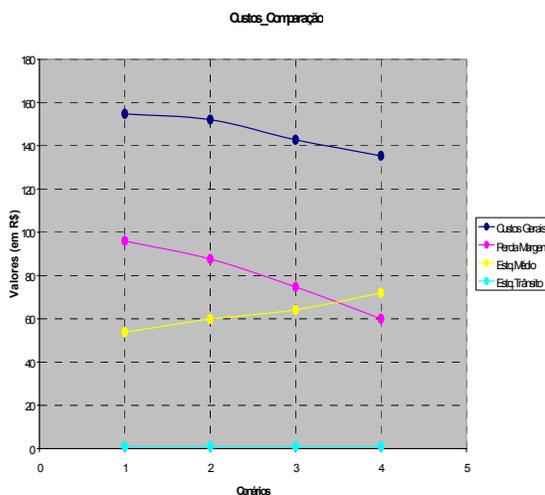
Picture 5

The following picture shows an example of a model being processed by **PROMODEL**



Picture 6

Some macros were also developed in **EXCEL** to present the **Results** and to calculate the average values obtained from the replications processing for each scenario. In addition, reports are generated at the end of each scenario simulation and there is also a customized report which compares graphically the results of the different scenarios created as it can be seen in the next picture.



Picture 7

There are two types of graphics generated by the system. One of them shows the evolution of the service level of the distribution centers when modifying security stock levels for each product handled in the model chosen. The other one is the evolution of the stock costs within the distribution system for each scenario created.

The reports generated at the end of each scenario simulation contains only customized information about it, which was developed in **EXCEL** in order to give the user much more flexibility to analyse and use the results.

4. FURTHER PROBLEMS (APPLYING NEW TECHNOLOGY)

After that, we have faced some problems related to the installation of the system, because it involves different softwares and also because of the different versions of the required softwares found on the client's machine. Besides that, the system consumes memory because it access information through a data bank and manipulate lots of information from different files and writes a lot of information into another files.

In addition, our client used to have short period decisions on it's daily workday and their resources (computer) could not be interrupted to do a long processing, because it could interfere their operational decisions.

In order to solve this difficult matter, we decided to analyse the **CITRIX** solution.

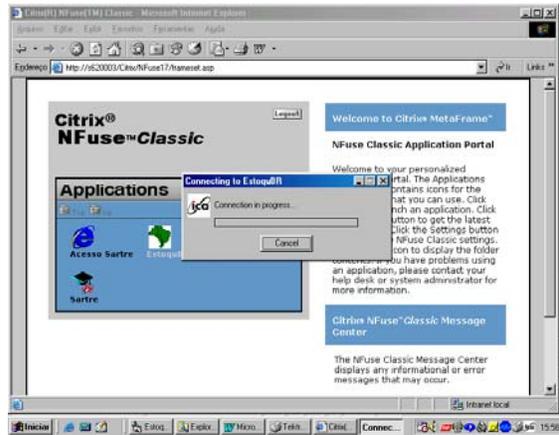
With this solution we could get some advantages:

- This operation costed almost nothing for our department in terms of equipment, maintenance and support, because we have already had a server available to install all the necessary files and programs and a team trained with this technology (**CITRIX**) to do the necessary assistance;
- This solution doesn't demand upgrade from the client's Hardware (computer);
- The system can be used in different Operational Systems, because the application runs in the server (Metaframe), releasing the clients resources (concept of *Thin Client*);
- Makes easier the updatings and management of the system (through the server);
- Reduces the costs of development, maintenance and more (usage of client's computer);
- Visualization through a common Web Browser (Nfuse Technology);

Drawbacks:

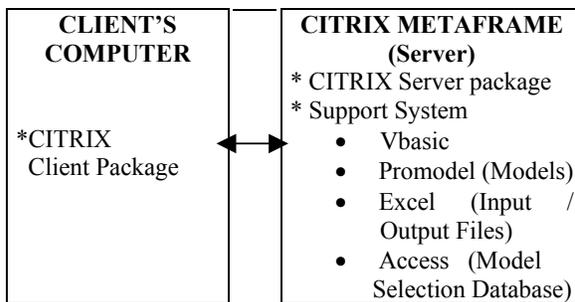
- Print the results through a local Printer (client);
- Availability of the Metaframe **CITRIX** (server) for a large usage and trustability (multiusers access).

Example of a connection through a common web browser



Picture 8

The structure of this functionality is shown below



Picture 9

5. CONCLUSIONS

The usage of new technologies is recommended to support and manage all the simulation process, but some important issues should be observed when many application are integrated to work all together.

Besides the simulation software, support software versions, PC hardware and Operational System should follow a common configuration set up by the developer. If it not possible to accomplish, because of the dependency of the applications used, the Citrix solution seems to be a good support decision.

We have also tried to use the system with more than one user at the same time, and everything seems to work well. The performance wasn't harmed by the multiple accesses.

Recently, we have been testing it exhaustively. As soon as it has been approved, we are going to

implement this solution for our users. It would save time in development, maintenance and, above all, costs.

6. REFERENCES

- ProModel Corporation, 2002, "ProModel User's Guide", Orem, UT.
- Law, A.M. & Kelton, D.W., 1991, "Simulation Modeling and Analysis", McGraw Hill
- Kelton, W.D., 1997, "Statistical Analysis of Simulation Output", Proceedings of the 1997 Winter Simulation Conference.
- Visual Basic 6.0 Help Online
- Excel 97 Help Online
- Site <http://www.citrix.com>

7. BIOGRAPHY



Guilherme Júlio Barbosa is 29 years-old, has graduated in Electrical Engineering at UFJF and has also a MSc Degree in Industrial Engineering at PUC-RJ. He has been working for 2 years as an Analyst of Operations Research for Petrobras, with simulation projects related to

the determination of stock levels from BR Distribution Centers (one of Petrobras Subsidiaries), determination of the economical number of critical resources, drilling or maintenance rigs and rental of specialized vessels to assist the FPSO's offloading operations in Campos Basin (Brazil), besides others. Nowadays he is taking his Executive MBA in Petroleum at COPPE / UFRJ.

Visit our company's site

PETROBRAS – <http://www.petrobras.com.br>

8. ACKNOWLEDGMENT

We would like to thank all the professionals involved in this project: Luciano Rosa Pereira, Isa de Barros Furriel, Nelson de Maria da Silva, Volnei de Barros Jorge and Carlos Alberto Ornellas Boquimpani. Our immediate manager, Roberto Iachan (Operations Research Team) and our manager Jésus Guimarães (Solutions Providing Area), for all the trust and opportunity. Finally and above all, our acknowledgment to PETROBRAS, a great and ambitious company, respected by all the brazilians and conducted by the passion of it's employees. We're really honoured to be a part of it.

ON THE SIMULATION OF QUEUES WITH PARETO SERVICE

PABLO JESÚS ARGIBAY-LOSADA
ANDRÉS SUÁREZ-GONZÁLEZ
CÁNDIDO LÓPEZ-GARCÍA
RAÚL FERNANDO RODRÍGUEZ-RUBIO
JOSÉ CARLOS LÓPEZ-ARDAO
DIEGO TEJEIRO-RUIZ

ETSE de Telecomunicación, Universidade de Vigo, 36200 Vigo, Spain

Abstract: In $M/G/n$ queues —with G a heavy-tailed distribution— the tail of G has low probability but a dramatic impact on the performance of the system. The analytical treatment of $M/G/n$ queues is difficult, so many times we must use simulation to study them. But the simulation of systems using heavy-tailed distributions presents difficulties. We need efficient simulation methods to study those systems, and we can use $M/G/1$ systems as workbenches since they have some analytical results to check the simulation results with. In this paper we try to gain some insight into the nature of those difficulties, and propose, develop and analyze a method to speed up simulations of $M/G/1$ systems when G is heavy-tailed.

keywords: heavy tails, queue systems, steady state.

1. INTRODUCTION

$M/G/n$ queues —where G is a heavy-tailed service time distribution— are used to model queue systems where a range of values of the service time, whose probability is very low, have a drastic impact on the overall performance of the system. The Pareto distribution is one of these heavy-tailed distributions and it has been proposed as the page size distribution in Web servers or as the file size distribution in FTP servers. The accurate analytical treatment of $M/G/n$ systems is very difficult and in many cases it cannot be applied. Simulation is a possible method to study them. But simulations with heavy-tailed random variables present some additional difficulties, and care must be taken when extracting conclusions from the results of these simulations. It is necessary to have accurate and efficient simulation methods. Efficient because we need to generate big quantities of data for our simulation study to be accurate enough. And their accuracy can be checked by means of comparisons with known results from simpler systems with analytical solution. One of these simpler queue systems that can be studied analytically is the $M/P/1$ queue. $M/P/1$ systems can be used then as a workbench for more efficient simulation methods, able to deal with the heavy-tail problematic. The slow convergence of the simulations to the steady state may be an important problem of the simulation of $M/P/1$ queues.

Recent studies have shown the problems involved in

simulating $M/P/1$ queues. The reason of these problems is the heavy-tailed condition of the Pareto: very high values of the demanded service time appear with very low —but not negligible— probabilities, in such a way that their effect in the waiting time distribution is drastic. The heavy-tailed condition decisively contributes to rise the mean queue waiting time. But problems relating to practical aspects of computer simulation like finite machine resolution and finite and low simulation time make the simulations underestimate the parameters of interest, typically the mean queue waiting time. Gross [Gross et al, 2002] studies the impact of finite resolution random number generation on the mean queue waiting time estimation. It is interesting to know how many of these problems can be avoided with better simulation techniques and computer resources, and how the power-tailed condition effectively limits our efforts to speed up the simulations.

In this paper we investigate this problem, try to get insight in the impact of the transient period in the mean value, and propose a method to try to start the simulation near the steady state. We compare the proposed method with the traditional start from empty system.

2. HEAVY-TAILED DISTRIBUTIONS

A random variable (RV) X , with cumulative distribution function (cdf) $F(x)$, is said to be heavy-tailed if its complementary distribution function, $1 - F(x)$, has

an hyperbolic decaying tail:

$$\exists \alpha > 0 \left| \lim_{k \rightarrow \infty} \frac{1 - F(x)}{x^{-\alpha}} = c \in (0, \infty) \right.$$

The Pareto cdf, clearly heavy-tailed, is given by $F(x) = 1 - (m/x)^\alpha \quad \forall x \geq m > 0$, where m is called the scale parameter, and α is called the shape parameter. In [Gross et al, 2002] a Pareto distribution with $m = 1$ is used in a M/P/1 queue to show the problems that appear when simulating such system when α is near 2. In this paper we also fix m to 1 to demonstrate the benefits of our method in the same scenario. The Pareto probability density function (pdf) is given by $f(x) = \alpha \cdot m^\alpha / x^{\alpha+1} \quad x > m > 0$. The Pareto k^{th} order moment exists if and only if $\alpha > k$. Its mean value exists if and only if $\alpha > 1$ and is given by $\bar{X} = \alpha \cdot m / (\alpha - 1)$. Its second order moment exists if and only if $\alpha > 2$ and is given by $\bar{X}^2 = \alpha \cdot m^2 / (\alpha - 2)$

3. PARETO TAIL PROBLEMS

Recent research has shown that the estimation using computer simulation of the mean queue waiting time of a M/P/1 queue, \bar{W} , converges very slowly to its theoretical value when α approximates 2 [Gross et al, 2002]: simulation run-lengths as long as some million observations do not give estimations of \bar{W} close to the exact theoretical value in these cases.

The **Pollaczek-Khinchin** formula states that the mean queue waiting time is directly proportional to the second order moment of the service time in a M/G/1 queue:

$$\bar{W} = \frac{\lambda \cdot \bar{S}^2}{2 \cdot (1 - \rho)}$$

where λ is the average arrival rate of customers, S the demanded service time random variable and ρ the utilization factor of the queue system — $\rho = \lambda \cdot \bar{S}$ —.

If we have a limited resolution random number generator we will not be able to generate the extremely large values of S that appear occasionally in the actual system, so the measured \bar{S}^2 will tend to be low, and this will probably make the estimation of \bar{W} low. Even if we have an infinite resolution random number generator, we can give a rough estimation of how many observations of customer queue waiting times we need before getting close to the real mean value. If we have a random number generator with finite resolution which is only able to produce numbers between 0 and K , we will loose in the simulation service times greater than K . But the appearance in our simulation of values beyond a certain number is not only a matter of resolution of the random number generator, but relates to the intrinsic probability of that value, or range of values.

If we have a range of values whose probability is p , the mean number of trials in order to get one value in that range is $\frac{1}{p}$. The weight of the tail of a Pareto beyond a certain limit K , i.e. the probability of getting one value in the range (K, ∞) , is given by $K^{-\alpha}$, so the probability of getting all the values smaller than K in r trials is $(1 - K^{-\alpha})^r$.

In the Pareto case, when α is near 2, the tail has a great influence in the value of its second order moment. For example, we select the utilization factor of the system $\rho = 0.5$. We choose a shape parameter $\alpha = 2.1$, so $\bar{S} = 1.909$ and $\bar{S}^2 = 21$. If we generate a sample of 1 million observations, the probability of getting all the values smaller than K , i.e., the probability of having a sample indistinguishable of that from a truncated Pareto with truncation parameter K is $P = (1 - K^{-\alpha})^{10^6} \simeq e^{-\frac{10^6}{K^\alpha}}$. Considering the service time RV, S , whose pdf is a Pareto, and a service time RV S_t , whose pdf is a truncated Pareto from the former, with truncation value K , we have $f_{S_t}(x) = \frac{f_S(x)}{1 - \Pr(x < K)} \quad x < K$ with $\bar{S}_t = \frac{K^\alpha - K}{K^\alpha - 1} \cdot \bar{S}$ and $\bar{S}_t^2 = \frac{K^\alpha - K^2}{K^\alpha - 1} \cdot \bar{S}^2$. If we now impose a probability of 99 percent about having obtained a truncated Pareto, the correspondent K is 6434. and the mean value of the associated truncated Pareto, \bar{S}_t , is $0.999934 \cdot \bar{S}$. So intuitively the probability of getting a mean value of 0.999934 times the theoretical value —this ratio will represent the accuracy in the estimation— is 99 percent. This may be considered negligible —we are correctly estimating the mean value of the Pareto RV. But the second order moment of the truncated Pareto is $\bar{S}_t^2 = 0.58 \cdot \bar{S}^2$, what means that with a probability of 99 percent we are underestimating the theoretical value of the second order moment, with the estimation being 0.58 times the theoretical value.

So we see that with a probability of 99 percent we will also underestimate \bar{W} in a factor of at least 0.58, nearly half the theoretical, and the cause is we are generating too few service times to be able to reach the steady-state.

This means that the *a priori* high value of the run size of the simulation is in practice a very low one when the service time distribution is heavy-tailed. To have more accurate results we need a much larger number of samples. For example, if we impose an accuracy of 99 percent in the second order moment estimation —equivalent to the accuracy in the mean queue waiting time—, we obtain $K = 10^{20}$, so with a probability of 99 percent we will need no less than 10^{40} samples. If we want an accuracy of 90 percent, with a confidence of 99 percent we will need no less than 10^{19} samples. Fig. 1 details this. In it we plot the tolerance —one minus the accuracy— versus the number of samples.

These examples show that although the M/P/1 process, with α near 2, is ergodic in theory, the run sizes of the simulations needed to check that ergodicity will

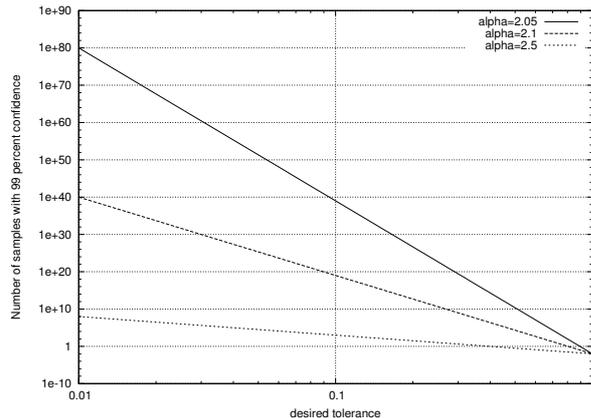


Fig. 1. Number of samples required for a given tolerance in the mean queue waiting time

probably be too high to consider the system ergodic in practice.

So we can see that estimating \overline{W} when the service time is heavy-tailed and the shape parameter α is slightly greater than 2—large variance—will probably be computationally very expensive if we start our simulations from an empty system. Thus, traditional simulation methods based on computing the samples of the involved RVs—the interarrival times of the customers and their service times—, will be too expensive due to the large amount of samples that must be generated before obtaining a representative set of samples of the involved processes.

4. CHOSEN INTERVAL LENGTH

We have developed a framework simulation model to achieve a greater accuracy in the simulations of M/P/1 queues with α slightly greater than 2. Its main idea is to try to initialize the simulation almost in steady-state. We can take advantage of our knowledge of the arrival process of the M/G/1 queue. When a user arrives at a M/G/1 system, it will possibly find some users in the queue and one in the resource. The queue waiting time of the arriving customer will be the residual life of the user in the resource, S_r ,—i.e., the remaining time that user will stay in the system— plus the service times of all the customers in the queue before our customer arrived. The distribution of the residual life of the customer in the resource will depend on the distribution of the service time, as it happens to the whole service time demanded by this user, L . Its pdf is given by [Kleinrock, 1975],

$$f_L(x) = \frac{x \cdot f_S(x)}{\overline{S}}$$

where $f_S(x)$ is the pdf of the demanded service time. From its definition, we can note that its mean, \overline{L} , will be $\overline{S^2}/\overline{S} = \overline{S} \cdot (1 + C^2)$.

So if our customer arrives to the system while there is somebody in the resource, it will arrive randomly in an interval described by $f_L(x)$, and, in average, it will have to wait $\overline{L}/2$ for the client in the resource to finish, plus some amount of time due to the users in the queue. If we denote M the number of clients in queue when the user in the resource entered it, and N the number of clients who arrived between the user in the resource began service and our user arrival, we can say that the queue waiting time for a user that has to wait is:

$$W = S_r + \left(\sum_{i=1}^M S_i + \sum_{j=1}^N S_j \right) \quad (1)$$

where the term between brackets represents the waiting time due to the customers in queue when our client arrived. So we can express the \overline{W} in the system as

$$\overline{W} = \rho \cdot \left(\frac{\overline{L}}{2} + \overline{M} \cdot \overline{S} + \lambda \cdot \frac{\overline{L}}{2} \cdot \overline{S} \right) \quad (2)$$

where we have used the fact that the waiting time will be non-null with probability ρ , and that the number of arrivals between the service start of the user in the resource and our client arrival is a RV with mean $\lambda \cdot \overline{L}/2$. To see what is the distribution of M , we can consider the queue of our M/P/1 system as another M/G/1 system. Since the departing customers from a M/G/1 system see the same distribution of the number of users in the system as the one seen by a random observer, the departures from the queue—to enter the resource— see samples of the Q RV.

Finally, we can write Equation (2) as

$$\overline{W} = \rho \cdot \left(\frac{\overline{L}}{2} + \overline{Q} \cdot \overline{S} + \lambda \cdot \frac{\overline{L}}{2} \cdot \overline{S} \right)$$

To achieve a good estimation of \overline{W} , we can simulate then a system where a user finds another customer being served, and whose service time follows the distribution $f_L(x)$, obtainable from $f_S(x)$. The number of users who arrive between the time the user in the resource began being served and our client arrival will be a Poisson RV whose mean will be known. The queue length when the selected interval began is one sample of the queue length distribution. If we could calculate a good estimation of the queue length Q , the sample of W when $W > 0$ would be obtained from Eq. (1).

We can approximate the theoretical convergence ratio of the classical simulation method (that starting from an empty system and simulating the system along the continuous time axis) and our proposed method using the considerations on probabilities of appearance of high-value samples we used in Section 3:

4.1. Classical Method

Consider the service time RV, S , whose pdf is a Pareto, and a service time RV S_t , whose pdf is a truncated Pareto from the former, with truncation value K

$$f_{S_t}(x) = \frac{f_S(x)}{1 - \Pr(x < K)} \quad x \leq K$$

Its first and second moments are

$$\overline{S_t} = \frac{K^\alpha - K}{K^\alpha - 1} \cdot \overline{S} \quad (3)$$

$$\overline{S_t^2} = \frac{K^\alpha - K^2}{K^\alpha - 1} \cdot \overline{S^2} \quad (4)$$

and we see that $\overline{S_t} < \overline{S}$ and $\overline{S_t^2} < \overline{S^2}$.

The probability of getting one sample value of a Pareto less than K is $1 - K^{-\alpha}$. If we generate a sample of size N_t of a Pareto distribution, the probability that this sample is indistinguishable of one of a truncated Pareto with truncation value K , i.e., the probability of all those samples are less than K is $(1 - K^{-\alpha})^{N_t}$, so with this probability we are getting a sample indistinguishable of one from a truncated Pareto whose truncation value will be K or less, so in this case an upper bound for the second moment is given by Eq.(4).

So if we want to calculate with a given confidence P an upper bound for the second moment with N_t samples of our untruncated Pareto process, we do the following:

$$P = \left(1 - \frac{1}{K^\alpha}\right)^{N_t} \simeq e^{-\frac{N_t}{K^\alpha}} \Rightarrow K \simeq \left(\frac{N_t}{-\ln P}\right)^{\frac{1}{\alpha}}$$

So the estimated second moment with a confidence of P over N_t samples, $\widehat{S^2}[N_t, P]$, is

$$\widehat{S^2}[N_t, P] = \frac{K^\alpha - K^2}{K^\alpha - 1} \cdot \overline{S^2} \simeq \frac{\frac{N_t}{-\ln P} - \left(\frac{N_t}{-\ln P}\right)^{\frac{2}{\alpha}}}{\frac{N_t}{-\ln P} - 1} \cdot \overline{S^2}$$

If we denote $\widehat{W}[N_t, P]$ the estimated \overline{W} with probability P over N_t samples, and define the accuracy in the estimation of \overline{W} as

$$A_1[N_t, P] = \frac{\widehat{W}[N_t, P]}{\overline{W}}$$

it results that the estimated accuracy in \overline{W} with a confidence of P over N_t samples, $A_1[N_t, P]$ is

$$A_1[N_t, P] = \frac{\widehat{W}[N_t, P]}{\overline{W}} = \frac{\widehat{S^2}[N_t, P]}{\overline{S^2}} = \frac{\frac{N_t}{-\ln P} - \left(\frac{N_t}{-\ln P}\right)^{\frac{2}{\alpha}}}{\frac{N_t}{-\ln P} - 1} \quad (5)$$

4.2. Proposed Method

We use the relationship

$$\overline{W} = \left(\frac{\overline{L}}{2} + \lambda \frac{\overline{L}}{2} \overline{S} + \overline{Q} \cdot \overline{S}\right) \cdot \rho \quad (6)$$

where L is the distribution of the chosen interval length. If S is a Pareto with shape parameter α , is easy to see that L will be a Pareto with shape parameter $\alpha_2 = \alpha - 1$. To calculate the estimation of \overline{L} as function of the number of samples, n , we use the same method as above.

Considering the Pareto RV L , the estimation of the mean of the correspondent truncated Pareto, L_t with shape parameter $\alpha_2 = \alpha - 1$ and with truncation value K is given by Eq (3)

$$\overline{L_t} = \frac{K^{\alpha_2} - K}{K^{\alpha_2} - 1} \cdot \overline{L} = \frac{K^{\alpha-1} - K}{K^{\alpha-1} - 1} \cdot \overline{L}$$

In Equation (6) there is a term that represents the average value of Q . We will estimate \overline{Q} producing an initial number of busy periods, randomly choosing one point in time and calculating the Q when the selected customer in service entered the resource. So if we generate n samples of this initial simulation, we think we can reasonably suppose that our estimation of \overline{Q} will tend to \overline{Q} with the same speed like the one we estimate \overline{L} with. That supposition has been backed with simulation results shown in Fig. 3, where we plot the empirical pdf of the estimated \overline{W} with 100 simulation runs of the classical method and the proposed method, and it can be seen that the obtained mean values are close to those predicted by the analytic expression in Eq. (7). Using this supposition, and if we denote A_2 the accuracy in the estimation of \overline{L} , \widehat{L}/\overline{L} , we have

$$\begin{aligned} \overline{W} &= \left(\frac{\overline{L}}{2} + \lambda \cdot \overline{S} \cdot \frac{\overline{L}}{2} + \overline{Q} \cdot \overline{S}\right) \cdot \rho \\ &= \left(\frac{\widehat{L}}{2 \cdot A_2} + \lambda \cdot \overline{S} \cdot \frac{\widehat{L}}{2 \cdot A_2} + \frac{\widehat{Q} \cdot \overline{S}}{A_2}\right) \cdot \rho = \frac{\widehat{W}}{A_2} \end{aligned}$$

$$A_2 = \frac{\widehat{W}}{\overline{W}} = \frac{\widehat{L}}{\overline{L}} = \frac{K^{\alpha-1} - K}{K^{\alpha-1} - 1}$$

The confidence for N_t samples being from a truncated Pareto with shape parameter $\alpha - 1$ and truncation point K or less is

$$P = \left(1 - \frac{1}{K^{\alpha-1}}\right)^{N_t} \simeq e^{-\frac{N_t}{K^{\alpha-1}}} \Rightarrow K = \left(\frac{N_t}{-\ln P}\right)^{\frac{1}{\alpha-1}}$$

So an upper bound for the accuracy of the estimated \overline{W} will be, for a confidence P and N_t samples,

$$A_2[N_t, P] = \frac{\frac{N_t}{-\ln P} - \left(\frac{N_t}{-\ln P}\right)^{\frac{1}{\alpha-1}}}{\frac{N_t}{-\ln P} - 1} \quad (7)$$

Fig. 2 compares the theoretical results for the upper bounds of the convergence rates of both methods, the classical one, given by Eq. (5), and the proposed one, given by Eq. (7), for a probability of 99 percent. We see that our method does not underestimate the real mean value of the queue waiting time as much as the traditional method. There is still a big difference between both estimations and the real value due to the fact mentioned in section 2: the probabilities involved for high values of the service times are too small for those values to appear in short simulations; but the improvement in the estimation is appreciable. This method can serve as basis for more improvements using known facts from the underlying processes, and we are working in the improvement of the simulation algorithm.

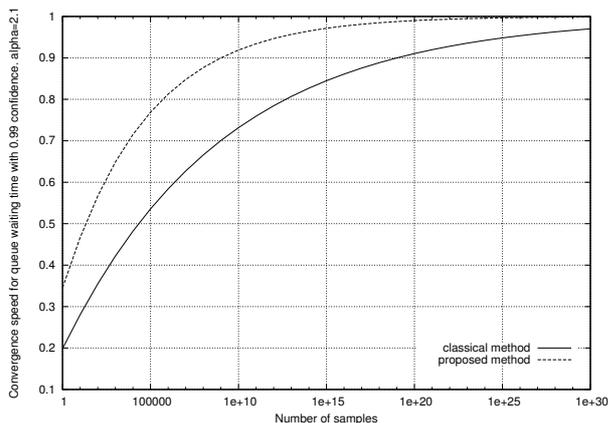


Fig. 2. Comparison between the 99 percent confidence upper bounds for the convergence rates of the classical and proposed method.

5. IMPLEMENTATION

To obtain samples of W with our method, we generate the value 0 with probability $1 - \rho$, and with probability ρ a sample of the service time length found by a typical customer that has to wait. This is a Pareto with shape parameter $\alpha_2 = \alpha - 1$, where α is the shape parameter of the Pareto representing the service time. Next, we choose a random point in the generated interval which will represent the arrival instant of a typical client. The queue length in this moment will be the clients in queue when the selected interval began, plus the number of clients who arrived between the beginning of the interval and the arrival of our client. This last number is a Poisson RV with mean $\lambda \cdot U$, with $U = L - S_r$, the elapsed time since the beginning of the interval and our client arrival. We can directly generate samples of this RV. But the number of users in queue

when the interval began follows the distribution of Q , which is unknown, so we will have to estimate it using a classical simulation. The waiting time of our client will be, then, the residual life of the interval plus the service times of the users in queue when it arrives.

6. PERFORMANCE

To evaluate the performance of simulations using this method, we note that it uses more samples of the random variables involved than the classical method. The classical method needs one interarrival time and one service time to produce one waiting time sample. Our method needs to generate one classical simulation to obtain estimates of Q , the queue length. To obtain one estimate of the waiting time, we need to estimate one queue length sample, Q_i , with a classical simulation; we need to generate one sample of a Pareto with shape parameter $\alpha - 1$; one Poisson to estimate how many customers arrive between the beginning of the chosen interval and our arrival, N , and $N + Q_i$ service times. Moreover, taking into account the fact that our estimates of Q will not be independent, because we are obtaining them from samples in one finite simulation, to reduce that dependence we can think of choosing a small proportion of estimates of Q from the total number of samples of our classical simulation. This makes the mean number of random values to generate one sample of the waiting time in our method bigger than that of the classical method. But that difference is not important enough to make the proposed method worst in performance than the classical.

If we have a $M/P/1$ with shape parameter α , the classical simulation will need 1 Poisson RV and 1 Pareto RV to obtain one sample of the waiting time. In our method, we need one sample of the queue length from a classical simulation. To reduce dependence between samples of it, we choose them sampling the classical simulation with a Poisson process with mean λ/n , with $n > 1$. We will need n samples of Q in the classical simulation to select one of them, Q_i , for computation, and that means n Poisson RVs and n Pareto RVs. We generate one more Pareto for the length of the selected interval, one Poisson for the number of arrivals in that interval prior to ours, N , and $Q_i + N$ Pareto RVs for the service times of all the arrivals. If we have a Pareto service time with $\alpha = 2.1$, and $\rho = 0.5$, using the **Pollaczek-Khinchin** formula we have $Q = 1.44$, and the average length of the chosen interval is 11. If we select in average one of every four samples of Q for computation, in the worst case, that in which we do not underestimate the theoretical Q —because in that case there are more service times to generate—we will need in average 4 Pareto + 4 Poisson + 1 Pareto + 1 Poisson + 1.44 Pareto + 1.44 Pareto = 7.88 Pareto RVs + 5 Poisson RVs. This implies that we need approximately 6.5 times more samples to obtain one sample of W than in the classical method. One fact that favours the

efficiency of our method is that it always produces samples of W with $W > 0$. The classical method generates interarrival and service times to produce the value $W = 0$ with probability $1 - \rho$. This is, we are wasting computer resources to generate one known value whose probability is known *a priori*. Our method only produces samples of W when $W > 0$, giving a mean value $W_{W>0}$. The final mean value of W will be $\rho \cdot W_{W>0}$. The lower is ρ , the better is our method in terms of efficiency compared with the classical method. So in the previous example, given that $\rho = 0.5$, we can consider our method to use $6.5/2 = 3.25$ times more samples than the classical one.

If we generate some simulations of the two methods, and represent the pdf of the estimated $\bar{W} = 5.5$, we obtain Figure 3, for which we run 100 simulations of 1 million samples of \bar{W} in every method. It is clear that the proposed method has better accuracy. If we take into account that our method uses more samples and represent the pdf of the two methods but this time the classical method uses four times more samples to compensate the more samples used by our method, we obtain Figure 4, which uses 100 runs of 1 million values of the waiting time with the proposed method and 100 runs of 4 million values in the classical method. The difference between the accuracies in both methods is lower than that in Figure 3, but it is still appreciable that the proposed method works better, now with similar performance.

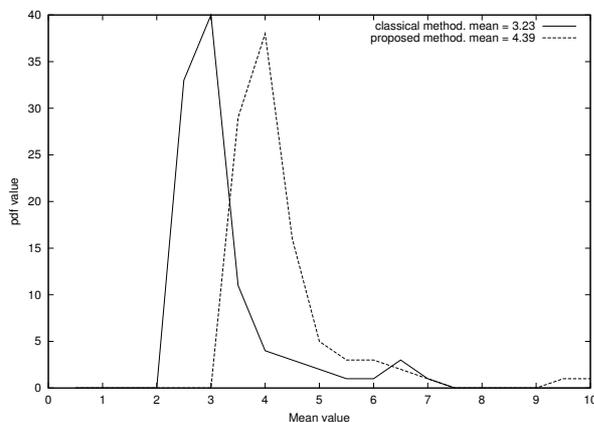


Fig. 3. Empirical pdf of \bar{W} of the classical and the proposed methods with 100 runs of 1 million waiting times each.

7. CONCLUSION

The computer simulation of M/P/1 queues presents important difficulties due to the slow decaying tail of the Pareto distribution. This makes extremely high values, with great influence on the statistical figures of the system, appear with so low probabilities that if we want to simulate the physical underlying processes, generating demanded times and time arrivals,

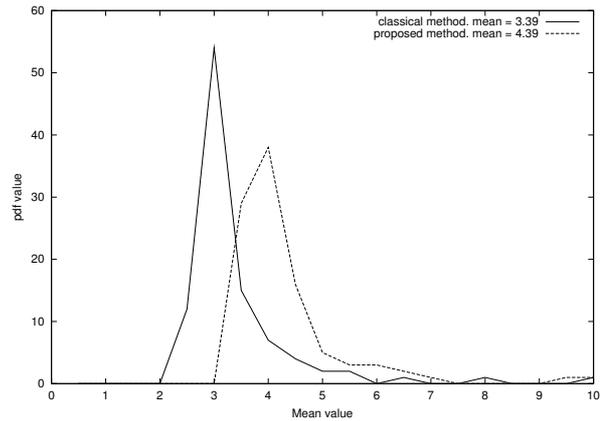


Fig. 4. Empirical pdf of \bar{W} with 100 runs of 10^6 waiting times in the proposed method and $4 \cdot 10^6$ in the classical one.

the cost in time will probably be prohibitive if we want accurate results. This forces to use all our knowledge of the statistics of the system inner processes, so the simulation can noticeably speed up.

REFERENCES

- Gross, D., Shortle, J.F., Fischer, M.J. and Masi, D.M.B. 2002. "Difficulties in simulating queues with pareto service". In *Proceedings of the 2002 Winter Simulation Conference*, 2002.
- Kleinrock, L. 1975. "Queueing systems". Wiley & Sons.
- Takács, L. 1962. "Single server queue with poisson input simulation." In *Operations Research* 10:388–397.
- Sigman, K. 1999. "A primer on heavy-tailed distributions." In *Queueing Systems*. 33: 261-275.

AUTHOR BIOGRAPHIES



PABLO JESÚS ARGIBAY-LOSADA is an assistant professor in the *Departamento de Enxeñaría Telemática* at *Universidade de Vigo*. He received a telecommunication engineering degree from *Universidade de Vigo* in 2001. Nowadays, he is working toward his Ph.D. in the *Departamento de Enxeñaría Telemática* at *Universidade de Vigo*. His e-mail address is <Pablo.Argibay@det.uvigo.es>.



ANDRÉS SUÁREZ-GONZÁLEZ is an associate professor in the *Departamento de Enxeñaría Telemática* at *Universidade de Vigo*. He received a Ph.D. degree in telecommunication engineering from *Universidade de Vigo* in 2000. He is a member of ACM. His current research interests include simulation methodology and analysis of stochastic systems. His e-mail address is <asuarez@det.uvigo.es>.

MODELLING HUMAN DECISION-MAKING

STEWART ROBINSON

*Operational Research and Systems Group
Warwick Business School
University of Warwick
Coventry
CV4 7AL*
stewart.robinson@warwick.ac.uk

Abstract

A series of projects have been, and are being, performed, that look at modelling human decision-making in simulations. The focus is on using a simulation to elicit knowledge about human decision-making. Artificial intelligence methods are then used to learn the humans' decision-making strategies. By linking the trained artificial intelligence system with the simulation, it is possible to assess the performance of the decision-maker. Results are presented from the most recent project. The motivation for modelling human decision-making is also discussed. In this work the prime motivation is to understand and improve decision-making, rather than to develop more accurate simulation models.

1. Introduction

Since the mid-1990s the author has been investigating the use of artificial intelligence methods as a means for representing human decision-making in simulations. This paper describes the history of this work and future work that is being undertaken. Starting from an idea generated when attempting to model rail marshalling yards, an artificial example of simulation and expert systems working in collaboration was generated. The ideas were then applied to a real case of maintenance operations at an engine assembly plant. Future work is looking into simulation as a means of knowledge elicitation. The paper briefly describes each of these phases of work and concludes by discussing why it is important to model human decision-making.

2. Forming Ideas

In the mid-1990s the author undertook an ESPRIT funded project looking into the simulation of industrial rail marshalling yards. This work, carried out in collaboration with a Belgian consultancy, aimed to identify the requirements for a rail yard simulator. Previous experience had shown inadequacies in the commercial software available. It was particularly difficult to represent the movement and shunting of individual wagons, backwards and forwards in a yard.

In discussing the nature of rail yard operations another important issue arose. A supervisor is employed to receive incoming trains and direct the splitting up of wagons within the yard. The supervisor is also tasked with selecting wagons from different locations in order to form outgoing trains. This involves complex decision-making, especially if wagons are to be directed and removed from locations in order to minimise the movement and disturbance to the yard. Since the supervisor's knowledge is largely tacit, it is difficult for him/her to express the strategies that are employed. As a result, there is no direct means for representing the decision-making strategy in a simulation model. Indeed, it became apparent that

modelling the human decision-making was more problematic than modelling the physical movement of wagons.

3. Proof of Concept

A possible solution to this issue was the use of expert system, or potentially other artificial intelligence methods. Researchers had previously attempted this with some success (Flitman and Hurrion, 1987; O'Keefe, 1989; Williams, 1996; Lyu and Gunasekaran, 1997). Some of this work had been carried out a number of years earlier and none of it seemed to entail the use of commercial software, which was the focus of the rail yard study.

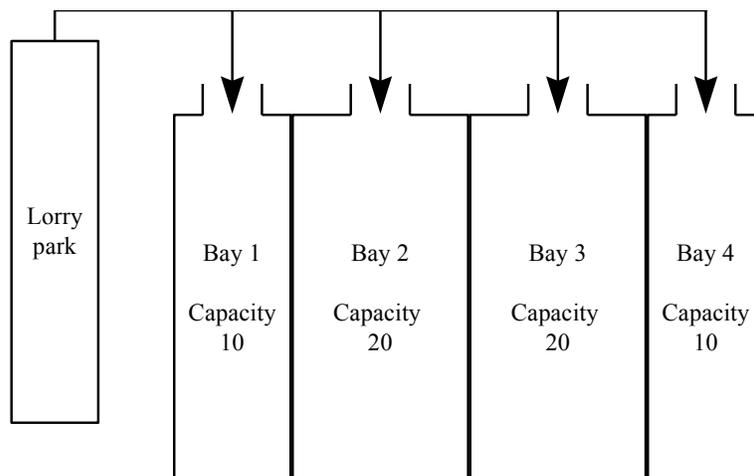
A small study was set-up to answer two questions:

- Could commercial expert systems be linked to commercial simulation software as a means of modelling human decision-making?
- Could the simulation model be used as a means of knowledge elicitation, by getting a decision-maker to interact with the model?

The aim of the second question was to see if the problem of tacit knowledge could be overcome by creating decision scenarios in a simulation and getting an expert to respond to those scenarios.

A simple simulation, based on a real case in a steel factory, was developed in Witness (figure 1). Lorries arrive at a lorry park requiring loads of between 5 and 20 items. On arrival the lorries are allocated to a loading bay by the bay supervisor, should a suitable one be available. In making this decision the supervisor must take account of the restrictions on the bay capacities. Lorries requiring more than 10 items must be allocated to bay 2 or 3, since bays 1 and 4 only have capacity for up to 10 items. Should a bay not be available then the lorry waits in the park until a suitable bay becomes available. Once a lorry is allocated, it moves to the bay where it is loaded before departing from the system.

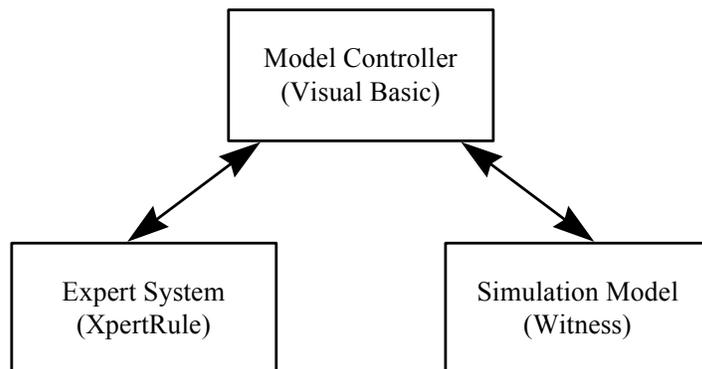
Figure 1 *Lorry Loading Bay Example*



XpertRule was used to develop the expert system that represents the supervisor's allocation decisions. This package was selected for two reasons. First, it adopts a rule induction approach. Second, XpertRule is one of the few expert systems packages available that has a true Windows implementation and is OLE compliant.

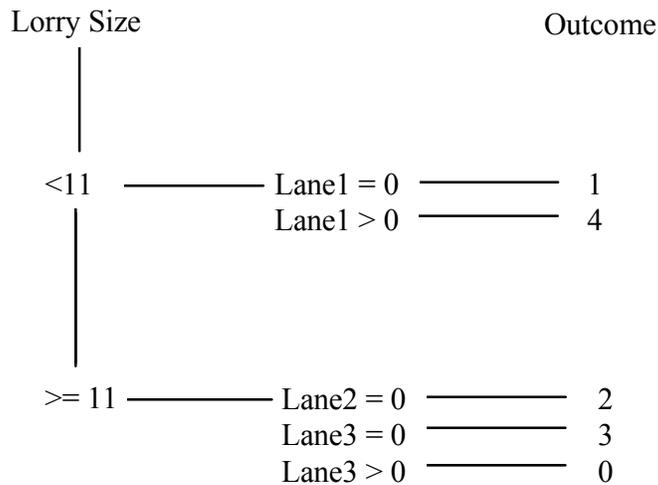
Since Witness can only work as an OLE slave, it was necessary to develop a model controller (MC) in Visual Basic (figure 2). The MC initiates the run of the simulation model. At a point where an allocation decision is required, the simulation model automatically stops and waits until the MC returns a decision and continues the run. Once the MC has detected that the model is not running, it extracts data from the model which it passes to the expert system for a decision. The decision is returned to the simulation model via the MC. Some effort was required to ensure that this sequence of events was adhered to. A particular difficulty was encountered in detecting whether the Witness model had stopped running before seeking a decision from XpertRule. If Witness could act as an OLE client it could call XpertRule directly, removing the need for the MC. This would have simplified the linking of the packages significantly.

Figure 2 *Linking Witness to XpertRule*



In order to develop the expert system decision tree, the simulation was first used as a knowledge elicitation engine. The simulation was run and at a decision point the user (the author) was prompted for an allocation decision. These decisions were logged in a data file along with variables describing the state of the system. These were then used to train the expert system. The decision tree shown in figure 3 was developed using this approach.

Figure 3 *Decision Tree Induced from Examples*



Once the decision tree had been defined, the simulation could be run with the expert system (rather than the decision-maker), in order to determine the effect of the decision-making strategy on the operation of the loading bay. Full details of this work can be found in (Robinson et al, 1998)

4. Modelling Maintenance Decisions in a Manufacturing Plant

The example above showed that commercial software could be linked for the purposes of representing human decision-making and that simulation could be used to good effect as a knowledge elicitation approach. As with previous work, however, this was an artificial example. The question, therefore, arose: could this approach be used in a real and complex case? In 1999 a three-year EPSRC funded collaborative project began, looking at this very issue. The project was a collaboration between Warwick Business School, Aston University, Ford Motor Company and the Lanner Group. The case considered was an engine assembly plant and the decisions taken by supervisors when a machine fails. The work is described briefly below and more fully in (Robinson et al., 2001)

In the engine assembly plant, blocks are placed on a 'platten' and pass through a series of automated and manual processes. For the purposes of this research, the maintenance operations on a self-contained section of the engine assembly line were considered. Prior to the research a simulation model of the complete engine assembly facility had already been developed. The model, developed in the WITNESS simulation software, was used to identify bottlenecks and to determine viable operating alternatives. The maintenance logic in the model assumed that when a machine fault occurred, the decision would be to make an immediate repair. Random sampling was used to determine the skill level of the engineer required to service the fault. These assumptions were considered to be adequate for the purposes of the study that was performed.

In practice, however, a maintenance supervisor has a number of options beyond repairing the machine immediately:

- Stand-by: an engineer manually processes parts until the end of the shift, when the machine is repaired.
- Stop the line
- Do nothing

The question was, could the simulation that already existed be used to elicit knowledge from the maintenance supervisors on how they made these decisions, and could this information be used to develop an artificial intelligence representation of the decision-maker? The aim was not so much to be able to develop a better simulation model, but to devise a means for identifying and then improving decision-making.

To address this issue, the knowledge based improvement (KBI) methodology was devised which consisted of five stages:

- *Stage 1*: Understanding the decision-making process
- *Stage 2*: Data collection
- *Stage 3*: Determining the experts' decision-making strategies
- *Stage 4*: Determining the consequences of the decision-making strategies
- *Stage 5*: Seeking improvements

These stages are described in detail in (Robinson et al., 2001).

Following a process of knowledge elicitation, up to 63 example decisions were collected from each of the three maintenance supervisors (one for each shift). Knowledge elicitation sessions lasted about one hour. This seemed to be a limit on the time the supervisors had available and on their ability to concentrate on making decisions in the model.

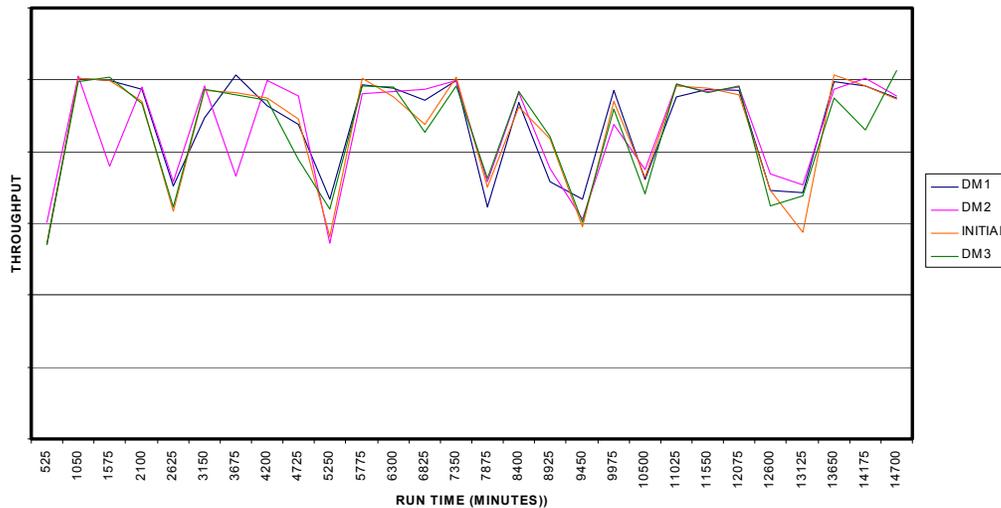
A number of artificial intelligence methods were trained using the examples obtained with varying degrees of success. Table 1 shows the proportion of example decisions that were incorrectly classified by the different methods. A zero score implies a perfect classification. The poor performance of the neural network is unsurprising, since they are known not to perform well with small training sets.

Table 1 *Misclassification Comparison*

	Decision-maker		
	1	2	3
ID3 (XpertRule)	0/63	0/63	0/53
CART (SPSS)	5/63	4/63	4/53
Neural Network (Matlab)	19/63	10/63	24/53
Logistic Regression (SPSS)	0/63	0/63	0/53

The simulation was then run with the ID3 decision tree. The results in figure 4 show the day-to-day throughput resulting from employing the three different decision-making strategies, as well as the results obtained from the decision logic in the original model developed by Ford. This shows some differences in the plant throughput as a result of the different decision-making strategies.

Figure 4 *Throughput under Alternative Decision-Making Strategies*



5. Knowledge Elicitation through Simulation

The work on the engine assembly case demonstrated the possibility of using the KBI methodology in a real situation, as well as some of the difficulties in its use. One particular difficulty was in obtaining realistic decisions from the supervisors and in obtaining sufficient example decisions to enable valid artificial intelligence representations to be trained. A further three year project started in October 2002 which will address these specific issue of knowledge elicitation. This work is also funded by the EPSRC with collaboration from Ford, Lanner Group and Aston University.

The specific objectives of the project are:

- To determine alternative mechanisms for eliciting knowledge from decision-makers using a visual interactive simulation
- To compare the alternative methods in terms of their efficiency (speed of data collection)
- To compare the alternative methods in terms of their effectiveness (accuracy of data collection)
- To compare the data collection methods in terms of the ability to train various artificial intelligence methods from the data sets collected

This will involve considering the following issues:

- *Level of visual display*: paper based, none, 2D, 2½D, 3D
- *Interactive interface*: number of decision-making attributes (key data upon which decisions are taken) that are reported to the decision-maker
- *Scenario generation*: use of historic scenarios, adapted historic scenarios to give more extreme examples, random sampling of scenarios, adapted random sampling of scenarios to give more extreme examples
- *Self learning*: learning responses to specific scenarios as the data collection progresses and automatically responding to future iterations of the same scenario

6. Conclusion: Why Model Human-Decision Making?

In conclusion, it is worth discussing the motivation for modelling human-decision making. Is it to enable the development of better models, or to help better understand and possibly improve human decision-making?

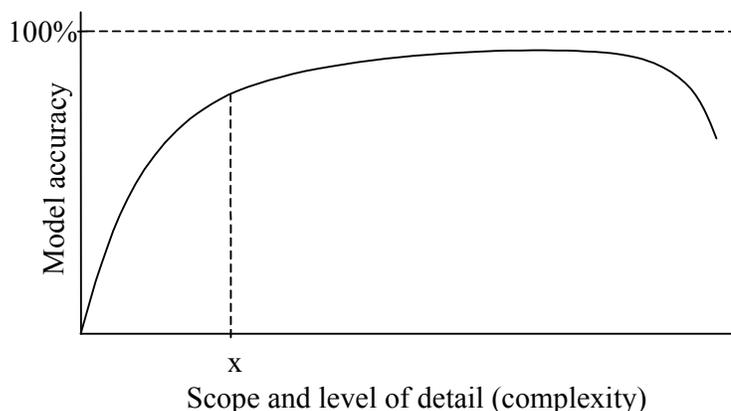
Checkland (1981) describes four types of system. Two are of interest here:

- *Designed physical systems*: systems designed by humans with no human interaction in day-to-day operations e.g. an automated warehouse.
- *Human activity systems*: systems of human activity, existing purely as human interaction e.g. political and social systems.

These represent systems at two extremes in terms of human interaction. In simulation modelling of the type considered in this paper (operations system modelling), we are rarely dealing with systems at either extreme, but somewhere between the two. Operations systems typically consist of a designed physical system in which humans interact e.g. a manufacturing line or a bank. Human decision-making is a key aspect of that human interaction. Since human interaction and decision-making are central to operations systems, there is a clear motivation for modelling that interaction.

But are we looking to develop better models by including elements of human decision-making? There is a potential problem with this motivation. Robinson (1994) presents the diagram shown in figure 5. This shows that there are diminishing returns, in terms of accuracy, from increasing the level of complexity in a model. Indeed, it is argued that there comes a point at which added complexity reduces the accuracy of a model because there is insufficient knowledge to support the detail being modelled. Modellers would argue that the optimum point, or best model, is around point x. This is the point at which the model is sufficiently accurate and beyond which there is little gain from additional complexity. The exact location of point x depends upon the purpose of the model, which in turn determines the required level of accuracy.

Figure 5 *Simulation Model Complexity and Accuracy (Robinson, 1994)*



One motivation for modelling human decision-making is to add extra complexity to a model in order to improve its accuracy. The danger of this approach is that it could be trying to climb along the flat part of the curve in figure 5 and so gives little gain. Indeed, it could be argued that although a slightly more accurate model is generated by modelling human decision-making this does not represent a better model, since a large amount of effort is required to obtain only an incremental improvement in accuracy. This argument depends very much on the modelling context and the required level of accuracy. There are cases where incremental gains in accuracy are needed, since a high level of fidelity is required.

Another motivation is to model human decision-making so it is better understood and it can be improved. This should help to improve the performance of the systems in which the humans are interacting. The concentration is no longer on making models more accurate, but on using the models to assess the effects of human interaction and to look for ways of changing the human interaction in order to improve system performance. In this case model accuracy plays a secondary role to generating insight and understanding. This is the motivation behind the knowledge based improvement methodology.

Acknowledgements

The author wishes to acknowledge the support and collaboration of the EPSRC, Ford Motor Company (John Ladbroke), Lanner Group (Tony Waller) and Aston University (Professor John S. Edwards).

References

- Checkland, P.B. (1981). *Systems Thinking, Systems Practice*. Wiley, Chichester, UK.
- Flitman A.M. and Hurrion, R.D. (1987). Linking Discrete-Event Simulation Models with Expert Systems. *J. Opl Res. Soc.*, **38** (8), pp. 723-734.
- Lyu, J. and Gunasekaran A. (1997). An Intelligent Simulation Model to Evaluate Scheduling Strategies in a Steel Company. *International Journal of Systems Science*, **28** (6), pp. 611-616.
- O’Keefe, R.M. (1989). The Role of Artificial Intelligence in Discrete-Event Simulation. *Artificial Intelligence, Simulation and Modeling* (L. E. Widman, K.A. Loparo and N.R. Neilsen, eds.), pp. 359-379. Wiley, NY.
- Robinson, S. (1994). Simulation Projects: Building the Right Conceptual Model. *Industrial Engineering*, **26** (9), pp. 34-36.
- Robinson, S., Edwards, J.S. and Yongfa, W. (1998). An Expert Systems Approach to Simulating the Human Decision Maker. *Winter Simulation Conference 1998* (D.J. Medeiros, E.F. Watson, M. Manivannan, J. Carson, eds.), The Society for Computer Simulation, San Diego, CA, pp. 1541-1545.
- Robinson, S., Alifantis, A., Edwards, J.S., Hurrion, R.D., Ladbroke, J. and Waller, T. (2001). Modelling and Improving Human Decision Making with Simulation. *Proceeding of the 2001 Winter Simulation Conference* ed. B.A. Peters, J.S. Smith, D.J. Medeiros, and M.W. Rohrer. The Society for Computer Simulation, San Diego, CA, pp. 913-920.
- Williams, T. (1996). Simulating the Man-in-the-Loop. *OR Insight*, **9** (4), pp. 17-21.

A SIMULATION MODEL FOR AIRCRAFT MAINTENANCE IN AN UNCERTAIN OPERATIONAL ENVIRONMENT

VILLE MATTILA*
KAI VIRTANEN
TUOMAS RAIVIO

*Systems Analysis Laboratory
Helsinki University of Technology
P.O. Box 1100
FIN-02015 HUT, FINLAND
E-mail:ville.a.mattila@hut.fi

Abstract: We present a discrete-event simulation model for maintenance operations of a fleet of fighter aircraft in crisis situations, where the fleet operations are affected by a threat of an enemy's actions. The model describes the flight process and basic modes of periodic maintenance and failure repairs. Features that are specific to crisis situations include battle damages of the aircraft, decentralization of airbases, specialized maintenance personnel and spares supply. Construction and validation of the model are based on expert knowledge and statistical data on actual flight and maintenance operations in peacetime conditions. The main use of the model is the evaluation of different maintenance strategies in elevated states of readiness and in presence of hostile activities. Built with a graphical simulation software the model provides an easily manageable tool for maintenance designers. In addition, it offers a valuable educational aid in training maintenance personnel by demonstrating the implications of airbase maintenance and logistics activities to fleet performance.

Keywords: Aircraft, maintenance, discrete-event simulation, logistics

1 INTRODUCTION

F-18 Hornet fighters and Hawk Mk51 jet trainers form the basis of the aircraft fleet of the Finnish Air Force (FiAF). The aircraft are used for the different tasks involved in maintaining the nation's air defense such as pilot training and air surveillance. The flight process of the fleet and the related logistic support constitute a system with complex dynamics. Different operating policies, i.e. the use of personnel resources, materials and equipment have to be fitted together to assure that the entire system functions as desired with regard to different operational goals. In peacetime operations, the goals might be the capability to sustain certain long-term level of preparedness or the capability to restore the level in a certain limited period of time. The complexities of the problem are further amplified in states of emergency, where the fleet operates under a threat of an enemy. It is of great importance for the planners of air defense strategies to be able to predict the supportability requirements for the aircraft and the level of performance that can be expected of the fleet.

This paper presents a discrete-event simulation model for analyzing the flight and maintenance operations of a fleet of F-18 Hornet or Hawk Mk51 aircraft in an uncertain operational environment. By uncertain environment, we refer to operating conditions of crisis situations. Compared to normal operations, a greater

uncertainty is involved in the flight and maintenance processes due to the limited knowledge and experience of such circumstances. The actual nature of operations is strongly affected by the actions of the enemy, which are difficult to predict.

From modeling perspective, the implications of the uncertainty of the environment are twofold. The shortage of initial data increases the uncertainty involved in determining the values of model parameters. Furthermore, the selection of an appropriate model form becomes complicated. For example, constructing a model that describes the flight process and sustaining of battle damages of the aircraft in varying operating conditions can be implemented in a number of ways.

The construction and the validation of the simulation model presented in this paper are based on expert knowledge and statistical data on actual peacetime flight and maintenance operations. Experiences on an earlier preliminary study of flight and maintenance operations are utilized in the implementation of the model, see [Raivio et al., 2001]. Thus, the validation of certain components of the model is based on formerly validated simulation results. Due to the absence of data on wartime operations, the use of expert knowledge in the construction and the validation is emphasized. In

addition, the model is aimed at providing an experienced user, such as a maintenance designer, enough flexibility to consider a wide range of scenarios without further programming. Flexibility is accomplished by making alternative model forms available through change of parameters.

The simulation model describes the flight process, failures of the aircraft and different types of maintenance. The characteristics of airbases and maintenance facilities, such as material and personnel resources, are included in the model. Parts of the model that specifically describe crisis situations include the battle damages of the aircraft, the decentralization of airbases, specialized maintenance personnel, and the supply of certain spare parts.

The model is implemented using Arena, a graphical discrete-time simulation modeling environment [Kelton et al., 2001]. A graphical user-friendly environment allows the maintenance designers to use the model independently in studying the effects of different operating policies and conditions on fleet performance. Aircraft availability, defined here as the fraction of mission capable aircraft to their total amount, is used as the primary measure of performance. However, a number of other logistic indicators can be monitored. The model allows dynamically evolving operating conditions which makes it possible for the user to consider multi-phased scenarios. By demonstrating the implications of airbase maintenance and logistics activities to fleet performance, the simulation model also serves as an educational aid in training maintenance personnel.

Simulation approaches have formerly been used in studying availability or supportability requirements of different weapon systems by, e.g., Pohl [1991], who presents a simulation model for flight and maintenance operations of a squadron of F-15E fighter aircraft. The model is used to study the performance of the squadron in both peacetime and wartime scenarios. Kang et al. [1998] examine strategies for reducing repair cycle-times in naval aviation depots. They present a simulation model, which primarily concentrates on the repair of aircraft components that are critical to readiness due to short supply. Balaban et al. [2000] consider the effects of proposed reliability improvement schemes on availability of C-5 Galaxy cargo aircraft through a Monte Carlo simulation model. In [Sadananda and Srinivasan, 2000] and [Cook and DiNicola, 1984] the availability of fleets of aircraft and helicopters, respectively, are modeled. Both of these papers consider battlefield operations.

2 FLIGHT AND MAINTENANCE OPERATIONS

The F-18 Hornet and Hawk Mk51 aircraft of FiAF are primarily operated in three squadrons that are located

in their own air bases. A majority of peacetime flight operations consists of pilot training. Along with the normal daily flying, the aircraft are used in exercises that may, e.g., involve wider scenarios or co-operation of forces. Other types of missions are patrol and identification missions. The daily flight schedules are planned in advance. In the planning process the effects of the cumulated usage and the maintenance requirements of individual aircraft on future flight and maintenance operations are taken into account.

Between flights the aircraft undergo turnaround inspections and replenishments. A pre-flight check is conducted before the first flight of the day. Maintenance of this type is referred to as everyday maintenance. Besides normal tasks, possible component failures are preliminarily analyzed during turnaround inspections. Aircraft that are defined not mission capable are directed to an appropriate repair facility. The aircraft are also subject to damages that are here defined as being caused by some unexpected event and not gradual deterioration of components.

Periodic maintenance constitutes a major part of all maintenance operations. The frequency of periodic maintenance is based on cumulated usage hours of the aircraft. Aircraft manufacturers initially specify maintenance intervals but they are generally later adjusted by the users. These intervals have certain amount of tolerance that allows variability in the actual time between maintenance operations. Thus, the workload of repair shops can be taken into account in the planning of these operations. In FiAF, six levels of periodic maintenance are performed for the Hawks.

The different types of maintenance are carried out in facilities of variable capabilities and resources. Turnaround and preflight inspections, some periodic maintenance as well as minor failure or damage repairs are conducted by each squadron at the airbase. Maintenance of this type is generally referred to as organizational level (O-level) maintenance. The squadrons also have separate aircraft repair shops that are located in the airbases. These repair shops handle more elaborate periodic maintenance and failure repairs. They are referred to as intermediate level (I-level) facilities. The most elaborate maintenance takes place at depot level (D-level) repair shops. In practice, the allocation of tasks to different levels is not strict because the planning of maintenance schedules and the availability of resources affect where the aircraft are ultimately maintained.

2.1 Crisis Situations

As there exists very limited amount of data on maintenance and flight operations in wartime conditions, the knowledge of these circumstances is based on expert judgement of FiAF personnel. In crisis situations, the fleet operates under threat of an enemy,

hereafter referred to as the opponent. Some insight of the nature of these kinds of operating conditions can be gained from war-game-like exercises and contingency plans. However, this data is classified to a large degree and has not been made entirely accessible to the model constructors. The general principle of the modeling effort has therefore been to develop a simulation tool with enough flexibility to allow the end users to independently analyze any scenarios that involve the use of confidential data.

The most evident change in flight operations between normal conditions and crisis situations are the engagements with opponent's aircraft. Subsequently, the fleet may suffer losses in the form of damaged or destroyed aircraft. Also, the average flight intensity most likely increases during a crisis. The flight pattern in wartime operations may be very uneven, with periods of high and low intensity operations recurring randomly.

Changes in the flight operations add to the requirements of the maintenance system. Besides battle damages, increased flight intensity increases the need for failure repairs. Demands for aircraft maintenance are further amplified by alterations in the nature of actual maintenance tasks. Larger amount of maintenance consists of failure and damage repairs. Furthermore, there is a pressure to restore the aircraft to a mission capable condition as quickly as possible. In crisis situations, non-critical maintenance can be discarded in order to ease the workload of repair shops.

The squadrons may be required to decentralize their operations and use alternate airbases that are located as to provide better defense against the threat caused by the opponent. These alternate airbases are categorized into three levels according to existing infrastructure and operational capabilities. Class I airbases refer to such facilities that can respond to all operational needs of a squadron. Basically, they correspond to the main airbases of peacetime operations. Class II and III airbases lack some of the operational capabilities and may, e.g., not be able to conduct certain elaborate maintenance tasks. In a decentralized setting, a squadron operates from multiple airbases. Benefits of decentralization include the added flexibility in directing the use of forces. However, relying on a cut down infrastructure can affect the conduction of maintenance activities or the supply of materials.

3 THE SIMULATION MODEL

The simulation model of the flight and maintenance processes describes the operations of three squadrons and a central depot-level maintenance facility. The structure of the model is presented in Figure 1. The arrows with solid lines represent the movement of the

aircraft between different processes. The dashed lines describe material and information flows.

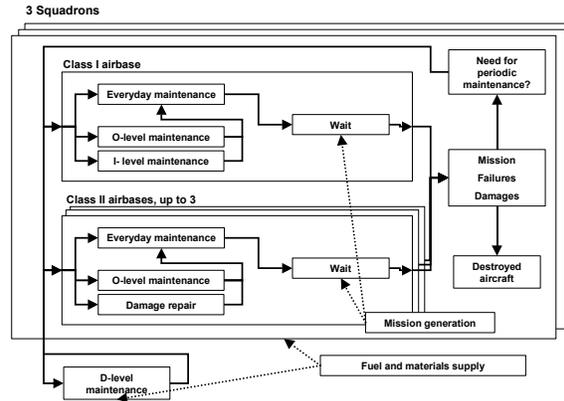


Figure 1: Structure of the simulation model

In general, all the essential characteristics of the flight and maintenance operations are included as input parameters in the model. Moreover, these parameters can be arbitrarily changed during simulation runs enabling the user to study dynamically evolving operating conditions or effects of changes in operating policies. This means that also the model structure is flexible to a certain extent. The aim has been to reduce the uncertainty that relates to the selection of the model form. A user that is knowledgeable of the underlying system and its operating environment is allowed as much freedom as possible in determining the characteristics of the model. For instance, the airbases of the squadrons can be removed out of use or introduced at all times making it possible to describe, e.g., the transfer of airbases to varying geographical locations.

In the model, the squadrons operate independently of each other. By default, the aircraft are always directed to their own airbases unless this airbase cannot conduct a required maintenance task. For simplicity, all operational aircraft are involved in the flight activities. Times between flight missions follow an exponential distribution. If a sufficient amount of operational aircraft do not exist, the mission is either carried out with fewer aircraft or discarded. The main interest in the flights is focused on the accumulation of flight hours and the occurrence of failures and battle damages. The missions do not involve specific objectives and discarding a mission does not have effect on other activities in the model. All missions that were conducted with fewer aircraft than required or that were discarded are simply registered in the simulation results.

Aircraft maintenance in the airbases is organized into different level facilities as described in the previous section. All airbases conduct everyday maintenance as well as O-level tasks. Class I airbases include a

separate I-level aircraft repair shop which is in class II airbases replaced by a certain amount of additional maintenance personnel that is specialized in damage repairs. Depot-level level maintenance takes place at a single central facility that serves all squadrons.

Aircraft requiring maintenance are directly transferred to appropriate maintenance facilities. The need for periodic maintenance is determined based on accumulated flying hours and pre-specified maintenance intervals. Failures of the aircraft also occur depending on accumulated flight hours. Times between failures are assumed exponentially distributed. For each occurrence the type of malfunction is defined randomly according to type specific probabilities. Six types of failures can be defined in the simulation model. Similarly, the model contains six types of battle damages. Aircraft that carry out a flight mission face hostile aircraft with a certain probability. If an encounter occurs, the aircraft are damaged or destroyed with assigned probabilities.

In the simulation model, all maintenance facilities have their own personnel. A resource requirement and a distribution of the task time is associated with each maintenance type. The actual time required to complete the maintenance task is calculated by dividing the initial duration with allocated number of mechanics. The aircraft are maintained in order of arrival, i.e., no prioritization of jobs is considered in the model. Variation in maintenance manpower due to holidays, sicknesses or other absences is not taken into account. Thus, the number of maintenance personnel describes the effective available manpower.

During flight missions the aircraft spend fuel and certain munitions and countermeasures. In addition, spare part requirements can be associated to all types of periodic maintenance as well as failure and damage repairs. Material inventories of the airbases are replenished according to a specified order point. Alternatively, new materials may be separately acquired each time a need arises.

3.1 Estimation of Input Parameters

Base values for maintenance times, failure and flight intensities as well as parameters related to the characteristics of the airbases are defined using data on normal operations of Hawk Mk51 aircraft and expert knowledge of FiAF personnel. In crisis situations, operating conditions for the fleet are largely dependent on the threat scenario under consideration. Input parameters are therefore chosen individually for each scenario. As initial data is scarce the parameters are necessarily based on expert judgement and existing contingency plans for war-time operations. The base values provide a starting point for definition of these parameters.

Raw statistical data for estimating I- and D-level periodic maintenance times is available from one I-level facility. Additionally, estimated values for mean and variance of maintenance times in one of the depot-level repair shops are at disposal. Based on graphs of the raw data, alternative models for maintenance times include several probability distributions. Statistical tests show that distributions with right-sided tail are a more suitable choice compared to symmetric distributions. As the number of observations is somewhat limited for certain maintenance types, stronger conclusions cannot be made. Ultimately, the gamma distribution has been chosen as the model for the duration of all periodic maintenance types. It shows a reasonably good fit for all data sets and in particular, provides a good fit for those types of which most observations exist. Gamma distribution is commonly used to model different task times [Law and Kelton, 2000].

Normal distribution is chosen as the model for the turnaround and pre-flight inspections. Justification for the choice is that the contents of these maintenance types generally remains fairly unchanged. Elaborate periodic maintenance may involve considerable amount of additional tasks such as delayed repair of non-critical failures causing the distribution of the maintenance duration to be skewed. Maintenance types with fewer tasks have less variability in contents and are less likely to be severely delayed even in individual cases. Values for the mean and standard deviation of the duration are provided as subjective estimates of maintenance personnel.

The mean and standard deviation of failure repair times as well as the mean time between failures are directly available from reference data provided by FiAF. Times between failures are assumed exponentially distributed. Failure repair times, on the other hand, are assumed to follow the gamma distribution. Failure repair times are commonly modeled with non-symmetric distributions such as gamma or exponential distributions. For simplicity, the gamma distribution was chosen in this case, since it provides as good a model as other right-tailed distributions with regard to the available data.

Average flight intensity and average flight duration are defined on the basis of statistical data on all missions of the Hawks during a time period of one year. The amounts of accumulated flight hours for each aircraft are also available from these statistics.

3.2 Validation of the Model

Since reference data does not currently exist on some aspects of the system under consideration, the process of verification and validation of the model relies, to a certain degree, on subjective measures. Close collaboration between the model constructors and the

representatives of FiAF has been maintained during the entire modeling effort. Additionally, the model structure, its underlying assumptions, principles of implementation, input parameter values and ultimately simulation results have been presented to a variety of logistics and maintenance personnel of different organizational levels. These reviews have taken place throughout the modeling process and feedback from these occasions has been actively utilized to further develop the model. Preceding the final accreditation, the simulation model will undergo independent tests of the end user.

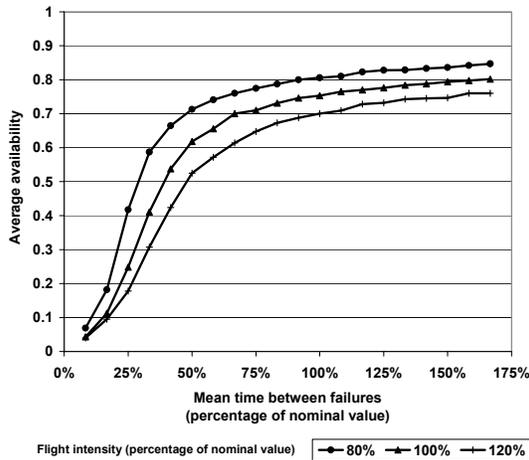


Figure 2: Sensitivity of the average availability to flight and failure intensities

As the characteristics of the model are adjustable to a large extent, it can describe normal operations with suitably chosen values of input parameters. The simulation results can therefore be partially validated with reference data from actual flight and maintenance operations in normal conditions. Available data includes values of aircraft availability from a period of four years. This data exists in the form of 3- and 12-month moving averages. The simulation model predicts an average availability of approximately 75% for normal operations, which differs slightly from the actual value. The difference is most likely due to the simplifying assumptions such as the exclusion of certain types of maintenance and administrative delays from the model.

The validity of the current model may also be assessed by comparing its outputs with other simulation results. The preliminary study of the flight and maintenance operations provides a validated model for this purpose, see [Raivio et al., 2001]. Sensitivity analyses were conducted to find out how responses of the current model are affected by variations in important input parameters and to evaluate the extent to which these results differ from those of the other model. Figure 2 shows an example analysis of the current model. In the example, the sensitivity of average availability to flight

and failure intensities in normal operating conditions is considered. The results show similar behavior with those of the earlier model and are therefore regarded as valid.

4 EXAMPLE SIMULATION

As an example of possible applications of the model, a scenario with dynamically evolving operating conditions is presented. We study how the timing of a change in maintenance policy affects fleet performance and specifically aircraft availability.

In the scenario, the operating conditions are assumed to change in four phases. In the first phase, the state of readiness is elevated and the flight intensity increases compared to normal operations. The second phase involves further increase in the amount of flight missions. Additionally, the operations of the squadrons are decentralized into four airbases. The third phase represents the transition to the actual combat phase as the squadrons respond to activities of the opponent. During the missions, the aircraft may be damaged or destroyed. In the fourth phase, the intensity of the combat decreases as the opponent is assumed to suffer losses that limit its operational capabilities.

The change in the maintenance policy involves discarding most of the periodic maintenance in order to release more aircraft to flight operations. Thus, the example examines one alternative periodic maintenance strategy compared to the maintenance program of normal operations. The alternative policy is applied to all aircraft with no exceptions. It might seem desirable to consider each maintenance decision separately, i.e., whether individual aircraft could be maintained during periods of lower flight intensity. In a highly uncertain environment, this would, however, entail a certain amount of risk by reserving the resources for non-critical operations. The assumption of no exceptions can here be regarded as reasonable.

The new policy is employed at the start of one of the first three phases of the scenario. These transition times can be thought of representing time instants where new strategic information on the opponent is received. At each time instant, the commander of operations is to decide on the course of action in the changed circumstances. Figure 3 presents the simulation results for the three cases where the new maintenance policy is employed and for the case where the policy is not employed. The plotted availability figures represent values that are averaged across 20 independent replications.

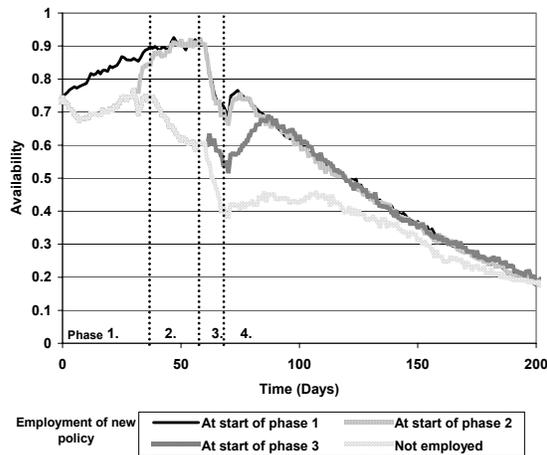


Figure 3: Effect of the timing of maintenance policy change on aircraft availability

Figure 3 clearly shows that some types of periodic maintenance have to be given up to maintain the capability of fulfilling the operational requirements of a high intensity crisis. Depending on how quickly the situation evolves, certain amount of periodic maintenance may be conducted at the early stages of the crisis. If all maintenance is completed without pre-emptions, the availability will, however, rise rather slowly.

For further conclusions sensitivity analyses are required to assess the effect of the underlying assumptions of the example. These assumptions are mainly concerned with the usage of resources in the airbases and the strategy of the opponent. Furthermore, the short-term effect of discarding periodic maintenance on failure intensity of the aircraft has to be considered.

5 DISCUSSION

The primary objective of the modeling effort is to gain new insight into the effect of maintenance policies and operating conditions on the overall performance of the aircraft fleet of the Finnish Air Force. The presented simulation model describes the essential features of the flight and maintenance operations in both normal conditions and crisis situations, where the fleet is faced with added operational uncertainty. The model provides a way to quantitatively assess the effects of proposed improvements to the maintenance system. The model is constructed and validated in close cooperation with representatives of FiAF. In addition, it is implemented with graphical simulation software and thus allows easily manageable simulation analyses.

The introduction of the model to FiAF has been started by initiating a user-training process for potential end users. The training aims at familiarizing these users to the general objectives of the modeling effort, principles of the simulation methodology and the

features of the model. Simultaneously, the training process serves as a way to collect feedback on the functionality of the model to support its further development and updating. Training is also necessary to assure that the simulation methodology will be correctly applied and its limitations are understood by the user-organization. Overall, early experiences of the use of the model suggest, that the model provides the Air Force a valuable aid in the design of aircraft maintenance policies and education of maintenance personnel.

REFERENCES

- Balaban H.S., Brigantic R.T., Wright S.A. and Papatyi A.F. 2000, "A Simulation Approach to Estimating Aircraft Mission Capable Rates for the United States Air Force". In *Proc. 2000 Winter Simulation Conf.* (Orlando, FL, December). Pp1035-1042.
- Cook T.N. and DiNicola R.C. 1984, "Modelling Combat Maintenance Operations". In *Proc. 1984 Annual Reliability and Maintainability Symposium* (San Francisco, CA, January). Pp390-395.
- Kang, K., Gue K.R. and Eaton D.R. 1998, "Cycle Time Reduction for Naval Aviation Depots". In *Proc. 1998 Winter Simulation Conf.* (Washington DC, December). Pp907-914.
- Law A.M. and Kelton W.D. 2000, "Simulation Modeling and analysis". McGraw-Hill.
- Raivio T., Kuumola E., Mattila V.A., Virtanen K. and Hämäläinen R.P. 2001, "A Simulation Model for Aircraft Maintenance and Availability". In *Proc. 2001 European Simulation Multiconference* (Prague, Czech Republic, June). Pp190-194.
- Kelton W.D., Sadowski R.P. and Sadowski D.A. 2001, "Simulation with Arena". McGraw-Hill.
- Pohl, L.M. 1991, "Evaluation of F-15 Availability during Operational Test". In *Proc. 1991 Winter Simulation Conf.* (Phoenix, AZ, December). Pp549-554.
- Sadananda, U. and Srinivasan N. 2000, "A Simulation Model for Availability under Battlefield Situations", *Simulation*, Vol. 74, No. 6. Pp332-339.

BIOGRAPHY

Ville Mattila received his M.Sc. degree in industrial engineering and management in Helsinki University of Technology in 2002. He currently works as a researcher and conducts his Ph.D. studies at Systems Analysis Laboratory in the Department of Engineering Physics and Mathematics. His research interests include simulation and optimization of discrete-event systems, specifically aircraft maintenance and airbase logistics systems.

VERIFICATION MODEL STRUCTURES FOR DIGITAL SYSTEMS DESIGN

SERGEY L. FRENKEL

*The Institute of Informatics Problems,
Russian Academy of Sciences,
Vavilova 44,2, 117333, Moscow, Russia.
E-mail: slf-ipiran@mtu-net.ru*

Abstract Exponential computational complexity of digital systems formal verification algorithms excludes any possibilities of full-automatic verification of complex digital systems. On the other hand, the informal design simulation is also impractical time-consuming. Probably, the possible outcome is to form a verification strategy which, on one hand would combine both approaches, and on the other hand would include a guide to issue verifications algorithms-and-tools appropriate for a given design. It implies a characterization of both verification algorithms and design process. In fact, it means a structurization of various models of design, which are used both explicitly and implicitly during design verification activity.

This paper, relying on the previous experience in testability design planning [1] as well as corrent publications in formal verification areas considers some possibilities of planning of digital systems verification activity to achieve high degree of functional verification.

Key words: formal verification, design verification, digital circuits simulation.

1. INTRODUCTION

Designers of complex digital systems (ASIC, application-specific/general-purpose microprocessors (MP)), etc.) need validation methods and tools to guarantee a perfect design before a process of its manufacturing is started. Errors detected after start of fabrication lead both to added production costs and delay the product. This delay may be very critical issue of market control. For example, some data in [2] shows that loss due to late marketing for 10-15 weeks may be up to half million USD.

This validation is performed mostly as a “verification”, checking if a system design is correct with respect to a specification (which is understood here as an initial description of aimed design on a given representation level (e.g., finite –state machine, register-transfer (RTL), or gate level).

The traditional and the most common method of the verification is verification via simulation. The alternative is so-called “formal verification” [3]. However, both these approaches have some drawbacks of high computational requirements. Thereby, the complexity of simulation-based methods is due to the large number of test vectors needed to manifest all functional issues, and the complexity of

the formal verification of large designs is due to very large state spaces, which cannot be handled even by such techniques as implicit state space traversal.

For example, in sequential circuits verification a central problem is the reachability analysis. In this activity, the properties to be checked by an automatic verification have to be reachable from the start state. Reachability analysis is the task of finding this set. If a system is represented as a finite-state machine (FSM), reachability analysis corresponds to a traversal of the state transition graph of an FSM, that as it is well-known may contain billions of nodes [3].

Strictly speaking, the same situation from the point of view of automatic (synthesis-directed) and simulation methods interaction takes place in other areas of Electronic CAD activities, first of all in test pattern generation (TPG). In this area a test designer also has to consider a trade-off between the exponential complexity of automatic test pattern generation (ATPG) (true synthesis) and the necessity to use various simulation tools (a “synthesis through analysis”) to check if an input test vector (“candidate to test”) provides detection of a fault considered. In fact, this methodology changing means the change of *design specification model*. While the design specification for ATPG consists merely of the

circuit description, the simulation-based TPG require also explicit input vector set description.

In other words, the reasonability of using either design methodologies depends on the suitable test design cost, which depends on labor force cost, equipment cost, time-to-market etc. A choice of ways of the design goal achieving can be considered as a design strategy planning. Various design testability measures may be used as the cost function during the strategy planning [1]. As it has been shown in this work, any relevant design cost function should be in monotony dependence on any testability measure, thereby, any fault coverage measure is a functional of the testability ones (with given TPG methods). Also note, that ATPG practicability depends greatly on the used fault model. Definition of well-known stuck-at-fault model in sixties [4] has led to ATPG performance increase dramatically. This is because of the considerable decreasing of the considered faults set. So, such issue should be studied for formal verification (FV) activity, namely what kind of bugs detecting model could be more reasonable.

In this paper we consider some factors of design functional verification cost together with various aspects of verification models and design features.

Let us emphasize that one should distinguish between the *verification algorithms development* activity and design functional verification activity *on the whole*.

In the first case the computational complexity of verification algorithm will serve as an indicator of practicability, while (besides the algorithm complexity) a complex cost function (labor cost, equipment cost, design tools cost, time-to-market) is a reasonable indicator in the second case. Thereby, a verification algorithm properties are only part of factors of design verification cost.

So, let us consider what kind of means may a designer use to control mentioned above factors planning his design verification activity.

For this aim, in Section 2 all principle components of this activity will be outlined. Section 3 describes some well-known tools from the point of view of design specification impact on overall verification cost. Sections 4-5 describe some design properties and possible design decomposition techniques.

2. ABOUT VERIFICATION PROCESS AND ITS COMPLEXITY

Design verification (Model checking [3], in particular) activity in industry uses the following methodology: A verification engineer reads the

specification, sets up a work environment and then proceeds to present the model checker with a sequence of properties in order to verify the design correctness. A design can be quite large nowadays. As a result the set of properties written and verified becomes large as well, to the point that the engineer loses control over it.

One of the basic questions is: "Have I described enough properties?" [5]. The current solutions consist in manually reviewing of the property set. It is important that the decision if it is possible to verify the correctness (both functional and timing) of a given design depends on many organizational issues. In fact, these issues are determined by the cost (either in money or in labor time terms) of the result obtaining, and, in the end, depend on the verification process planning and organization. Obviously, to provide the verification scenario planning we have to consider and define all features of target system having an impact on the verification algorithm complexity and, correspondingly, on the choice of preferable verification algorithm, some characteristic of design process to guide a verification process, supposing, first of all, that a cost model of design verification process is available.

In general, we can represent amount cost C_V of a design verification as :

$$C_H^M + C_E^M + C_{ad}$$

where index $M = \{fv, s\}$ reflects one of verification method, namely, either formal verification (fv) or via simulation (s),

C_H^M means a "human" cost factor, which is the cost of various verification models development and manual input data preparing (and, maybe, a software supporting and modification), C_E^M is an equipment cost including, for example, amortization cost, the "machine time" spent up to verification result, a software acquisition cost, power resources costs etc., C_{ad} is any additional expenses. These partial costs depend on the way of verification.

As we try to deal merely with formal methods, the human (or manual) component of the above expenses correspond to a model development and description (ideally, using merely some hardware description language, e.g. VHDL, and design properties (e.g., in terms of some temporal logic [3]), and, maybe, to some programming activity. This work requires very high qualification of a verification engineer who has to know all modern logical-mathematical techniques (computational tree logic (CTL), model checking, etc.). Time of a verification algorithms execution

depends both on the algorithm and size of the circuit designed. Obviously, C_E is a monotone-increasing function of the time. Formally, the model checking algorithms are linear with respect to number of states, but, the number of the states increase exponentially with number of terms of logical formulas describing the verification conditions. Thus the “machine” cost of formal verification is $C_E^{fv} \sim f_p(N)$, where $f_p(N)$ is a power function (exponential, in particular) of number of variables, describing the verification problem. Note, that even dealing with some components of entire systems e.g., with some buffers of a microprocessors [Biesse01] we encounter with thousands of variables, that leads to huge numbers of states. Correspondingly, several days may be required to check simple properties of such designs even using rather power platforms, e.g. 700 MHz 64-bit Alpha [6].

As for design bugs finding via simulation, then $C_H^M = C_H^R + C_H^F$, where C_H^R stands for random testing (simulation our design under random-generated tests to observe the design bugs), C_H^F corresponds to cost of so-called “focused” testing, which are some hand-generated tests to cover specific areas of design, not covered by the random tests. Obviously, this activity supposes some involving of the design developers. For example, the tests may be focused to detect some bugs of caching mechanisms, ALU, etc.

However, it should be taken into account that such activity may require to involve many technicians in the simulation process to run hundreds focused tests variants! Although, in general, the computational of computational complexity of simulation is a quadratic relatively to variables number, C_E^S should not be considered as such function, because the simulation of various parts of the design usually is very redundant from the point of view of design bug checked. So, although for separate design components as a rule $C_H^S < C_E^{fv}$, it may be not true for the design as a whole. So, the way out should be based on trade-off between using of simulation and formal verification approach. The table 1 shows a typical example (verification of a memory bus adapter design) of this compromise [7].

Table 1 Design bugs detected with various techniques

Verification techniques	Bugs founds (%)
separate unit simulation	41
formal verification	24
visual design analysis	20
entire chip simulation	15

However, from the point of view of labor cost, an increase of formal verification weight would be very attractively.

Note that besides the computation complexity, simulation-based methods are no longer adequate for complex hardware (HW) designs. Although simulation can catch many design error, part of bugs are frequently sleeping through. Detecting by simulation of every bug resulting from the complex interaction of concurrent event may be very time-consuming task. In particular, in the considered instance, about 40% of bugs that had been found with formal verification, it turned out impractically to find with any simulation tools [7].

Let us consider some possibility of formal verification (FV) cost reducing.

For this aim we must define and fix, on one hand, various properties of FV algorithms/tools, and on the other hand, various design features affecting the FV cost.

Since this is a combinatorial problem, and as it is well known, combinatorial algorithms may mostly be realized only by the problem description decomposition, we need also to have a characteristics of decomposition ability.

3. VERIFICATION TOOLS AND DESIGN SPECIFICATION ACTIVITY

Since effectiveness of design verification depends on adequacy of logical functions verified representation, it is important that a design tools selected for FV activity would allow the using of various Boolean function representation techniques. Thereby, this representation may depend on both design and requirements specification manner (model).

For example, in [8] the highest level description of a microprocessor is given as an instruction-set specification. At this level the verification may be performed either from actual pipeline design description or representing the stream of executed instructions with a table [8], which describes the effect of individual instructions.

However, there are many properties of the pipelined machine and instructions that can be more easily expressed and reasoned about with help of some tables [8]. For example, a *Read After Write dependency* between instructions is much easier to represent using our instruction table instead of lifting the necessary information from design.

The basic structure of the design specification for formal verification is Computational Tree Logic

(CTL) [3]. For example, well-known VIS package [The release 1.4. of VIS: <http://vlsi.colorado.edu/~vis>] uses a Verilog front-end and supports fair model CTL checking, language emptiness checking, combinational and sequential equivalence checking, cycle-based simulation, and hierarchical synthesis.

In a program called EMC (Extended Model Checker [9]) the model checking is solved using efficient graph-traversal techniques. Thereby, if the model is represented as a state transition graph, the complexity of the algorithm is linear in the size of the graph and in the length of the formula. However, an explosion in the size of the model may occur when the state transition graph is extracted from a finite state concurrent system that has many processes or components (e.g. dealing with simultaneously-performed six instructions in Alpha processor [10]).

The CUDD package provides functions to manipulate Binary Decision Diagrams (BDDs) [<http://vlsi.colorado.edu/~fabio/CUDD/cuddIntro.html>], and Zero-suppressed Binary Decision Diagrams (ZDDs represent switching functions like BDDs, however, they are much more efficient than BDDs when the functions to be represented are characteristic functions of cube sets, or in general, when the ON-set of the function to be represented is very sparse. But they are inferior to BDDs in other cases.). The CUDD package can be used in three ways:

- As a black box. In this case, the application program that needs to manipulate decision diagrams only uses the exported functions of the package. The rich set of functions included in the CUDD package allows many applications to be written in this way.
- As a clear box. When writing a sophisticated application based on decision diagrams, efficiency often dictates that some functions should be implemented as direct recursive manipulation of the diagrams, instead of being written in terms of existing primitive functions.
- Through an interface. Object-oriented languages like C++ and Perl5 can free the programmer from the burden of memory management.

In the package Almanac, (developed at the LaBRI (Universit e Bordeaux-1)) a *Heuristic methods* based on analysis of the original boolean formula abound is used, and can be subdivided into *static* techniques, that inspect the formula off-line.

Being very popular, these dynamic methods present many problems. The first is that they require that we have already constructed the BDD or some part of it in memory, which is impossible for large systems. A

more troublesome problem is that existing techniques are based on *sifting*, which exchanges adjacent variables. Unfortunately, in real systems variables come in *blocks* of related variables, that need to be kept together in the final order or the size explodes.

Note, that the best known BDD-based algorithm for finding an optimal order is of complexity $O(n^3)$, where n is a number of variables.

In tools which are based on Bounded Model Checking [3] accept a subset of the SMV (Symbolic Model Verification) language in which the user can specify a finite state machine and a temporal specification.

Given a bound k , BMC outputs a propositional formula which is satisfiable iff there is a counterexample of length k . An efficient implementation of the Davis-Putnam technique [11] and PROVER [12] are based on Stalmarck's method to decide propositional satisfiability.

Note, that a lot of modern tools are based on a philosophy of "Satisfiability solvers" (SAT)-base model checking, which sometime is considered as an alternative to BDD approach (although, as remarked [13] SAT may be considered as "an interesting complement to model checking with BDDs"). In general, SAT algorithms mission is to decide whether there exists a satisfying assignment for the corresponding formula. Thereby, in spite of mentioned above remark on relationship of BDD and SAT techniques, in [6] was shown that the SAT method for bounded model checking can reduce the verification runtime from days to minutes on real, deep, microprocessor bugs when compared to a state-of-the-art BDD-based model checker.

So, basic features of algorithms underlining various verification tools are good basis for their comparison in the framework of a verification procedure planning.

4. SOME EXAMPLES OF TARGET DESIGN PROPERTIES IMPACT

Intuitively, the complexity of the BDD is a function of how much information must be remembered as one passes from one level of the BDD to the next (i.e., from one variable to the next). For example, in [14] a pipeline examples which were verified had approximately $5 \cdot 10^{20}$ states, which puts it far outside the range of model checkers like the one reported in [3]. It required a BDD with 42000 nodes to represent the transition relation. These data are concerned very simple pipelines that perform three-

address logical and arithmetic operations on a register file. The complete state of the register file and pipe registers are modeled. The pipelines in this design had three stages. On the first stage, the operands are read from the register file, on the second stage an ALU operation is performed, and on the third stage the result is written back to the register file. ALU has a register bypass path, which allows the result of an ALU operation to be used immediately as an operand on the next clock cycle, as is typical in RISC instruction pipelines. The inputs to the circuits are an instruction code, containing the register addresses of the source and destination operands, and a STALL signal, which indicates that the instruction stream is stalled.

However, what kind of the circuit's properties enabled such impressive results?

The point is that the information stored from one "bit slice" of the data path to the next was rather small; it amounts to the state of the control bits plus at most the value of the ALU "carry" bit. In particular, this amount of information is not increased as one increases the number of bits, so the BDD becomes deeper, but no "wider".

Although these research [14] are concerned the timing verification, these conclusions are true also for functional verification as in both cases verification algorithms use a Boolean encoding of the elements of the model domain, and represents relations with Boolean decision diagrams.

So, in case the information quantity stored from one "bit slice" of the data path to the next is a system designed characteristics affected the BDD using effectiveness.

5. ABOUT DECOMPOSITION POSSIBILITIES

Let's consider what current state-of-the-art in formal verification may suggest us to decompose design as a way of verification cost reduction.

Mostly a design description decomposition is trying to avoid the state explosion problem. The goal is to verify properties of individual components, infer that these hold in the complete system, and use them to deduce additional properties of the system. It may also be necessary to make assumptions about the environment (that is both other components of the system and various external signals). This approach may be exemplified by Pnueli's assume-guarantee paradigm [15]. A formula is true if whenever M is a part of system satisfying', the system must also satisfy.

Since we consider this problem from the point of view of design tool using, let us consider what kind of requirements the model checking should meet to.

First of all, it must be able to check that a property is true for all systems which can be built using a given component. More generally, it must be able to restrict to a given class of environments when doing this check. It must also provide facilities for performing temporal reasoning. Most existing model checkers were not designed to provide these facilities. Instead, they typically assume that they are given complete systems. A way to obtain a system with the above properties is to provide a preorder on the finite state models that captures the notion of "more behaviors" and to use a logic whose semantics relate to the preorder [16, 17].

Note, that along with design decomposition it can be used also various types of circuit's reduction. For example, the merge buffer, an important component of the Alpha MBox for a next-generation Alpha chip has been considered in [6]. The function of the merge buffer is to receive requests to write into memory, and to reduce the trajectory on the memory bus by merging stores to the same physical address. The merge buffer is essentially a large buffer with a very complex policy for reading in entries, merging stores, and writing out stores to the memory. It has about 14 400 latches, 400 primary inputs, and 15 pipeline stages. The pipeline has complex feedback that prevents us from retiming away latches. The original RTL description of the circuit is used as design input.

First of all, the authors tried to reduce the size of the model for verification using standard model checking technology. The idea is to remove portions of the state in the circuit in ways that do not alter the circuit behavior with respect to the properties of interest. After the reductions, the merge buffer has about 40 primary inputs. When the merge buffer is in use, these inputs will be connected to the four subboxes with which the merge buffer communicates. The final model has about 600 state nodes in the cone of most properties. However, before sending the model to a tool input, it is needed to write down the property of interest in a format that the tool we want to use accepts. Given the model and the property, the verification tool then either produces a failure trace, or tells us that the property is true.

6. DISCUSSION AND CONCLUSION

Full-automatic formal verification of complex processors design is a dream of all system designers. Unfortunately, its exponential complexity is well

known, that, it seems, excludes this dream realization, at least for large designs with very large state spaces, which cannot be handled even by techniques such as implicit state space traversal. Obviously, the result of such activity has to be obtained even if system description is so large (either in terms of state space or formulas clauses number) that no formal verification algorithm which could allow to do it Very obvious way to achieve it is a combination of formal and informal (simulation) verification models. Since complex microprocessors systems design verification activity deals, in general, with many optional variants, it should be useful to have a characterization of both verification algorithms and verification process on a whole, which includes the decomposition issues, dividing possible (potential) design bugs classes between formal verification and simulation, final quality analysis etc.

Obviously, we need a guide to provide this hybridization. Following well showed itself conception of coverage analysis, use widely in test pattern generation practice, it would be very attractively to have also similar one for the design formal verification. Some steps towards this notion development are just in progress [Hoscotte99, Chochler01]. As for formal verification, the notion of coverage in functional verification is to cover the entire functionality specification required from the implementation. This notion involves two questions:

-whether we can provide (to take into account) (explicitly or implicitly) all possible input sequence,

- whether the specification contains a sufficient set of properties.

So, the coverage analysis is a search of some dissimilarity between the implementation and specification, which points out a possibility to reduce target design description to enhance the verification possibilities.

Along with coverage characteristic, it is important, not resolved problem is to characterize both formal verification and simulation tools that could be chosen for design verification. They may be characterized by a rate characteristic, e.g. as a number of states per second for formulas. Thereby, on one hand, this rate depends on a way, in which design specification is described, and on the other hand, specification language may determine qualification requirement of personal, affecting the verification cost (e.g., time consuming).

REFERENCES

1. S. Frenkel, Testability Measure as a Test Pattern Generation Cost Function, in *Proc. of 7th IEEE North-Atlantic Workshop (NATW'98)*, West Greenwich, RI, USA, 1998, pp.42-50,
2. W. Rosenstiel, Rapid Prototyping, Emulation and Hardware/Software Co-debugging, in "SYSTEM-LEVEL SYNTHESIS" ed. by A. A. Jerraya and J. Mermet, NATO Science Series, Kluwer Academic Publisher, 1998, pp.219-262,
3. E. Clarke 1, O. Grumberg, and D. Long, "Model Checking", in Springer-Verlag Nato ASI Series F, Volume 152, 1996,
4. R.G. Bennets, Design of Testable Logic Circuits, Addison-Wesley Pub. Company, 1984
5. S. Katz, D. Geist, and O. Grumberg. "Have I written enough properties?" a method of comparison between specification and implementation. In *10th CHARME, LNCS 1703*, pp. 280–297, 1999,
6. Per Bjesse, Tim Leonard, and Abdel Mokkedem, Finding Bugs in an Alpha Microprocessor Using Satisfiability Solvers, in Proceedings of 13th International Conference, CAV 2001, Paris, France, July 18-22, 2001, LNCS 2102, p. 454,
7. T. Shlipf et al, Formal verification made easy, in IBM Journal of Research and Development, vol. 41, No 4/5, 1997,
8. Jun Sawada Design Verification of Advanced Pipelined Machines, Doctoral Dissertation, University of Texas at Austin, Computer Science Dept, 1996,
9. M. Clarke, O. Grumberg, D.E. Long: "Model Checking and Abstarction", ACM-TOPLAS, Vol. No. 5, pp.1512- , September 1994.
10. Alpha 21264 Microprocessor Hardware Reference Manual, Compaq Computer Corporation 2000,
11. E. Dantsin et al, Algorithms for SAT and Upper Bounds on Their Complexity, Electronic Colloquium on Computational Complexity, Report No. 12 (2001),
12. M. Sheeran, S. Singh, and G. Stalmarck, Checking safety properties using induction and a SAT-solver. In Formal Methods in Computer Aided Design, 2000,
13. Per Bjesse and Koen Claessen, SAT-based Verification without State Space Traversal 2000, Journal of Global Optimization, , 1{36}
14. J. R. Burch, E. M. Clarke, K. L. McMillan, and D. L. Dill, \Symbolic Model Checking:10 20

- States and Beyond," Information and Computation, vol. 98, no. 2, pp. 142-170, 1992.
15. A. Pnueli. In transition for global to modular temporal reasoning about programs. In K. R. Apt, editor, Logics and Models of Concurrent Systems, volume 13 of NATO ASI series. Series F, Computer and system sciences. Springer-Verlag, 1984.]
16. Yatin Hoskote, Timothy Kam, Pei-Hsin Ho, Xudong Zhao, Coverage Estimation for Symbolic Model Checking, DAC'99, p.300
17. Hana Chockler et al, A Practical Approach to Coverage in Model Checking, in Proceedings of 13th International Conference, CAV 2001, Paris, France, July 18-22, 2001, LNCS 2102, p. 66.

VISUALIZING THE CREATION OF DYNAMIC SYSTEMS SIMULATIONS

RYSZARD TOLWINSKI

*Department of Computer Science of the Bialystok University of Technology
Wiejska 45A, 15-351 Bialystok, Poland
rystol@ii.pb.bialystok.pl*

Abstract: In this paper comparison of the possibility of the creation of the program simulating dynamic systems are presented. There are shown advantages and disadvantages intended for the general package – JavaBeans and the package for the simulation of dynamic – PASION.

Also there are presented the suggestions multiplying the visualization of the simulating programs especially in the computing net.

Keywords: distributed programming, cooperating multiagent system, PASION.

INTRODUCTION

Before we form the simulation of the dynamic systems we should choose the language or program's package which will satisfy our criterions, such as:

- it would be able to solve different problems,
- it is the package which has been made to simulate components,
- it let programs to create,
- using language and package is free or cost very little.

In this moment there is very much languages, packages and program's platforms that satisfy all criterions but not criterion price. They are expensive. So we need the language or the package that are free (JAVA) or cost very little (PASION). Both programs permit creation the simulating systems with using units (JavaBeans) or standard units (PASION). If we use standard dynamic control units we need not know the mathematics description. We have to know only block diagram.

In JavaBeans the particular beans may be locate in Netware computing net. This haven't got PASION shown properties.

In this paper will be same suggestions create end employed simulate programs on the localhost end in Internet too. During simulate dynamic arrangement the visualization is situated in three stage:

1. Analyze of work of real system,
2. Create computer program,
3. Exit of output results.

VISUALIZING THE ACTION REAL ARRANGEMENT

Before we begin the creation of simulating dynamics arrangement the programmer has to recognize: detailed activity system, its structure, the functions of particular elements, relationship and dependences between units of system, simplification and so on.

Presentation dynamic's systems description with use block diagram which consist of typical elements of control system is very convenient. Fig.1 shows control system consisting of the typical dynamics units.

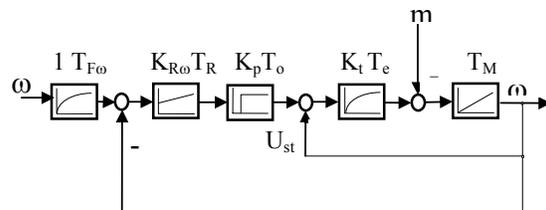


Fig.1. Block diagram of the direct speed control with the dc motor.

VISUAL ANALYZE OF ORGANIZATION PROGRAM OF SIMULATED ARRANGEMENT

Very important thing, with view point see of programmer, is graphic presentation concept of program. It can preserve us not to make deferent errors. Fig. 2 shows organization the calculating with use the specialist agents that have been situated in localhost or computing net [Tolwinski R 2002].

If we take advantage of client-server model in the computing net we should show how the calculation will be organized and where should be located particular objects in the net. On fig.3 the server has to parallel serves of many clients. Each new client which is approaching to server allocates a new thread. In the threads will be implement distinct objects.

VISUAL CREATION COMPUTING PROGRAM

One package and one platform in this paper will be discussed. These program give possibility edit

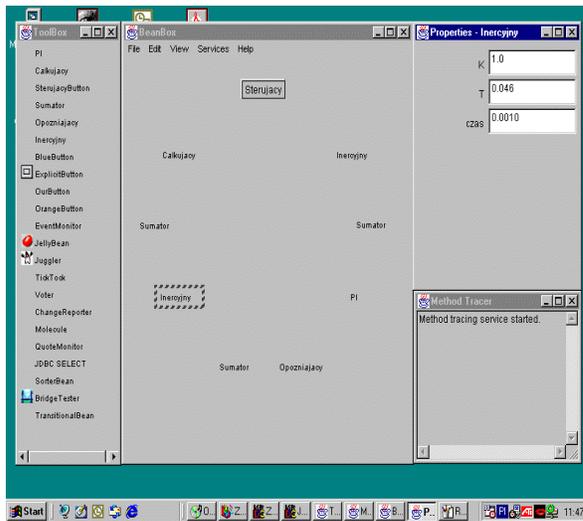


Fig.4. Properties-BeanBox window. Setting parameters of the Inertial beans.

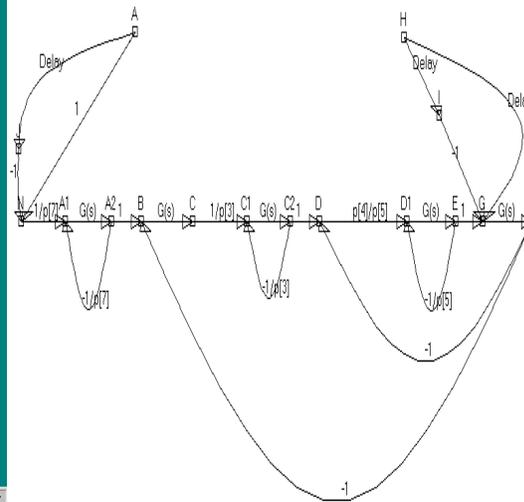


Fig.5. Diagram of direct speed control of dc motor presented in PASION package.

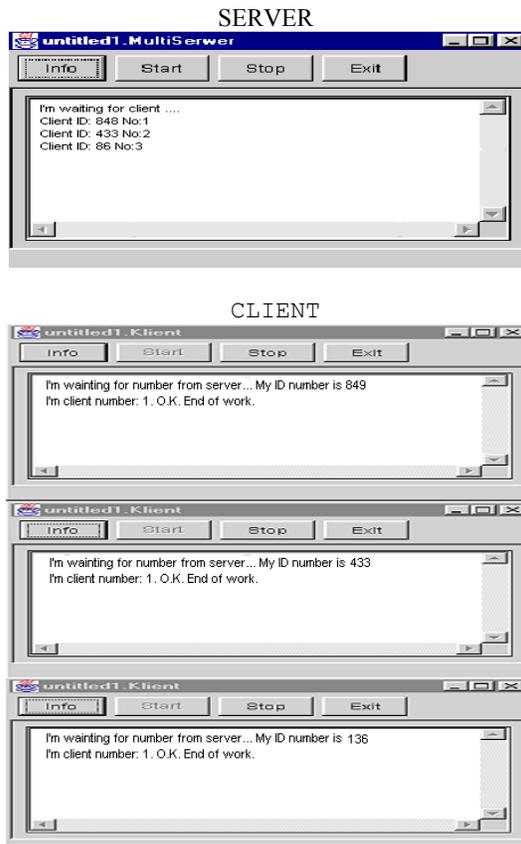


Fig. 6. Visualization of simulation based on multiserwer's working system.

with taking of Internet advantages complete prepositions shown on the fig.7 and fig.8. Fig.7 lets create the programs consisting of agent with simulate basic dynamic units. The dynamic units are

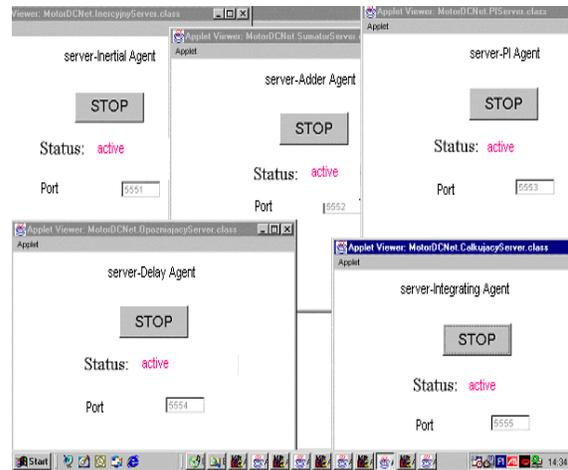


Fig.7 Active agents prepared for work on servers that simulate of typical control units.

located in separate servers. We became some control system if we appropriate connected these units. Fig.8 shows the complete control system on fig.1 which is simulated in computing net. These visualization permits the choose any address and any port of computer in net. On the screen we can set up the parameters of units fitted up in server.

VISUALIZATION OF RESULTS OF SIMULATED SYSTEM

Tool BDK give output only in numerical form. Result of presented form depends on programmer. In these situation output's forms depend on invention and ingeniousness of programmer.

The result can be presented directly on the screen which controls net (fig.8) or separate plots (fig.9).

On fig.9 have been shown results of using servers and RMI (Remote Method Invocation) to computing and simulating the direct speed control with the DC motor for different control systems. Thanks to these solution we can observe plots on the one screen for different motors.

Package PASION has very much possibility of presentation results of simulating calculates. It can present the results in numerical and graphic forms. The plots permits to preset many diagrams of speed in the same time both 2D plots (fig.10) and 3D plots (fig.11). We can observe 3D plots with different point of environment. PASION lets to present the state variables on the phase-plane (fig.12).

CONCLUSION

On the base carry out experiments regarding possibility visual creating programs and graphic elaborate result of simulation have been find out:

- thanks to visual method created the program we can leave out the difficult mathematical description,
- advantage of JavaBeans is the possibility of work in net, disadvantages are lack ready-made

dynamic units and a little suggestion visual create programs (for example in BDK),

- advantages of PASION package are ready-made dynamic units, analogue produce block diagram on the screen and very much possibilities graphic illustrate of output results; disadvantages is lack possibility net's programming,
- the propositions of visualization programs in computing net, its creating, handle ready-made programs and present output result fill to a certain degree a gap especially in parallel and distributed calculate.

BIBLIOGRAPHY

1. Raczynski S. 1998, "PASION for Windows 95/98". Package Manual.
2. Tolwinski R. 1999, "Simulation of DC Drive Systems with Use PASION Package". Modelling and Simulation: a Tool for the Next Millennium-13th European Simulation Multi-conference, Warsaw, Poland, June 1-4.
3. Tolwinski R 2002. "Simulation of Multiagent System in Parallel and Distributed Programming". 16th European Simulation Multiconference (ESM'2002 Darmstadt, Germany), June 3-5.

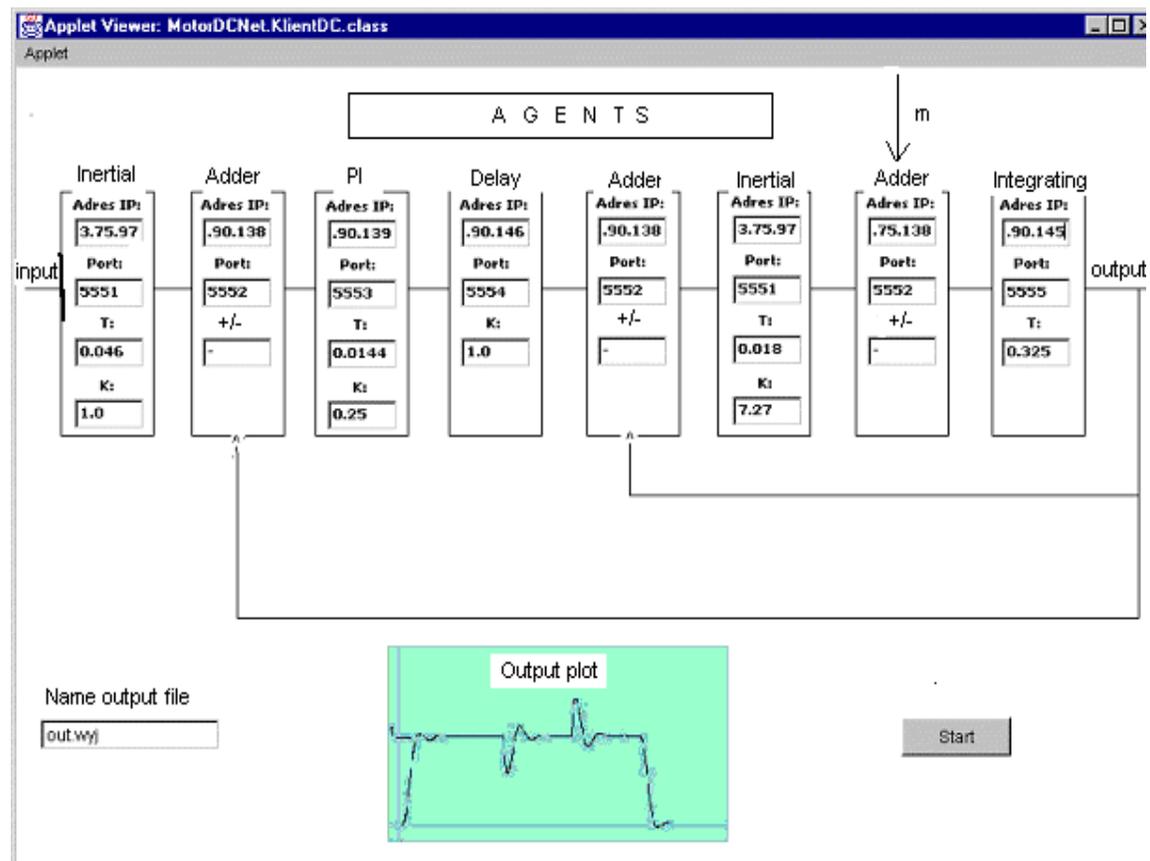


Fig.8. Control agent's screen that is the client of servers.

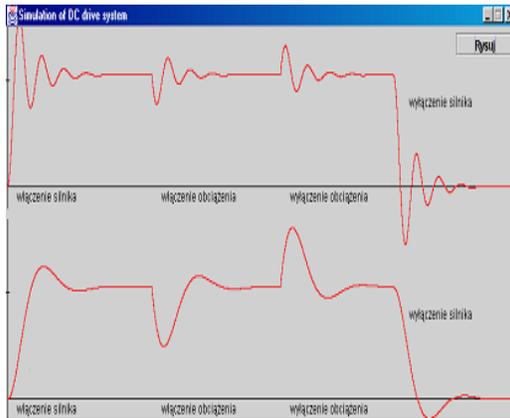


Fig.9 Diagrams of speed for the direct speed control with the DC motor.

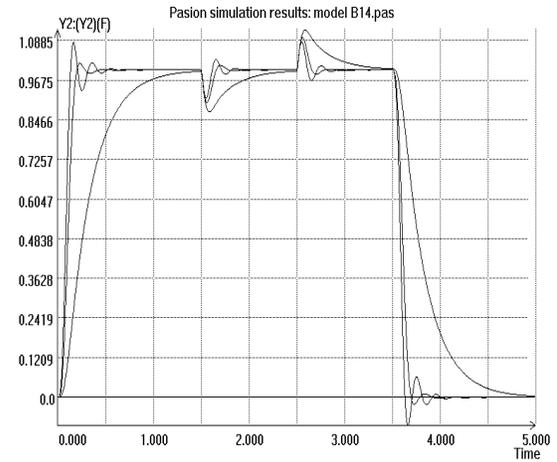


Fig.10. Diagram of speed for different parameters of control for different control system.

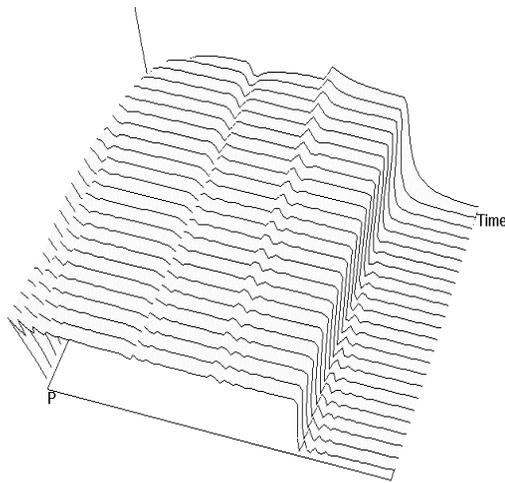


Fig.11 Diagrams of speed in 3D with the gain varying of controller.

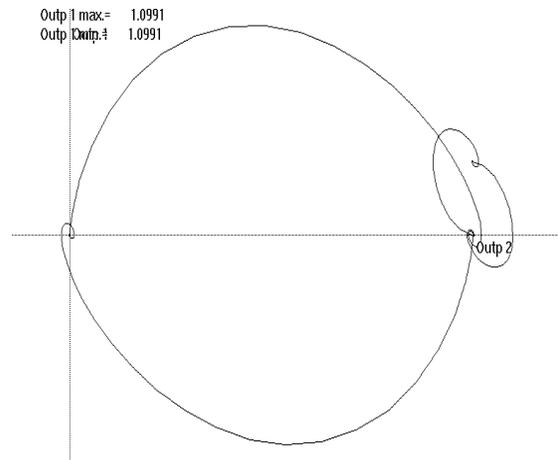


Fig.12 Oscillations of the speed and the armature's current on the phase plane.

TOWARDS COLLABORATIVE SIMULATION MODELLING: IMPROVING HUMAN-TO-HUMAN INTERACTION THROUGH GROUPWARE

SIMON J E TAYLOR

*Department of Information Systems and Computing
Brunel University
Uxbridge, Middx, UK.
simon.taylor@brunel.ac.uk*

STEWART ROBINSON

*Warwick Business School
Warwick University
Coventry, UK
stewart.robinson@warwick.ac.uk*

JOHN LADBROOK

*Dunton Engineering Centre
Laindon, Basildon
Essex, UK
jladbroom@ford.com*

Abstract: Collaborative simulation modelling, as defined by the GROUPSIM Network, involves the study of human-to-human interaction, computer-to-computer interaction, and synergies between the two, to support simulation modelling practices. This paper investigates the improvement of human-to-human interaction through the use of groupware. Interaction is introduced as C3, a combination of communication, coordination and collaboration. Simulation modelling introduces from the perspective of the roles that people take in a simulation study and the tasks that these roles must perform. The paper then presents results from an evaluation of NetMeeing groupware in the support of human-to-human collaboration. Several novel areas of future research are suggested.

Keywords: Simulation, Groupware, Collaborative Simulation

1. INTRODUCTION

Advances in distributed systems technology have created new possibilities for innovation in simulation modelling and the creation of new tools and facilities that could improve the productivity of simulation. Collaborative simulation modelling (CSM) is a term introduced by the GROUPSIM Network (www.groupsim.com) to refer to the many possible forms of human and computer collaboration that exist in simulation modelling.. Research topics focus on the support of human-to-human (H2H) interaction (computer supported cooperative work/groupware and simulation) and support of computer-to-computer (C2C) interaction (distributed simulation, parallel and distributed simulation, and web-based simulation), and the synergies between the two. Figure 1 shows the current overview diagram of CSM. As can be seen the diagram is developed on the basis of H2H interaction and C2C interaction through the simulation model. C3 is used to denote that the

interaction is made up of communication, coordination and collaboration activities. The distinction between the three activities is useful: we define communication as the exchange of information, coordination as the balanced and effective interaction of actions, and collaboration as the joint working with another or others on a shared project.

Looking to the future we see, in terms of H2H C3 distributed computing technology in the form of groupware facilitates interaction between simulationists (individuals and teams) and stakeholders (see the discussion on roles in section 2). For example, communication can be supported in various ways by using audio, visual, text, etc. over conventional or novel (wireless, PDA) technologies; shared diaries, shared activity planning, model version control and such like can facilitate coordination; and novel approaches to collaboration such as online application sharing can improve collaboration. Individual models are built and simulated in the same organisation, or multiple models are built

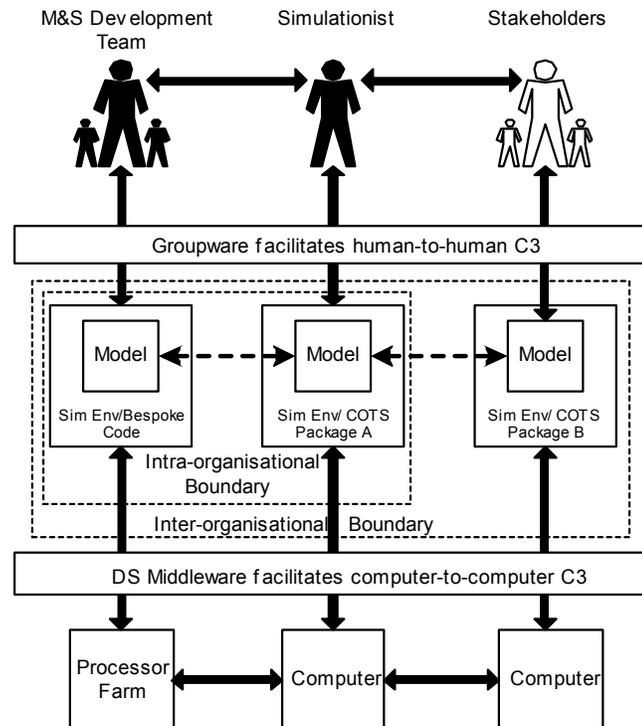


Figure 1: The GROUPSIM View of Collaborative Simulation Modelling.

that cross inter- and intra-organisational boundaries. Models can be software engineered in general purpose development environments or dedicated commercial-off-the-shelf (COTS) simulation packages. Distributed computing (in this context more commonly known as distributed simulation) middleware allows multiple models to interact over the Internet (or intranets) and to use processor farms to execute replications and experiments at high speed. The HLA-CSPI Forum (www.cspif.com) is dedicated to the development of C2C distributed simulation solutions that support C3 interaction between COTS simulation packages. Other aspects of this work are the focus of workshops that will take place this year (and will be reported in future publications available from the GROUPSIM website). In this paper we consider one facet of CSM, the results of introducing one type of groupware to simulation modellers.

The paper is structured as follows. In Section 2 we review C3 in simulation modelling and the roles and interactions that might be taken during a simulation study. Section 3 introduces groupware and one example, the net-conferencing tool NetMeeting. Section 4 presents some results of a survey and study of

the use of NetMeeting to support H2H interaction in simulation. Section 5 concludes the paper with some novel areas of research.

2. C3 IN SIMULATION MODELLING

To consider how one might support communication, coordination and collaboration in simulation modelling, it is useful to consider the general roles that people might take in a simulation study. Ormerod (2001) conveniently provides a useful characterisation in the definition of various groups in operational research interventions (amongst which simulation modelling is a key technique):

- *The doer*: in this case the simulation modeller
- *The done for*: the clients
- *The done with*: members of the simulation modelling team
- *The done to*: those from whom information and data are obtained
- *The done without*: those not involved, but nevertheless with a vested interest in the outcome

Table 1 shows this in the context of a simulation study. In other words, a person may take on

more than one role, or many people may be required to share a single role – the real world is not a tidy place (a modeller is often the project manager and the model user, in that he/she performs the experimentation). There may, however, be a number of people tasked with being data providers. A model user can be both a *done for* and a *doer* (a model user begins as a client and then becomes a doer as they use the model to provide information to the organisation).

The first three categories have direct involvement in the project team, while the latter two have little or no involvement. A wide group of people may need to be interviewed in order to obtain information about the system being modelled, but they do not need to have direct involvement in the simulation project. There may be a great many beneficiaries of the project, some of whom are even unaware of its existence. Workers in a factory are probably not aware that a simulation model has been developed to improve the level of production. They are, nevertheless, beneficiaries (or possibly a victims!).

2.1 C3 BETWEEN THE ROLES

The simulation modelling process can be described as a number of stages, as shown in figure 2. Four key stages are performed in an iterative manner: conceptual modelling, model coding, experimentation and implementation. In parallel with each of these are various verification and validation processes. The level of C3 required in a simulation study is now discussed by considering the process set out in

figure 1 in the context of Ormerod’s groups.

The nature and level of C3 between the simulation modellers and each of these groups will vary, and will depend upon the stage of the study that has been reached. Generally, the *doer* performs the simulation study with the *done for*. Where additional help is needed from subject matter experts and for supporting the modelling effort, the *done with* become involved. Interaction is also required with appropriate *done to* groups to gain relevant information and data. The *done without* are not involved (their role and their effect on the simulation study is outside the scope of this paper).

During the simulation study the frequency with which the groups interact is determined by the stage of the study. Consider a manufacturing system where a client wants to investigate the cost of manufacturing a new product with current production facilities. The client has enlisted a simulation modeller to help him or her make a decision (i.e. we assume a single modeller and not a team). To begin the simulation study the real world problem must be identified and a conceptual model of the system being studied must be built. In this case the problem is to evaluate the cost of production. A conceptual model is needed to identify what system elements (scope) and detail (depth) must be simulated to investigate the problem. Conceptual modelling is an intensive activity as the modeller must develop an understanding of the system being studied. The modeller and the client, as well as any appropriate information sources (i.e. personel involved in the production

Doers	Project manager	Responsible for managing the process; may not have specific modelling skills
	Modeller	Develops the model (conceptual and computer)
	Model user (in later stages)	Experiments with the model to obtain understanding and look for solutions to the real world problem
Done for	Clients	The problem owner and recipient of the results; directly or indirectly funds the work
	Model user (in early stages)	Recipient of the model
Done with	Data providers	Subject matter experts who are able to provide data and information for the project
	Modelling supporter	A third party expert (software vendor, consultant or in-house expert) provides software support and/or modelling expertise
Done to	Those interviewed for information	A wide group of people from whom information is obtained
Done without	Management, staff, customers	Beneficiaries of the project, but not involved; in some cases they are not aware of the project

Table 1: Roles in a Simulation Study

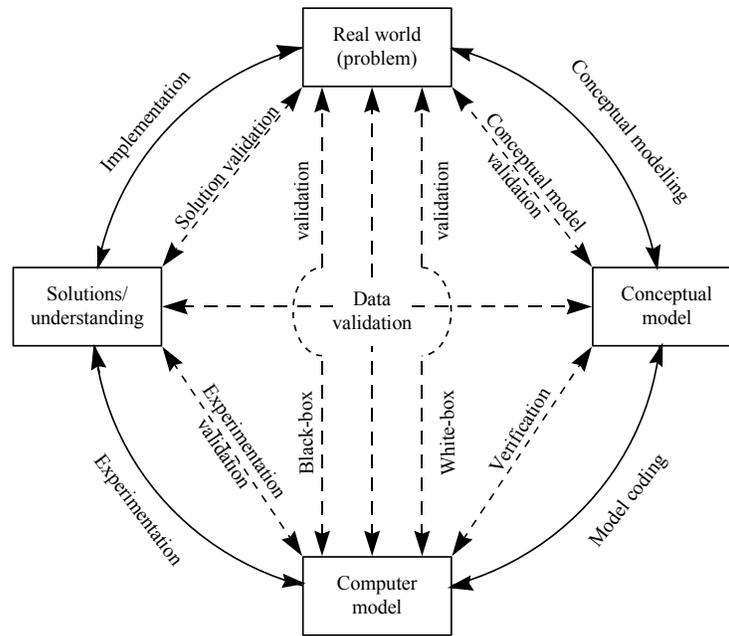


Figure 2: The Simulation Modelling Process (Robinson, 1999).

process), must therefore interact frequently so that the modeller can accomplish this. The *doer* must interact frequently with the *done for*, *done with* and the *done to*. Indeed, it is during conceptual modelling that the level of C3 needs to be at its highest.

In model coding the need for interaction is reduced. The modeller spends much time developing the computer model away from the eyes of the other parties. Verification is performed largely in isolation, since the modeller checks the model against the design stated within the conceptual model. That said, white-box validation (a detailed check of the computer model against the real world) is performed at regular stages during model coding, and so the model needs to be presented to the other parties for critique. The same is also true for black-box validation, which can only be performed once the model is believed to be complete. In terms of our study, the modeller would meet less frequently with the groups involved in the manufacturing system. The *doer* interacts moderately with the *done with* and the *done to*. Interaction with the *done for* is probably greater, since it is necessary to keep them apprised of progress.

Once the computer model is completed, the model user performs experiments with the clients to develop an understanding of how the complex relationships in the system being studied impact on the problem. In our case, experiments are performed with the computer

model of the manufacturing system to understand the probable cost of the new product. Significant interaction is required between the model user and the clients in order to share the understanding gained from the experimentation and to direct the continuing experimentation. It is expected that there will be much C3 between the *doer* (now the model user) and the *done for*. The *done with* and certainly the *done to* will be needed to a much lesser degree, although the need for help and information is not completely removed during experimentation.

The final stage (of a cycle) in the study is to implement the solutions and/or understanding that have been developed from the experimentation. Apart from fully explaining the results from the experimentation, the *doer* often has little involvement in implementation. That said, it is sometimes necessary to maintain the model or to provide results from further runs. C3 are often at their lowest at during the implementation stage.

Figure 3 summarises the discussion above, indicating the level of C3 at each stage in the simulation modelling process. It shows that there is a changing requirement for C3 as a simulation study progresses.

The volume of C3 required for successful simulation modelling add to the cost of performing a simulation study. This is further exacerbated if the groups involved in the study

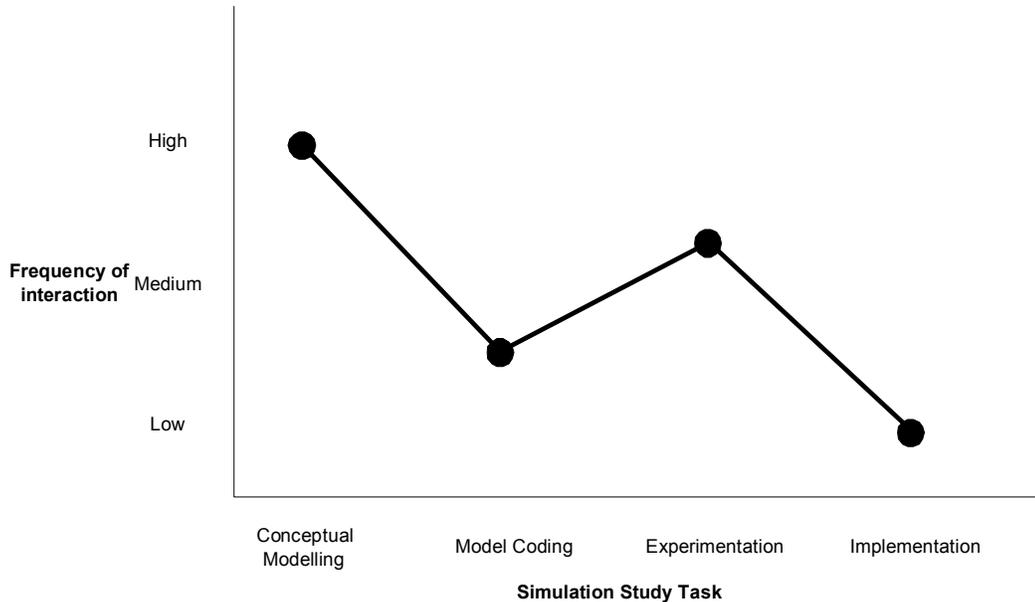


Figure 3: Frequency of interaction in the Simulation Modelling Process

are inconveniently or distantly located. In the next section we present a possible technological approach to reducing this cost.

3. GROUPWARE

The previous section highlighted the need for communication and collaboration in a simulation study. The field of Computer Supported Cooperative Work (CSCW) is a multi-disciplined research area that draws on expertise from both social and technical disciplines including distributed systems and internetworking, multimedia, communication, computer science and socio-organisational theory (Borghoff and Schlicter, 2000). Research in CSCW has led to Groupware, the practical application of CSCW research, a technology that pervades (often without the user knowing) many computing applications (for example IBM's Lotus Notes and Microsoft Office products have several examples of groupware). There are specific Groupware technologies to support specific tasks. These can be characterised as systems that support cooperative meetings or work as categorised on the basis of a simple two dimensional time/location matrix. The matrix divides groupware on the basis of time and location. During some task, people may meet at the same time, at different but predictable times (shift working on a project) or at different and unpredictable times (drop in team rooms). Similarly, people may meet in the same place (a room), in different but known locations

(different offices) or different and unpredictable locations (mobile workers). Note that many Groupware technologies support activities that fall simultaneously into many of the groups.

Conveniently Microsoft's NetMeeting is a groupware tool that combines various aspects of tele- and video-conferencing (only two users) with information sharing applications such as text chat, whiteboard, file transfer and application sharing. The product is reported (principally in Microsoft's press) as being used for applications such as remote training, collaborative design, augmenting existing software applications, virtual team support, accessibility, user support and many other situations where the emphasis is on reducing travel costs and saving time. NetMeeting works acceptably on a laptop connected to the internet via a normal modem (faster communications are preferable for ease of use). NetMeeting is accessed either through the Start menu, via a menu in a Microsoft Office application, or through Run by typing conf in the dialog box. The actual choice depends on the version of Microsoft Windows being used.

Text chat allows users to interact via a text conversation. The Whiteboard application allows users to draw various shapes on a shared drawing space (effectively shared Microsoft Paintbrush). Another application is File Transfer. This appears in a similar form to text chat; a menu of participants lists allows the user

to choose to transfer a file to another single participant or to the entire complement of participants. The final, and possibly most powerful feature of this package is the application sharing feature. This allows a participant in a NetMeeting session to share any application running on his or her computer. For example, a simulation package can be “shared” by selecting application sharing and selecting the simulation package from a list of running applications that NetMeeting can find on that participant’s computer. Once the package has been shared, all participants receive an image of the package as if it were running locally on their computer. Each participant can see the shared package and the results of any manipulation performed by the owner of the package. For example, the owner may communicate to the other participants (by text chat for example) that s/he is going to run the model to demonstrate how a part of the model works to the other participants. The owner runs the model as normal and the other participants will see the model animation as if the package were running on their own computer (with the caveat of communication speed). If one of the participants wanted to point out a model feature, or indeed stop the model and change some aspect of the model, the participant could request control from the owner. If control is granted, then all participants will see the mouse arrow annotated with the ID of the participant. The participant is then in *direct* control of the package running on the remote machine of the owner and may modify the model as they wish. See Taylor (2001) and Taylor *et al.*, (2002a) for more details on this technology.

4. EVALUATION

The approach taken for evaluation was in two stages. The first stage invited participants to take part in a “standard” demonstration of NetMeeting and then follow up with a questionnaire that invited participants to consider how potentially useful they might find this application in their role as a simulationist. The second stage was then to visit the participants two to three months later to see how (if any) adoption of the software was progressing. During the two to three month gap staff at Brunel University provided user support in the implementation of NetMeeting facilities at a participant site. Staff were restricted to the user support role. Care was taken to ensure that staff did not introduce new ideas and experience into the process – our objective was to examine the individual innovation made by a participant and not that given by shared experience.

4.1 Stage 1

In the first stage, the demonstration was given at nine different sites to approximately seventy subjects (one site involved a GROUPSIM workshop led by the Simulation Study Group of the UK Operational Research Society). Eleven returns were made from users in industry, defence, and academia. The results are presented here therefore as an indication rather than exhaustive evidence.

The demonstration took the form of an example collaboration between two users (the modeller doer and the system owner done for). A laptop with NetMeeting was connected via a standard modem to a global NetMeeting server. Each of the groupware features were demonstrated in turn with application sharing left for last. The application was loaded at Brunel University (UK) and local and remote interaction was demonstrated. The communication mechanism used was telephone (mobile) rather than the audio feature of NetMeeting. This was due to feedback when audio was placed on external speakers (necessary for the demonstration). The evaluation of audio was therefore on the basis of telephone (in two cases conference calls). The most unpredictable element of the exercise was making the modem connection as various methods were used each time to find a working phone point. The video was shown – the image was quite jerky and it was pointed out that this was smooth if a network connected to the Internet was used.

Each participant was asked to rate each of the demonstrated features of NetMeeting according to how potentially useful they found the feature on a scale of 1 to 5. Figure 4 shows the results from the evaluation. Ranks 1 to 5 indicates the perceived value of a feature with 1 indicating a low perceived value and 5 a high value. Relatively speaking, audio shows favourable results. Video performed moderately. Whiteboard performed well. Text chat performed poorly. File transfer also returned well. However, without doubt, application sharing performed the best and was considered an outstanding feature. Although the cross-section of the simulation modelling community was small, there is an indication that some aspects of this conferencing groupware are useful. Audio was (possibly obviously) useful to communicate with participants. There might be some confusion concerning the use of computer-based audio; most demonstrations used telephone/conference call rather than the application’s audio (which was feedback prone). Video was liked by some but was observed

several times to be a “novelty.” The information sharing applications were the most popular. The ability to conveniently document shared conversations via the text chat application was well liked. The whiteboard was also liked and, in several cases, it was observed to be a convenient “brainstorming” tool. The file transfer utility was found to be useful as it was considered helpful by some to transfer files to all participants by a click of a button rather than having to use email attachments. Application sharing, however, was evaluated as the outstanding feature of the groupware. Many different uses of this facility were discussed. All oriented around the ability for multiple users to take control of another’s application to demonstrate various points on-line.

industrialists were singled out as innovators in the use of NetMeeting in their simulation modelling activities. Overall, in terms of interaction with the various groups involved in a simulation project experience with the use of this tool has seen the augmentation of regular communication between the doers and the done for. Our returnees emphasised that this technology must not replace face-to-face meetings with remotely led net-conferences. However, since it appears that meetings can significantly contribute to the cost of a project, several modellers have commented on the use of net-conferencing to replace *some* meetings. Their innovative uses of NetMeeting are outline below.

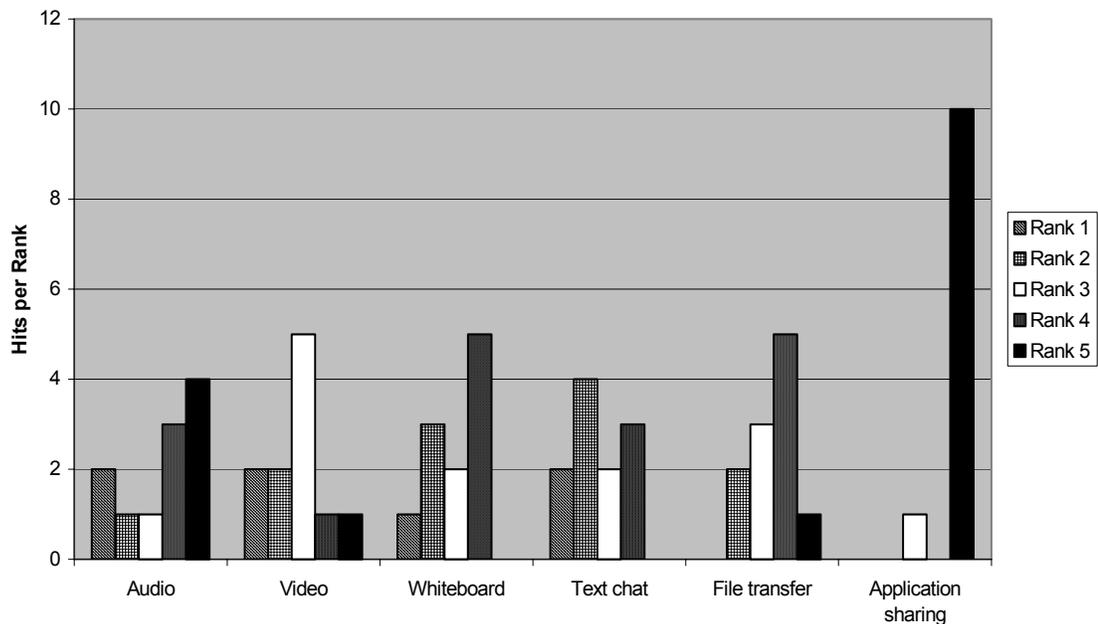


Figure 4: NetMeeting Features by Rank

4.2 Stage 2

This stage involved discussions on the use of NetMeeting with all returnees. The results were very black and white. Two to three months after the return the questionnaire, either the returnee did not use NetMeeting or they were now supporting some parts of the simulation modelling task. The only major difference between the users and non-users was the amount of modelling performed by returnee. In subsequent follow-up meetings to study the way in which NetMeeting was being used, three

Conceptual Modelling. As has been mentioned, in this activity the ‘doers’ require frequent and regular contact with the ‘done for’ and ‘done with’ in order to understand the nature of the problem situation, to define the modelling objectives and to define the conceptual model. In discussions specifically related to NetMeeting, the main application that has appeared is the use of the Whiteboard to collaboratively map out the boundaries and details of the conceptual model. In this situation, several computers have been networked in the same room, possibly with one being linked to a projected display. A

discussion takes place about the model, usually run by a facilitator, and participants draw appropriate diagrams sharing the whiteboard. This is a computerised version of a flip chart with the bonus of being able to import figures and diagrams and interact between parties in real time. No specific use of NetMeeting has been identified for the 'done to' (the providers of the data necessary for the development of the model).

Model Coding (especially white-box validation). In this application face-to-face meetings are required to discuss whether or not a model is being coded correctly (although less than in Conceptual Modelling). Typically the 'doer' demonstrates the model to the 'done for' and to the 'done with' to determine correctness and to promote belief in the model. Several modellers are now using the application sharing feature of NetMeeting to replace some of the meetings. This combined with a phone call (or conference call) allows the 'doers' to interact remotely with the 'done for' and to the 'done with' by allowing both parties to interact with the modelling software. In addition to this, the text chat feature has been used to "formally" document the agreement between parties that a change in the model coding has been agreed. This has been used to add to the model documentation.

Support Tasks. In addition to Conceptual Modelling and Model Coding (and Validation), NetMeeting has found use between the "doers" and an unexpected group of members of the "done with." These are the support teams found in large simulation groups and simulation vendors. There are some project costs that come as a result of the need to install new simulation software (or software tools), training to use the software, and support on tool use problems (rather than on Validation). The ability to share a simulation application through NetMeeting means that simulation software can be installed remotely (in one case across two continents), can be used to augment (not replace) existing training strategies, and can make support on tool use completely remote. This point was reinforced by the insistence of one returnee requiring that the support on their simulation software was performed through NetMeeting. This has resulted in NetMeeting being integrated in the vendor's support package and is now being rolled out to their customers.

5. CONCLUSIONS

This paper has introduced collaborative simulation modelling and has discussed how the

introduction of groupware can assist H2H C3. It has reported on a two stage evaluation of one groupware technology. The results of this work have shown that this technology has been used in three novel ways amongst the different roles. This technology is now being used by three companies to good effect.

In a wider context, this paper has shown that one part of collaborative simulation modelling research carried out by the GROUPSIM Network is of major interest to the community. Technology assisted C3 can save project costs – an important contribution as simulation modelling is a costly technique. What is of interest is the following research topics that require further study.

- techniques for effective communication, coordination and collaboration between the roles in a simulation study
- usability of groupware specifically within the above
- design of integrated COTS simulation package groupware tools
- the support of H2H C3 with C2C C3 distributed simulation modelling techniques.

We hope that this paper will engender further research into the support of simulation modelling through technology. For more examples on the use of NetMeeting, see Ladbrook and Januszczak (2001) for a study of how groupware has changed work practices in a multinational company and Taylor (2000) for more details on the use of NetMeeting. For an introduction to the issues of C2C C3 see www.cspif.com and Taylor *et al* (2002b) and Taylor (2002).

ACKNOWLEDGEMENTS

This work has been partially supported by the EPSRC GROUPSIM Network (GR/N/35304). We would like to thank Neil Bowerman (Nestle UK), Simon Dennis and Ray McKirdy (BTEExact) for valuable comments on this work.

REFERENCES

- Ormerod, R.J. (2001) Viewpoint: The Success and Failure of Methodologies – a Comment on Connell (2001): Evaluating Soft OR. *Journal of the Operational Research Society*, 52(10). 1176-1179.
- Robinson, S. (1999). Simulation Verification, Validation and Confidence: A Tutorial. *Transactions of the Society for Computer Simulation International*, 16(2). 63-69. 1999.

- Borghoff, U. M., and J. H. Schlicter (2000). Computer Supported Cooperative Work: Introduction to Distributed Applications. Springer-Verlag, Berlin, Heidelberg, Germany..
- Ladbrook, J. and A. Januszczak (2001). Fords Power Train Operations – Changing the Simulation Environment. In *The Proceedings of the 2001 Winter Simulation Conference* Washington, USA. Eds: B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, Association for Computing Machinery, New York, NY. 863-869. 2001.
- Taylor, S.J.E. Groupware and the Simulation Consultant. In *The Proceedings of the 2000 Winter Simulation Conference* Orlando Florida, USA. Eds: Joines, J. A., Barton, R. R., Kang, K., and Fishwick, Association for Computing Machinery, New York, NY. 83-89. 2000.
- Taylor, S.J.E. (2001). NETMEETING: A Tool for Collaborative Simulation Modelling. *International Journal of Simulation: Systems, Science & Technology*, 1(1-2), pp. 59-68.
- Taylor S.J.E. (2002) Interoperating COTS Simulation Modelling Packages: A Call for the Standardisation of Entity Representation in the High Level Architecture Object Model Template. In *Proceedings of the 2002 European Simulation Symposium*, Dresden, Germany. Society for Computer Simulation, San Diego, CA.
- Taylor, S.J.E, V. Hlupic, S. Robinson and J. Ladbrook (2002a). GROUPSIM: Investigating Issues in Collaborative Simulation Modelling. In *Proceedings of the UK ORS Simulation Study Group Two-Day Workshop*, Birmingham, UK. UK Operational Research Society, Birmingham, UK. pp. 11-18.
- Taylor, S.J.E., R. Sudra, T. Janahan, G. Tan, and Ladbrook, J. (2002b). GRIDS-SCS: An Infrastructure for Distributed Supply Chain Simulation. *SIMULATION*. 78(5), pp. 312-320.

COMPLEX SYSTEMS

MODELING OF THE KNOWLEDGE DYNAMICS OF STUDENTS OR EMPLOYEES

A P SVIRIDOV

*Moscow State Social University
107150 Moscow, Losinoostrowskaya str., 24
E-mail: sviridovap@smtp.ru*

Keywords: Statistical dynamics of knowledge, intensity of forgetting, knowledge flow, learning flow, open and closed model of knowledge dynamics, computer testing of knowledge.

Abstract. Statistical Theory of Teaching and Learning (STTL) [1-9]. includes statistical dynamics of knowledge [2,4,6-9], the methods of computer testing and diagnosis of knowledge (CTK and CDK) [1,4,5], the algorithms for supervised and unsupervised Learning of Teaching, Learning and Raiting Systems, metrological support of tutorial process. The principles of theory of statistical dynamics of knowledge and tutorial process control have been developed. They include the main ideas and characteristics of dynamics of knowledge in different subtopics, topics and educational disciplines, the intensity of learning and loss of knowledge, intensity of presentation of the educational material and knowledge restoration, the load on the person being trained and other related factors. The methods of the macro- and micro-models of learning and forgetting identification have been considered. On their basis the number of optimum strategies of the training quality's (anti- abnormal training) control, including the error filing when CTK are suggested.

The principles of CTK standardisation theory have been developed. They include the principal ideas, features and methods of analysis and synthesis of CTK plans, with use of quantitative, qualitative, alternative and linguistic sign with errors in identification of correct or wrong answers. The system of plans has been suggested. The system includes: 1) normal, intensified and short plans; 2) the rules of passage from one plan to another.

1. KNOWLEDGE DYNAMIC PARAMETERS OF THE DIALOGING SYSTEM

The numerical dynamic characteristics of knowledge are determined on the base of training diagrams and retentional curves, which is the

temporal dependence of complex index of training quality: 1) the reaction time, 2) the probability how is the work done right or wrong by trainer/operator....

Training diagrams and retention curves can be analyzed on macro- and micro-level [2,4,5,9]. In first case we have got usual exponential macromodels, the dialogue system is made for it's identification.

The definition example of intensity of forgettable λ exponential models decision of the task on the program language: Turbo-PASCAL – from 0,3 to 1,535; C - from 0,8 to 2,07 (per year), the mean time of knowledge keeping by Turbo-PASCAL is more than by C.

Task to solve: 1) Give an example of definition program of minimum element one-dimension massive; 2) Make the procedure of parallelepiped graphic representation.

Transformation micro-models of one structure, as a rule are nonlinear, nonmonotonous, as a rule includes local maximums and minimums and plateau, reflecting evolutional places of perfection some strategies and transition from them to more perfect one. For it's identification it is necessary to use more powerful methods of quantum physics than the theory of probability. There is a new conception among them - complex mark wave function and amplitude of probability. The condition of knowledge "a" and transition to condition "b" we can compare amplitudes of probability $A(a)$ and $A(a,b)$ – complex numbers, module square is probability of condition "a" and transition from this to b: $P(a) = |A(a)|^2$, $P(a,b) = |A(a,b)|^2$. The use of quantum-physics approach will be shown on the process of forgetting. Compare to the process of forgetting with exponential macro-model $Q(t) = Q_0 \exp(-\lambda t)$,

where

Q_0 - probability to solve right the task in the moment of time $t=0$,

λ - intensity of forgetting,

$A_1(t)$ and $A_2(t)$ - two amplitudes of probability in aspect:

$$A_1(t) = (0,5 Q_0 \exp(-\lambda t))^{1/2} (\sin \omega_1 t + j \sin \omega_2 t)$$

and

$$A_2(t) = (0,5 Q_0 \exp(-\lambda t))^{1/2} (\cos \omega_1 t + j \cos \omega_2 t)$$

The resulting amplitude of probability $A(t)$ in the moment of time t is defined by sum

$$A(t) = A_1(t) + A_2(t) \quad \text{or}$$

$$A(t) = (0,5 Q_0 \exp(-\lambda t))^{1/2} (\sin \omega_1 t + \cos \omega_1 t + j(\sin \omega_2 t + \cos \omega_2 t)).$$

So the probability of solving right of task in the moment of time t is: $Q_1(t) = |A(t)|^2 = 0,5 Q_0 \exp(-\lambda t) (2 + \sin 2\omega_1 t + \sin 2\omega_2 t)$.

Example:

For macro-model of the forgetting process by informative-measuring technique $Q(t) = 0,745 \exp(-0,018t)$ (t per month) [2,4,9] we have got a micro-model $Q_1(t) = 0,373 \exp(-0,018t) (2 + \sin 220t + \sin 499,1t)$.

With meaning ω_1 and ω_2 in accordance with the frequency

$$f_1 = 0,67 * 10^{-5} \text{ Hz}, \quad f_2 = 1,53 * 10^{-5} \text{ Hz}.$$

Such a way of micro-models building can be used in training modeling, developing, dynamics of difficult systems, scientific technical and social progress, but wavelike processes and characteristic for them too. That is why we use wider interpretation of “learning”.

2. MODELING AND PROGNOSTICATION

For the description knowledge dynamic models the main ideas are used: studying material flow (knowledge flow) L and flow of comprehensibility/rehabilitation of knowledge (learning flow) A . They are determined in the way of putting conforming flows L_i and A_i ($i \in (1, \dots, N)$) for the separate tasks with their life circles (LC) S_i :

$$L = L_1 + L_2 + \dots + L_N, \quad A = A_1 + A_2 + \dots + A_N$$

(knowledge and learning flows).

Life cycle (LC) S_i represents a time interval, during which a situation i (or standard activity units SAU)) assimilation is necessary. Subjects flow L_{1i}

is formed by time moments $t_0^{1i}, t_1^{1i}, t_2^{1i}, \dots$, where $t_{k+1}^{1i} > t_k^{1i}$ ($k \geq 0$). A time moment t_0^{1i} corresponds to the moment of the first presentation of the subject in a current situation (SAU) according to the program, and the following time moment $t_1^{1i}, t_2^{1i}, \dots$, correspond to the forgetting moments.

Knowledge assimilation (recovery) flow A_{2i} , attached to situation i (SAU) is formed by time moments $t_0^{2i}, t_1^{2i}, t_2^{2i}, \dots$, where $t_{k+1}^{2i} > t_k^{2i}$ ($k \geq 0$). Time moment t_0^{2i} corresponds to assimilation moment of situation i (SAU) after its first presentation, and the following time moments of knowledge recovery after their forgetting.

The analogical way is for gave and done tasks in the other cases of human activity [2,4,9]. All these permits to use dynamic knowledge models in modeling their human activities.

The knowledge dynamic is determined:

1. By the flow of studying material (knowledge flow) X ;
2. By the flow of comprehensibility/treatment of knowledge (learning flow) Y ;
3. By the number of N task/function in the discipline, which must be done by the student/operator;
4. By the way of learning and treatment of knowledge r ;
5. By the number of s channels of knowledge learning and treatment (for example: visual and acoustic channels);
6. The capacity buffer memory m .

The open knowledge models of dynamic are described by 6 symbols $X/Y/N/r/s/m$ (wider symbolism of Kendall). Closed models must be in brackets. There can be M, G, GI, D, E_k insert X and Y . So, $Y=M$ means that time of learning is free and has exponential assessment.

Let's see net dynamic models of knowledge $\langle M/M/N/2/1/(N-1) \rangle$ and $\langle G/M/N/2/1/(N-1) \rangle$. They are identical of closed one channel stage learning/treatment with N source, which conforms to N task of the specialties (learning disciplines)

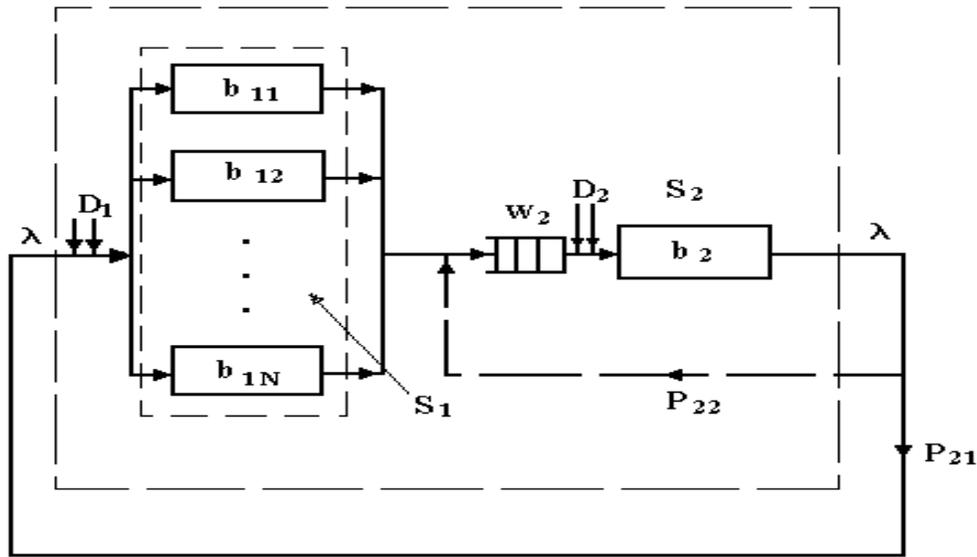


Fig.1. Closed model of Knowledge dynamic

for the trainer, (N-1) place of waiting and to the stage of learning without loosing. The discipline of learning - "First Come First Served" (FCFS). The dynamic model of knowledge (Fig.1) consists of stages of forgetting S_1 and learning S_2 , and learning could be done in the time-sharing regime. In this case for learning or reanimation of knowledge in some field of task distinguished time quantum. If this time quantum is enough for learning, than the task with probability p_{21} comes to the stage of forgetting S_1 . On the contrary task with the probability p_{22} comes to the stage of learning S_2 . Let as mark:

λ - the mean intensity of task forgetting,
 μ - the mean intensity of task learning,
 b_{1i} ($i=1,2,\dots,N$) - the mean time of forgetting time waiting of i position/task/function, $b_2=1/\mu$ - the mean time of learning/reanimation of knowledge by condition/task/function,
 D_1 and D_2 - disciplines of forgetting and learning,
 w_2 - the buffer capacity for forgetting condition/task/function before the learning stage S_2 .
 There are more learning channels (visual...) used in the learning process by studying (multimedia) courses, the intensity of learning is higher and intensity of forgetting is lower.

The main characteristics of this model:

The probability that the student is able to

$$p_0 = \left[\sum_{n=0}^N \frac{N!}{(N-n)!} (\lambda / \mu)^n \right]^{-1}$$

solve all N tasks

The average number of tasks waiting time in the system (knowledge reanimation)

$$L_q = N - \frac{\lambda}{\mu} (1 - p_0)$$

The average waiting time in the waiting line

$$w_q = \frac{L_q}{\lambda (N - L)}$$

The average number of forgetting tasks

$$L = L_q + (1 - p_0)$$

The average waiting time in the system (reanimation)

$$w = w_q + 1/\mu$$

The probability of forgetting n tasks

$$p_n = \frac{N!}{(N-n)!} \left(\frac{\lambda}{\mu} \right)^n p_0, \quad n = 0, 1, \dots, N$$

For the programming language C we have got: $N=51$, $\lambda=1,05$ [1/per year]= $1,22 \cdot 10^{-4}$ [1/per hour], $\mu=1$, $p_0 \approx 0,994$, $p_1 \approx 0,006$, $L_q \approx 1,56$, $w_q \approx 260$ hour, $L \approx 1,566$, $w \approx 261$ hour.

Mounted: optimum sequence of setting relative priorities in reanimation of forgetting tasks/functions are determined by relation c_i/b_{2i} , where c_i - is the fine/waste for the lack of knowledge i position/tasks/functions (for disability of making i tasks or functions). Close net dynamic models of knowledge might be used as mathematical model in case of task solving

ensuring **guaranteed quality** of professional training for N tasks solved or **guaranteed quality** of student's training for the final number of training disciplines.

3. OPTIMIZATION OF QUALITY MANAGEMENT

The forgetting intensity might be use for optimal planing of lerninig material planing (LMP) [2,4,9]. A number of optimal strategies of preparation quality control are developed using:

- intensity of assimilation and forgetting;
- economic factors (education and control costs, losses due to the error actions, possible benefits, etc.);
- under-estimate or over-estimate risks on the CTk.

Let us analyse two strategies:

Strictly periodical organization strategy of LMP.

Taking into account knowledge under-estimate (α) and over-estimate (β) risks and correctness of exponential time distribution, a learner (operator) professional availability ratio is determined as follows [4,9]:

$$K(a) = \frac{(1-\beta)(1-\exp(-\lambda a))}{\lambda(a+t_k)(1-\beta \exp(-\lambda a)) + t_a(1-\beta)[1-(1-\alpha)\exp(-\lambda a)]}$$

where λ - intensity of forgetting, t_a - average time of knowledge recovery, t_k - average time of knowledge control, and a - interval between knowledge tests and repetitions (recovery). For an operator who has $\lambda=0,1$ month⁻¹, $t_k=0,5$ h, $t_a=3$ h, $\alpha=0,35$ and $\beta=0,4$, an optimal interval between tests and anti-failure trains will be: $a_0=110$ h. Note that maximum value of $K(a_0)$ is 0,97.

For trainers we have got: the intensity of forgetting $\lambda_1=0,05$ и $\lambda_2=0,40$ 1/month, average time of knowledge reanimation $t_e=3$ hour, average time of knowledge control $t_k=0,5$ hour, the underrate or overrate risk on the computer testing of knowledge (CTK, computer-based test) $\alpha=0,35$ and $\beta=0,40$. Optimal time intervals between repeating (anti damage training), guaranteeing the maximum coefficient of professional training, in case of periodical management strategy of training for 1 and 2 trainers/operators is $T_1=160$ hour, $T_2=50$ hour. In case of LMP (anti damage training) using through the intervals, the coefficient of workers professional training is lower.

Strictly periodical strategy of LMP with two checking types.

$N=(1,2,\dots,n)$ – multitude position of tasks, functions which must be learn by students/users. They should pass tests for professional training guarantee complete and part knowledge checking are used with further knowledge maintaining on forgetting ideas and tasks. If the test is global - whole knowledge is controlled and if the test is partial just part of knowledge is controlled ($A \subset N$).

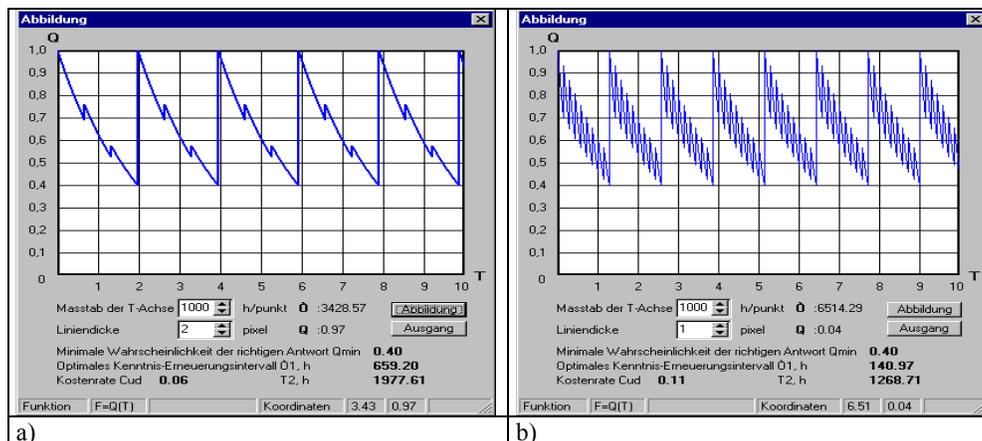


Fig 2. The probability dependence of correct tasks by students/users.

λ_1 and λ_2 – are the forgetting intensity of tasks A and N\A, c_1 and c_2 ($c_2 > c_1$) – the expenditures for test organization and reanimation of forgetting

knowledge A and N, a_1 and a_2 – intervals between global and partial tests, in this case $a_2 = r a_1$ ($r > 1$). The optimization problem is definition of meaning

a_{10} and a_{20} , the average expenditures are minimum, and probability of correct tasks $Q(a_2-0)$ is not below of some Q_0 . On the Fig.2 are dependences $Q(t)$ for $Q_0=0,4$, $c_1=5$, $c_2=100$: a) $\lambda_1=0,000139$ [1/hour], $\lambda_2=0,000417$ [1/hour]; b) $\lambda_1=0,002$ [1/hour], $\lambda_2=0,0005$ [1/hour]. The answer is: a) $a_{10}=659,20$ hour, $a_{20}=1977,61$ hour; b) $a_{10}=140,97$ hour, $a_{20}=1268,71$ hour.

The strategies have realization in dialogue system. These models might be used in integration of business and knowledge management.

4. STANDARDIZED METHODS OF CTK

Statistic tool CTK is a plan determined by a number of tasks/questions that taken from (potential) general combination for the learner, by volumes of the tasks and by conditions of giving different grades. Such plans are widely used for control and preparation quality certification by Universities and companies.

Main characteristics of CTK plans

1. Operating characteristics OC,
2. Risks of underestimation and overestimation of knowledge α and β (error of the first kind, α -error, error of the second kind, β -error),
3. Realization of psychometric function of teacher,
4. ASN=average sample number,...

Synthesis of equivalent and ε -equivalent single sampling plan inspection of computer-based knowledge test while understanding the reality of answers with and without errors (for example in choosing way of entering the answers=Multiple-Choice-Input) can be done: 1) by two dots OC (P_1 , $1-\alpha$) and (P_2 , β), where P_1 - max. part of misunderstood questions when "credit" is given, P_2 - min. part of misunderstood questions when "no credit" is given, 2) by dot of indifferent OC (IQL=indifferent quality level) and curve in it (plans of ($P_{0,5},h$) kind); 3) by meaning P_1 and α or P_2 and β ; 4) by giving psychometric function of teacher/expert; 5) by economic data. It is better to use Larson-nomogram for graphic synthesis.

Example.

Given: $P_1=0,2$, $\alpha=0,3$, $P_2=0,3$, $\beta=0,35$. On the basis of Larson-nomogram of the cumulative binomial distribution we have: $n=12$, $c=3$, i.e. learner is offered 12 questions (tasks) "credit" is given if max. number of errors is not more than 3 (single sampling plan (12,3)).

Let it use choosing way of entering with average number of answers for choosing $S=4$, than on the

basis of Larson nomogram we have: $n=18$, $c=3$ (plan (18,3)). Plans (12,3) and (18,3) are equivalent from the point of view of risks of underestimation and overestimation of knowledge.

Plans of knowledge control composition

Let's enter the following levels of control: 1) task, 2) section (for example lab work), 3) theme (part of activity), 4) learning discipline (activity), 5) preparation quality on specialty in University, 6) organization/institute, 7) (higher) education. Correlation between elements of different levels of correction - "consists of". Number of level of correction - **difficulty degree of control system, diagnostics and certification**. Let us see the way of determination of OC composition of control plans of 2 neighbor levels.

Let the correction on one level be done according to the plan with OC $L_1^*(Q)$. Such level can be: a) knowledge control of separate student on different sectors of learning discipline, b) knowledge control of students for the whole learning discipline or specialty. On the next level the decision is being made according to the second plan with OC $L_2^*(Q)$, i.e.:

- a) about understanding of the whole learning discipline by the student,
 - b) about the preparation quality on specialty or learning discipline in organization/university.
- The result OC of control plan on the second level looks like:

$$L_{res}^*(Q)=L_2^*[L_1^*(Q)].$$

Example.

For two-mark CCK of students of some specialty is used one step plan (10,2). The conditions of preparation quality certification on specialty on the results of correcting separate students according to the plan (10,2): $P_1=0,1$, underestimation risk $\alpha=0,2$; $P_2=0,3$, overestimation risk $\beta=0,35$. It is necessary to synthesize one-step plan of preparation quality certification on specialty in the university.

Solution.

We have:

$$P_1^*=0,1*0,65+0,9*0,2=0,245,$$

$$P_2^*=0,3*0,65+0,7*0,2=0,335.$$

On the basis of Larson-nomogram we get: $n=45$, $c=11$. This means: one should choose 45 students from general combination and correct them according to the plan (10,2). If the number of unprepared students/operators is not more than 11, than the preparation quality on the specialty in organization/university is positive.

Ways of analysis and synthesis of computer-based knowledge-tests for various tests and examination layers (especially for individual students,

employees and courses of studies) are realized in dialog system.

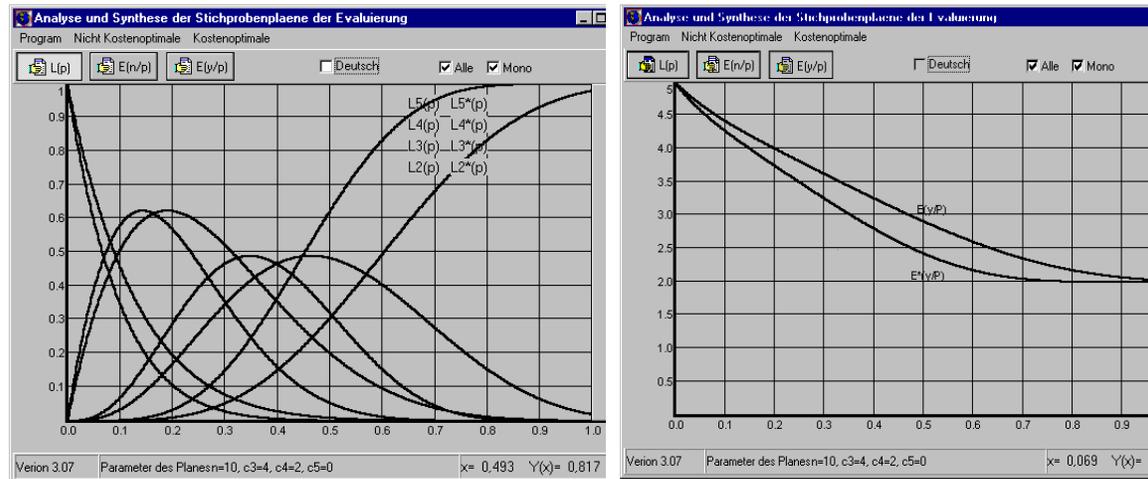


Fig. 3: a - OC $L_i^*(P)$ and $L_i(P)$, $i=2,3,4,5$; b - realization $E^*(y/P)$ and $E(y/P)$ psychometric function of teacher/expert

Let us see the examples of its using.

On Fig. 3a and 3b are shown:

- $L_i^*(P)$, $i=2,3,4,5$ - OC of one step plan $(10,0,2,4)^a$ of four mark CCK without mistakes in determining the reality of the answer (writing or oral answers),
- $L_i(P)$, $i=2,3,4,5$ - OC of the same plan $(10,0,2,4)^a$ in choosing way of entering the answers, when approximately 4 possible answers are given for each question; these OC are moved right to OC $L_i^*(P)$;
- $E^*(y/P)$ - realization of psychometric function of teacher/expert (dependence of average mark y on possibility of wrong answer P) with the help of intellectual learning or supportive system that realizes plan $(10,0,2,4)^a$ according to writing or oral answers (low curve),
- $E(y/P)$ - realization of psychometric function of teacher/expert with the help of intellectual learning or supportive system of the same plan $(10,0,2,4)^a$ with choosing the way of entering the answers when approximately 4 possible answers are given for each question (high curve).

REFERENCES

- [1] Sviridov A.P. - „Introduction to the Statistical Theory on Teaching and Learning. Part I. Sampling Plans for Computer-Based Knowledge Tests“. (Wwedenie w statisticeskiju teoriju obucenija. Cast 1. Standartisirowannye metody kontrolja snanij). - Moscow: Published by "Moscow Power Energetics Institute", 1974.- 134 P.
- [2] Sviridov A.P. - „Introduction to the Statistical Theory on Teaching and Learning. Part II.

- Statistical Knowledge Dynamics“. (Wwedenie w statisticeskiju teoriju obucenija. Cast 2. Elementy statisticeskoj dinamiki snanij). - Moscow: Published by "Moscow Power Energetics Institute", 1974.- 152 P.
- [3] Sviridov A.P. - „Supervised and Unsupervised Learning of Teaching, Learning and Rating Systems“. (Obucenie i samoobucenie obucajuschich i kontrolirujuschich maschin). - Moscow: Published by "Moscow Power Energetics Institute", 1976.- 182 P.
- [4] Sviridov A.P. - „Fundamentals of the Statistical Theory on Teaching and Learning“. (Wwedenie w statisticeskiju teoriju obucenija i kontrolja snanij). - Moscow: Published by "Wysshchaja schkola" (Higher school), 1981.- 262 P.
- [5] Swiridow A.P. - „Statistische Lehrtheorie und Humanisierung von computerunterstuetzten Lehrsystemen“. -IBM-Hochschul-Kongress. Dresden 30.9-2.10.92. Proceedings.
- [6] A. Swiridow, I. Schalobina. - „Kenntnis-Dynamik ueber ein Lehrfach“. -Ilmenau: 42. Internationales Wiss. Kolloquium an der TU Ilmenau, 1997, Bd.1, pp. 401-406
- [7] Swiridow, D. Slesarew, I. Schalobina. „Simulation von Kommunikationssystemen und -netzen als sozio-technische Systeme“. In: 43. Intern. Wiss. Koll. An der TU Ilmenau.-Ilmenau: TU Ilmenau, Bd.1, 1998.- pp. 174-180
- [8] A. Swiridow, D. Reschke, N. Slesarewa. - „Modellierung der Kenntnis-Dynamik“. In: 43. Intern. Wiss. Koll. an der TU Ilmenau.-Ilmenau: TU Ilmenau, Bd.1, 1998.- pp. 282-189
- [9] R. Suesse, A. Swiridow. - „Statistical Knowledge Dynamics“. (Statistische Kenntnis-Dynamik).-Ilmenau: Wissenschaftsverlag, 1998.- 256 P.

INTEGRATED RESOURCE SCHEDULING AND SIMULATION FOR DYNAMIC LOGISTIC MANAGEMENT

ROBERTO MOSCA, AGOSTINO BRUZZONE, ALESSANDRA ORSONI

*University of Genoa - Department of Production Engineering
Via Opera Pia 15, 16145 Genoa, ITALY
e-mail: {agostino, aorsoni}@itim.unige.it*

Abstract: Resource scheduling is a critical step in the management of complex logistic networks. The paper proposes an integrated scheduling and simulation approach for dynamic resource allocation in complex transportation logistics. An example application, specific to the maritime logistics of the chemical supply chain, is discussed in the paper, along with preliminary testing of the system by industrial users.

Keywords: resource scheduling, transportation logistics, dynamic logistic management, chemical supply chain

INTRODUCTION

Complex logistic networks servicing distributed production environments require efficient and flexible resource scheduling in order to meet the dynamic needs of production. Costs and risks of poor resource scheduling rise as the size and the geographical distribution of the supply chain are increased. The paper describes an integrated scheduling and simulation system for dynamic resource allocation in the maritime logistics of distributed chemical processing. In this context, resource scheduling refers to the allocation of commercial vessels types and sizes to a multiplicity of product transportation requirements subject to the stochastic variability of calendar constraints. Such constraints are concurrently determined by variable product pick-up/delivery dates, vessel availability and current location, equipment availability and set-up times at different docking facilities and port infrastructures. Simulation provides the context for scenario customization and testing of the logistic solution as interfaced to the production network. In particular the simulation module tests the feasibility of each scheduling solution and provides quantitative measures of its performance, intended as combined logistic and production performance for the specified industrial context. In the assessment of each scheduled plan the system accounts for weather conditions, influencing ships navigation times, for congestion and failures at each facility, affecting port operations times, and for variable production rates, which impact product stock and available storage capacity at each processing site. For these purposes, a dedicated database receives hourly updates on the status of the dynamic scheduling parameters such as, plant production rates, storage levels, and ships locations, directly from the operative information systems. Some of these parameters are collected on-line and in real-time (i.e. Estimated Time of Arrival – ETA – directly provided by all the ships currently

operating in the network, their position through the Geographic Positioning System –GPS–, and Storage Level in each Port/Plant’s Reservoir.) others are extracted from the transitional informative and management systems (i.e. calendar changes in the availability of resources and infrastructures). The database is structured to ensure that different users may create their own scenarios modifying the detailed parameter settings (i.e. capacity of pipeline Z of plant X), without introducing changes in the reference data for operative scheduling. A hierarchical user authorization procedure ensures the consistency of the baseline scenario which is the current/actual reference for operative planning an scheduling. The system has the major advantage of displaying within a single application the entire set of information required to complete the designated scheduling tasks. Scheduling solutions can be obtained which are fully compatible with the entire range of technical, logical, and regulatory constraints characterizing the actual transportation network.

SCHEDULING MODULE

Operative scheduling of the entire transportation network requires large sets of data including the full list of product flows ranked by priority and complete with all the relevant information such as, ports of origin and destination, product, quantity, loading/unloading calendar slot, estimated unloading/loading slot, and estimated navigation time. The second set of information refers to the port characteristics, for instance number of docking facilities, saturation index and possible interference/overlap for each flow included in the list. Finally, a list of ships compatible with the selected flows is required along with all the relevant ship information such as name, size, type of contract, saturation, current position, last product delivered and ETA).

The priority index employed for ship mission ranking purposes can be calculated as a function of:

- ◆ Product flows managed by the ship mission
- ◆ Cost of the ship allocated to the mission
- ◆ Penalties associated to the given ship/contract
- ◆ Proximity index for the next loading/unloading calendar slot (difference between the beginning of the slot and the current date)

The proximity index allocates scheduling priority to the nearest ship missions in time, assuming that the cumulative effects of the stochastic phenomena concurring to determine the ETA of later ships on later missions will concurrently contribute to facilitating their fitting a feasible schedule. Equation 1 is used to determine the priority index.

$$\pi_i = \eta_{Fl} \sum_{j=1}^{j=PM} F_{ij} + \eta_S C_S + \eta_{Pn} C_{Pn} + \eta_{Pr} (d_{start} - d_{now} - \gamma_{gate}) \quad (1)$$

In the equation

- π_i = Priority of i-th ship mission/order
- F_{ij} = j-th flow of mission i-th
- η_{Fl} = Flow's weight coefficient on that ship
- C_S = Cost of ship hire
- η_S = weigh of ship hire
- C_{Pn} = ship penalties by contract
- η_{Pn} = weight of penalties
- d_{start} = calendar slot start date
- d_{now} = current date
- γ_{gate} = slot duration in days
- η_{Pr} = weight of slot proximity

Each one of the physical objects, namely resources, involved in the scheduling process has an associated calendar where busy/available times are recorded. Resources are allocated to ship missions according to their priority ranking and accounting for all the applicable constraints, these include:

accessibility constraints: requiring for instance ship/dock compatibility in terms of geometric parameters such as ship's length, width and deadweight.

product compatibility: requiring specialized procedures between product unloading and any new product loading if the two types of products are classified as non compatible for storage/transportation.

temporal interference: related to the possible overlap of the loading/unloading calendar slots for different ships operating in the same port/dock.

production sustainability: concurrently determined by plants storage capacities and production rates:

sustainability is an indicator of the number of days that production can be carried out independent of a particular ship's arrival.

resource constraints: determined by the simultaneous need to employ the same resource for different tasks such as, more than one ship per dock, multiple loading/unloading tasks per pump/pipeline/equipment, multiple connection requirements for a pipeline segment enabling the connection to different reservoirs.

When the application is run in the automatic mode, if a conflict and/or violation of the constraints occurs, the scheduling problem is flagged out to the user and possible solutions are suggested based uniquely on cost effectiveness considerations. However, more experienced users may have reasons to force some of the constraints, knowing for instance that the deadweight of a particular ship is compatible with the accessibility requirements of a given dock if the ship is carrying half or less of its maximum capacity: therefore when run in semi-automatic mode, the system allows for user intervention in forcing some of the pre-set constraints. The allocation of a ship to a given product flow grouping automatically changes the saturation levels of both ship and docking facilities. Color coding is used in the interactive operation mode to display the saturation level of each resource: green (less than 50%), yellow (up to 75%) and red (more than 75%). By clicking on any of such indicators the program displays a multilevel Gantt chart for each object (ship/facility) where all the busy time slots, relevant to the scheduling horizon, are recorded. The different objects determining the occupation state of each resource/facility are displayed on the Gantt Chart and may be moved by the user causing the recalculation of the entire set of parameters for constraint verification/satisfaction purposes. Constraint and interference verification enables the identification of potential conflicts, however the interactive scheduling mode accepts the definition of highly incompatible ship missions. The stochastic variability associated to each component/operation, in fact, can often create conflicts even within perfectly timed missions, therefore the identification of conflicts does not have blocking consequences in the scheduling process nor does it force the user to make immediate changes to rectify the situation. Temporal overlaps are pointed out by the system and recorded in the calendars of the relevant objects where each occupied slot is specified in terms of start/end dates, designated user and purpose.

Conflict management, as in the case of overlapping calendar slots for two or more ships with respect to a same docking facility, is entirely handled assessing the cost implications of each alternative solution at the level of the entire scheduled plan (i.e. accounting for secondary and tertiary impacts). The allocation of docking priority is the result of a negotiation among the owners/operators of the different ships (or trade-

offs, if the ships are operated by the same company) taking into account the daily costs of each ship, possible penalties, the costs of downstream delays, and the sustainability of production. The cost of the entire scheduled plan is considered on three temporal horizons: short, medium and long term, each one of them carrying a different weight in the performance evaluation procedure of the current schedule. Short term costs have higher impact on the performance of the current schedule, therefore they carry a higher weight in affecting scheduling choices. Typically the short term scheduling horizon is fixed to three months, the medium term is approximately six months and the long term is one year. As shown in figure 1, risk analysis is performed in order to estimate possible delays in loading/unloading calendar slots and the likelihood of finding the designated docking facility busy at the time of ship arrival.

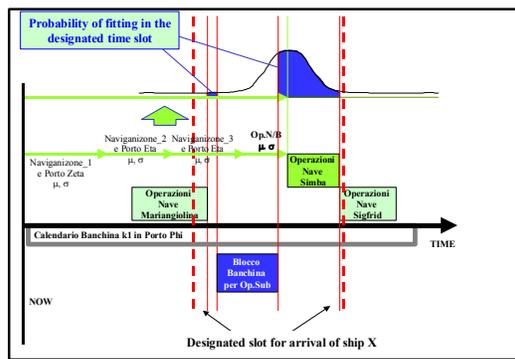


Figure 1: Dock Calendar: Example of Risk Analysis

Any change introduced in the scheduling parameters, such as the ETA of a given ship or the loading/unloading calendar slot for a given product, triggers the re-scheduling of all the events of resource occupation for the corresponding ship mission and causes the update of the relevant object calendars (i.e. the change in ETA to the loading port/dock causes changes in the ETA and saturation levels at the corresponding unloading port/dock). External events such as maintenance, failure, decommissioning of any of the resource, originating in the company but not within the logistic management function, need to be systematically transferred to the system's database as they introduce important changes in the scheduling constraints.

SIMULATION MODULE

The scheduled plan, as generated by the scheduling module, is only statically verified because the module alone does not account for the stochastic

variability of either process parameters or external factors (and their synergies i.e. late/early arrivals and early/late completion of each sequence of operations). For instance, the preliminary schedule fails to account for the probability that the ship may find docking facilities and equipment busy, due to the late arrival of other ships. Such a probability, instead, is fully accounted for by the simulator which tracks the detailed evolution of the scheduled scenario.

The output of the simulation run is a detailed evaluation of the performance measures associated to the scheduled scenario; given the stochastic nature of the simulation model, multiple replications of the same scenario (using different random number generation seeds each time) lead to an estimate of the experimental error, of its impact on the simulation output, and of the associated risks.

In the simulator each physical component of the logistic network (e.g. ships, docks, equipment) and of the production interface (e.g. plants, reservoirs) is modeled as an object described by a set of both static and dynamic parameters. Along with such objects there are purely logical objects such as, Routes, Tactical Missions, Ship Missions, Product Flows, and Calendars, which are user-defined and have specified interactions with the physical objects.

The nature of the objects, their interactions, and their mutually imposed constraints suggest that the basic simulation logic should be both dynamic and discrete-event-based. In other words, the time advancement mechanism is set by the occurrence of un-conditional events, which in turn create the conditions for conditional events to take place, leading to variable time steps. Because the sets of coordinates describing the position of each ship need to be continuously updated along with its Estimated Time of Arrival (ETA), while simulating all the events involving the different system objects, the time advancement mechanism has to be "mixed" in nature, enabling for punctual re-calculation of the continuous variables at each time step. The continuous variables include:

ETA (along with positional and kinematic variables) for each ship currently in navigation during the time period included between the two most recent events, considering the entire set of boundary conditions influencing the motion of the ship, and their variability, according to the following equation:

$$ETA_j(t_i) = ETA_j(t_{i-1}) - Vel_j(t_i - t_{i-1}) \quad (2)$$

where

- ♦ ETA_j = ETA of ship j at time t_i
- ♦ Vel_j = speed of ship j in the period $t_{i-1} \rightarrow t_i$
- ♦ t_i = time of occurrence of event i

Storage levels for each reservoir during the period between the two most recent events, considering

plant production rates as well as import/export activities.

$$LS_j(t_i) = LS_j + \sum [F_k^j (t_i - t_{i-1})] \quad (3)$$

where

- ◆ LS_j = Storage Level of reservoir j at time t_i
- ◆ F_k^j = product flow j in the period $t_{i-1} \rightarrow t_i$
- ◆ t_i = time of occurrence of event i

Statistics update based on the time elapsed between the two most recent events, considering the current status of the simulated objects.

The stochastic nature of the simulated process is accounted for in terms of

- ◆ Navigation times
- ◆ Plant production rates
- ◆ Import/export volumes
- ◆ Component/Ship/Equipment Failures

Probability distributions are associated to such variables, building from historical data, and the Montecarlo technique is employed to extract punctual values out of such distributions during the simulation run. The types of distributions included in the simulation, by category of representation are

- ◆ Component Failures \rightarrow Negative Exponential
- ◆ Component Repair \rightarrow Standard Bell-Shaped
- ◆ Plant Production Rates \rightarrow Beta
- ◆ Navigation Times \rightarrow Beta

The active objects of the simulation are ships, product flows and orders. Evenly distributed statistics sampling events are designed in order to ensure a uniform description of the simulated processes throughout the simulation run. Such events are exactly the same in nature as other process events, but they are only intended to capture pictures of the logistic situation at time intervals of approximately one simulated day.

The physical interactions among ship, dock and loading/unloading equipment are represented in figure 2. Loading and unloading times are functions of the actual product quantity and of the flow rates managed by the available pumps.

As indicated in the figure, the model accounts for simultaneous product loading/unloading operations: as many as allowed by the product-compatible pumps/pipelines available on the dock. If the ship needs to load products from empty reservoirs or unload products into full reservoirs, it enters a conditional wait state until the reservoir in question can be accessed for the designated operations.

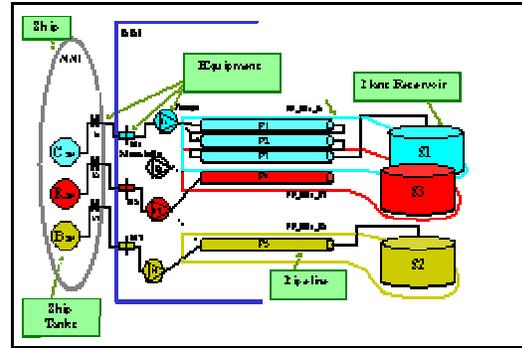


Figure 2: Dock-Ship-Equipment Interactions

Accessibility and compatibility constraints require the simultaneous availability of all the equipment connecting the ship's tank to the plant's storage reservoir and each of the pieces of equipment involved to be compatible with the type of product to be loaded/unloaded. Different combinations of pump/pipeline are possible, as long as they lead to the designated product reservoir and that they are compatible with the product to be transferred.

CONCLUSION

The paper presented the key features and implementation issues of an integrated scheduling and simulation tool for dynamic resource allocation in complex supply chain logistic applications.

The system, currently at the final implementation stages in a large chemical company has been preliminary tested by industrial users. "Turing tests" have been performed to validate the system: such tests involve the participation of Subject Matter Experts (SMEs), namely maritime logistics experts from the company, and requires them to discriminate between schedules developed by the system and schedules developed by human planners. Simulation-based testing shows that the scheduled plans proposed by the system are usually feasible in reality, however they are typically more conservative than the schedules proposed by human experts, thus leading to marginally lower ship capacity utilization and slightly higher costs in favor of higher production sustainability (i.e. negligible risks of stock-out and over-stock events at the production sites.). Such results call for fine-tuning of the decision heuristics built into the scheduling module and of the coefficients weighing the different components of the target cost function.

REFERENCES

- [1] Bruzzone, A.G., R. Mosca, R. Revetria, and A. Orsoni. "System architecture for integrated fleet management: advanced decision support in the logistics of diversified and geographically distributed chemical processing" In *Proceedings of AIS'02 Conference on AI, Simulation and Planning in High Autonomy Systems*, ed. F.J. Barros, and N. Giambiasi, Lisbon, Portugal. April 2002, pp. 309-314
- [2] Mosca R., R. Revetria, A. Orsoni, F. Bertoni, "Fleet Management System Requirements for the Maritime Logistics of the Chemical Industry", *Proceedings of the 4th International Conference on the Modern Information Technology in the Innovation Process of the Industrial Enterprises (MITIP 2002)*, Savona, Italy, June 27-29 2002, pp. 77-81
- [3] A.G. Bruzzone, P. Giribone, "DSS & Simulation for Logistics: Moving Forward for a Distributed, Real-Time, Interactive Simulation Environment", *Proceedings of the Annual Simulation Symposium IEEE*, Boston, MA, USA, 4-9 April 1998, pp. 158-169.
- [4] A.G. Bruzzone, R. Signorile "Simulation and GAs for Ship Planning and Yard Layout", *SIMULATION*, Vol.71, no.2, , August 1998, pp. 74-83.
- [5] Frankler E.G., *Port Planning and Development*, John Wiley and Sons, New York, NY, 1997.
- [6] R. Mosca, P. Giribone, and A.G. Bruzzone, "Study of Maritime Traffic Modelled with Object-Oriented Simulation Languages", *Proceedings of WMC'96*, San Diego, CA, 14-17 January 1996, pp. 87-93.
- [7] R. Mosca, P. Giribone, and A.G. Bruzzone, "Simulation of Dock Management and Planning in a Port Terminal", *Proceedings. of MIC'96*, Innsbruck, Austria, 1996, pp. 129-134.
- [8] Bruzzone, A.G. and Kerckhoffs E.J.H. Simulation in Industry. SCS, 1996, Vol. I pp.633-662.
- [9] Nevins M., Macal C., Joines J. (1998) "A Discrete-Event Simulation Model for Seaport Operations", *SIMULATION*, 1998, vol. 70, no. 4, pp. 213-223.
- [10]Thiers G., Janssens G. "A Port Simulation model as a Permanent Decision Instrument", *SIMULATION*, Vol. 71, no.2, August 1998 pp. 117-125
- [11]Bruzzone A.G., Merkuriev Y.A., Mosca R. (1999) "*Harbour Maritime & Industrial Logistics Modelling & Simulation*", SCS Europe, Genoa, ISBN 1-56555-175-3
- [12]Bruzzone A.G., Gambardella L.M., Giribone P., Merkuriev Y.A. (2000) "*Harbour Maritime & Multimodal Logistics Modelling & Simulation 2000*", SCS Europe, Genoa, ISBN 1-56555-207-5
- [13]Bruzzone A.G., Giambiasi N., Gambardella L.M., Merkuriev Y.A. (2001) "*Harbour, Maritime & Multimodal Logistics Modelling & Simulation 2001*", SCS Europe, Marseille, ISBN 90-77039-03-1
- [14]Bruzzone A., Signorile R. (2001) "Container Terminal Planning by Using Simulation and Genetic Algorithms", *Singapore Maritime & Port Journal*, pp. 104-115 ISSN 0219-1555.
- [15]Bruzzone A.G., Mosca R. (1998) "Special Issue: Harbour and Maritime Simulation", *Simulation*, Vol.71, no.2, August
- [16]Merkuriev Y., Bruzzone A.G., Novitsky L (1998) "*Modelling and Simulation within a Maritime Environment*", SCS Europe, Ghent, Belgium, ISBN 1-56555-132-X
- [17]Liu, J.S. 2001. *Monte Carlo Strategies in Scientific Computing*. Springer Verlag, Hamburg.
- [18]Fishman, G.S. 1996. *Monte Carlo: Concepts, Algorithms, and Applications*. Springer Verlag,
- [19]Gentle, J.E. 1998. *Random Number generation and Monte Carlo Methods*. Springer Verlag, Hamburg.

A METHOD FOR GENERATING STRUCTURALLY ALIGNED GRIDS USING A LEVEL SET APPROACH

ALIREZA SHEIKHOLESLAMI, CLEMENS HEITZINGER, AND SIEGFRIED SELBERHERR

*Institute for Microelectronics, TU Wien
Gußhausstraße 27–29/360
Vienna, Austria
Email: sheikholeslami@iue.tuwien.ac.at*

Abstract: We describe a technique to generate structurally aligned triangular grids. The main advantage of this method is the adjustable propagating speed of the front in different parts of the simulation domain in order to achieve different densities of triangles in each part of the simulation domain. This feature is usually needed in semiconductor device simulation. Other advantages of this technique are twofold: firstly, the grid can be very well adapted to the structures, and secondly, the grid elements fulfill desirable requirements like Delaunay triangulation and the minimum angle criterion. The technique is based on viewing the boundary of the simulation domain as a front which is propagated structurally at different speeds. A smooth propagation is achieved by the level set method by viewing the front as the zero level set of a higher dimensional function whose equation of motion is described by a partial differential equation.

KEYWORDS: Grid generation, Delaunay triangulation, level set method, semiconductor device simulation.

INTRODUCTION

We describe a method to generate structurally aligned triangular grids and illustrate it in two examples. We use the level set method to propagate the boundary of the simulation domain as a front by viewing it as the zero level set of a higher dimensional function with an adjustable speed depending on how fine the triangular grid should be. The equation of motion of this higher dimensional function is given by a partial differential equation, which is approximated by techniques borrowed from the numerical solution of hyperbolic conservation laws which guarantee that the correct entropy satisfying solution will be produced. The evolving front is thus a hypersurface, e.g., a curve in two space dimensions and a surface in three space dimensions. The resulting algorithm can be used to generate two and three dimensional grids around complex bodies containing sharp corners and significant variations in curvatures. We use this technique to generate different grids around a variety of shapes for different device structures.

The most important advantage of this method is the adjustable propagating speed of the front which provides an automatic way for generating grids with different densities of grid cells in particular parts of its domain. The his-

tory of two-dimensional process and device simulation leads to the observation that a stable triangulation engine is one of the most important prerequisites for simulation purposes. In the second part of our algorithm the final grid elements are produced using the TRIANGLE program [Fang and Piegler 1993, Shewchuk 1996]. Furthermore, thereby grids are very well adapted to the structures and are of high quality because we can enforce minimum angle criterion which guarantees that the triangles have angles which are equal or greater than a certain minimum angle and therefore we can well control the shape of the triangles.

Although the level set method has been used for generating structurally aligned grids [Sethian 1994], the method presented there cannot generate anisotropic grids and no condition concerning the quality of the grid, e.g., minimum angles, can be enforced.

The outline of this paper is as follows. Firstly, the basic ideas of the level set method are shortly explained. Secondly, the grid generation algorithm as a combination of the level set method and triangulation is presented. Thirdly, an algorithm for equalizing the length of segments is presented. Finally, examples for two simple initial structures and a real device structure are given.

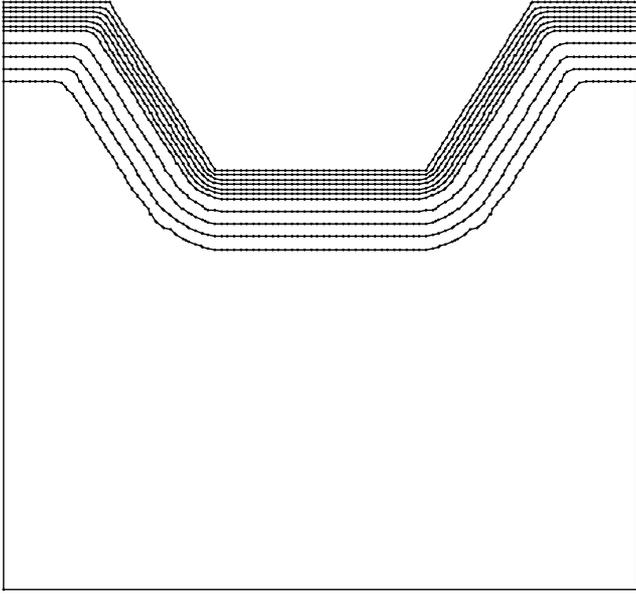


Figure 1: The extracted boundaries at 10 time steps.

THE LEVEL SET METHOD

The level set method [Sethian 1999] provides means for describing boundaries, i.e., curves, surfaces or hypersurfaces in arbitrary dimension, and their evolution in time, which is caused by forces or fluxes normal to the surface. The basic idea is to view the curve or surface in question at a certain time t as the zero level set (with respect to the space variables) of a certain function $u(t, \mathbf{x})$, the so called level set function. Thus the initial surface is the set $\{\mathbf{x} \mid u(0, \mathbf{x}) = 0\}$.

Each point on the surface is moved with a certain speed normal to the surface which determines the time evolution of the surface. The speed function $F(t, \mathbf{x})$ generally depends on the time and space variables and we assume for now that it is defined on the whole simulation domain and for the time interval considered.

The surface at a later time t_1 shall also be considered as the zero level set of the function $u(t, \mathbf{x})$, namely $\{\mathbf{x} \mid u(t_1, \mathbf{x}) = 0\}$. This leads to the level set equation

$$u_t + F(t, \mathbf{x}) \|\nabla_{\mathbf{x}} u\| = 0,$$

$u(0, \mathbf{x})$ given

in the unknown variable u , where $u(0, \mathbf{x})$ determines the initial surface.

Having solved this equation the zero level set of the solution is the sought curve or surface at all later times.

Although in the numerical application the level set function is eventually calculated on a grid, the resolution achieved is in fact much higher than the resolution of the grid, and hence higher than the resolution achieved using a cellular format on a grid of same size.

In summary, first the initial level set grid is calculated as the signed distance function from a given initial surface. Then the speed function values on the whole grid are used to update the level set grid in a finite difference or finite element scheme. Usually the values of the speed function are not determined on the whole domain by the physical models and therefore have to be extrapolated suitably from the values provided on the boundary, i.e., the zero level set. A fast and efficient level set algorithm combining extending the speed function and narrow banding was presented in [Heitzinger et al. 2002, Heitzinger and Selberherr 2002]. There a surface coarsening algorithm similar to the one used in this work was described as well.

GENERATING THE LEVEL SET STRUCTURED TRIANGULATED GRID

Our basic philosophy is to advance the front through the simulation domain using different speed functions. Throughout this section we restrict ourselves to two-dimensional grids. At discrete chosen time intervals, zero level set functions are constructed using a boundary extraction algorithm. In our example we have assumed a constant speed for the first 6 time steps and $8/3$ times this speed for the next 4 time steps. This is shown in Fig. 1. We can see that the whole simulation domain is now divided into three different parts according to three different grid resolutions depending on the application. An arbitrary number of segments and speed functions can be used if desired.

Based on the edges constructed in the first step the grid generator TRIANGLE is used to obtain a Delaunay triangulation. In this example we demanded that the produced triangles have no angles smaller than 20 degrees. Requiring minimum angles is important since it enables a priori error estimates and estimates of the order of convergence [Knabner and Angermann 2000].

Fig. 2 shows the triangulated simulation domain. Because of different lengths of the segments which are obtained by each boundary extraction, we can clearly see that this triangulation contains triangles which are too small. An enlargement of this undesirable situation is shown in Fig. 3. We introduce an algorithm for overcoming this problem in the next section.

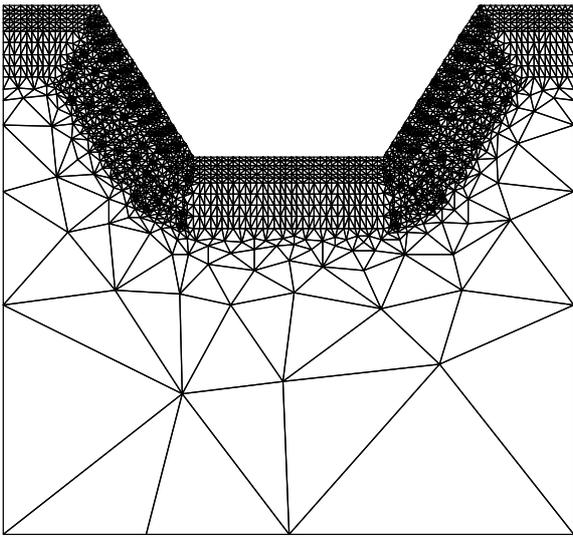


Figure 2: The triangulated grid without using the segment length equalizer.

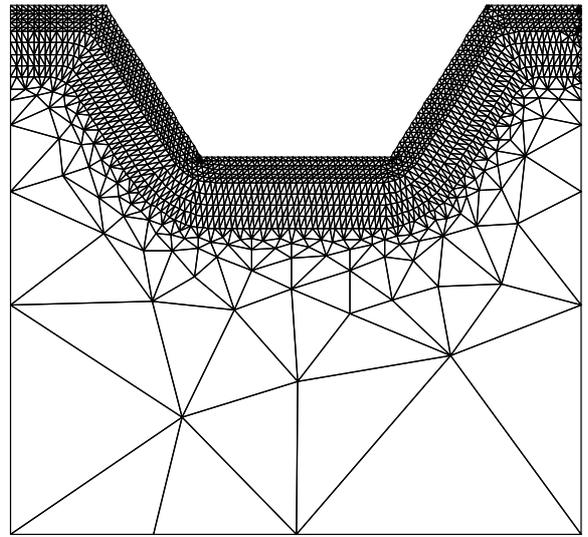


Figure 5: The triangulated grid is caused using the segment length equalizer.

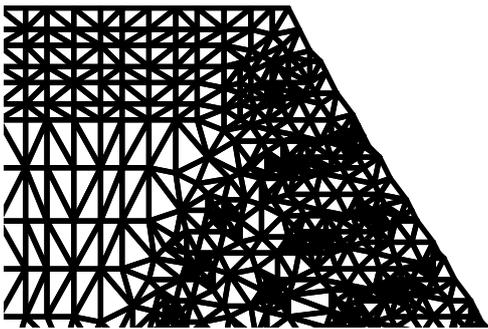


Figure 3: A part of the above grid on a larger scale.

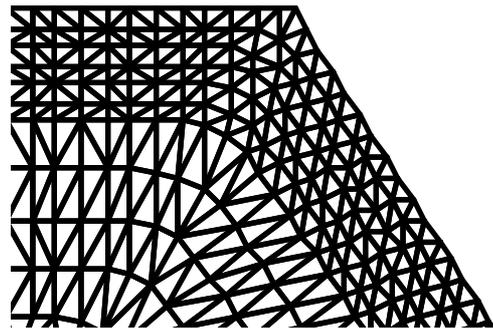


Figure 6: A part of the above grid on a larger scale.

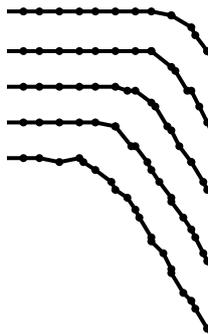


Figure 4: The last five steps of advancing the front is shown partly on a larger scale. The varying lengths of the segments are shown clearly.

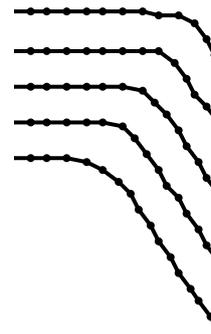


Figure 7: The last five steps of advancing the front is shown partly on a larger scale after equalizing the lengths of the segments. The length of the segments are not more different.

THE SEGMENT LENGTH EQUALIZER

To find the origin of this problem we briefly describe the boundary extraction algorithm which uses an interpolation method to find the points of the boundary and represents these as a list of segments with different lengths. Fig. 4 shows a part of the last five steps of advancing the front on a larger scale to show more clearly the varying lengths of the segments. The segments may become arbitrarily small and are the cause of the areas of dense triangles. To overcome this problem we need to ensure that all segments of the boundary have about equal lengths.

We start the algorithm by choosing a certain common length d for all segments. In our example we chose the minimum value of the vertical or horizontal distance between the points of our original rectangular grid which is used in the level set step. The first point of the extracted boundary stays without any changes but to find the second point we have to discern two cases. The first one is that the distance between the second and first point of the originally extracted boundary is equal or greater than our d and in the second one this distance is smaller than d . In the first case we compute the second point of the new boundary in this manner that we get a point which fulfills two restrictions: first, the caused segment must be along the first segment of the originally extracted boundary and second, the length of the new segment must be equal to d . In this case the new segment is a part of the old segment but the length of the new segment is equal or smaller than the old one. In the second case we compute the second point of the new boundary along the next segment of the origin boundary and like the first case fulfilling the length requirement. In this case the new segment is parallel to the second segment of the origin boundary and the length of the new segment is greater than the old one. These steps are iterated until we reach the boundary of the domain. Fig. 7 and Fig. 5 show the resulting segments with the enlargement and triangulated grid after equalizing the lengths of the segments. Furthermore in Fig. 6 a part of Fig. 5 is shown on a larger scale. In Fig. 8, Fig. 9 and Fig. 10 we show a simulation domain with a rectangular advancing front as another example and the resulting grid also with the enlargement.

GRID GENERATION FOR A REAL DEVICE STRUCTURE

Fig. 11 shows the device structure of a trench gate UMOS transistor. This device is useful for power switching at high voltages [Bulucea and Rossen 1991, Shenai 1992, Dharmawardana and Amaratunga 1998]. Trench gate UMOS transistors also provide advantages because of their geometric layout, i.e., because their inversion and accumulation channel regions are perpendicular to the wafer surface. Hence they enable to maximize the ratio of cell perimeter to area

and thus increase packing density. An analytical model for a typical trench gate UMOS transistor is given in [Dharmawardana and Amaratunga 2000].

The model is derived using the charge control analysis of the channel and drain drift regions and gradual channel approximation is assumed to be valid in modeling the channel region. The shape of the different junctions is obtained by the doping concentration profile which is modeled with a Gaussian distribution.

For the grid generation we used four boundaries which follow the three junctions. At the $n^+ - p$ junction we used three boundaries in each direction of the initial boundary which follow the junction with a distance of $0.02\mu\text{m}$ between any two adjacent boundaries.

At the $p - n$ junction we used one boundary above and below the initial boundary and a distance of $0.02\mu\text{m}$. At the $n - n^+$ junction in the lower part of the device we took into account two boundaries with a distance of $0.5\mu\text{m}$ going downwards from the initial boundary following the junction. For the last prescribed edges we started at the tight hand side of the p region and moved to the left using three boundaries at a distance of $0.005\mu\text{m}$.

Finally, we applied the TRIANGLE program requiring a minimum angle of 25° with the prescribed edges as input. The grid produced is shown in Fig. 12, and it resolves very finely the junction areas as demanded.

CONCLUSION

A technique for generating structurally aligned triangulated grids using the level set method was described and implemented in two dimensions. In contrast to previously generated structurally aligned grids based on the level set method [Sethian 1994] the anisotropy of the grids and their quality can be controlled. The simulation domain can be divided into parts with different resolutions using adjustable speeds for advancing the front through the simulation domain with level set method. This adjustable grid resolution is essential in semiconductor device simulation where high resolutions are required in certain parts of the simulation domain. Furthermore the grid can very well adapted to different structures. Finally enforcing the minimum angle criterion is important for the numerical behavior of the subsequent finite element calculations and ensures high quality grids. At the same time, the diameter of the triangles may vary over several orders of magnitude within one simulation domain (cf. Fig. 5, Fig. 9, and Fig. 12). Our technique enables to produce triangulated grids for each form of semiconductor device structure with demanded resolution at different junctions (cf. Fig. 12).

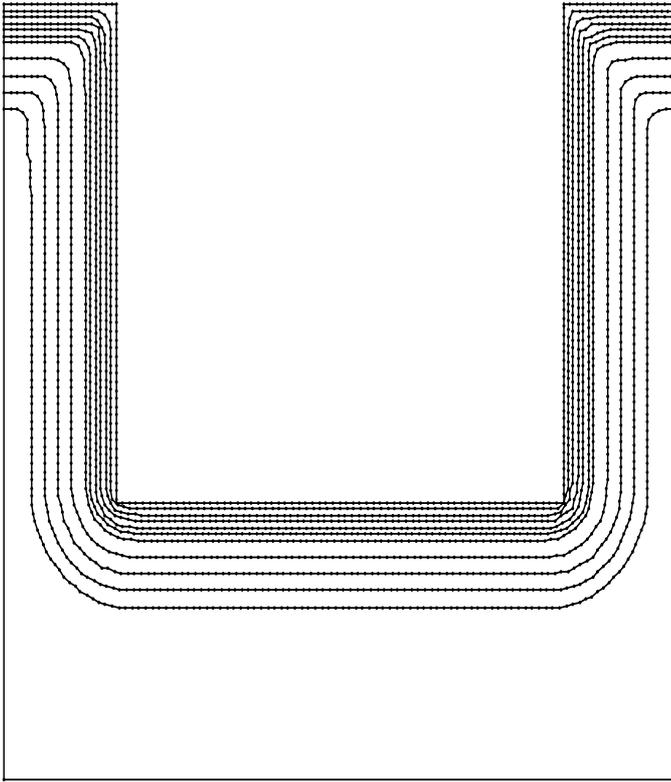


Figure 8: The advancing rectangular front after 10 time steps. As same as Fig. 5 the ratio of the speed in the first 6 steps to the last 4 steps is $3/8$.

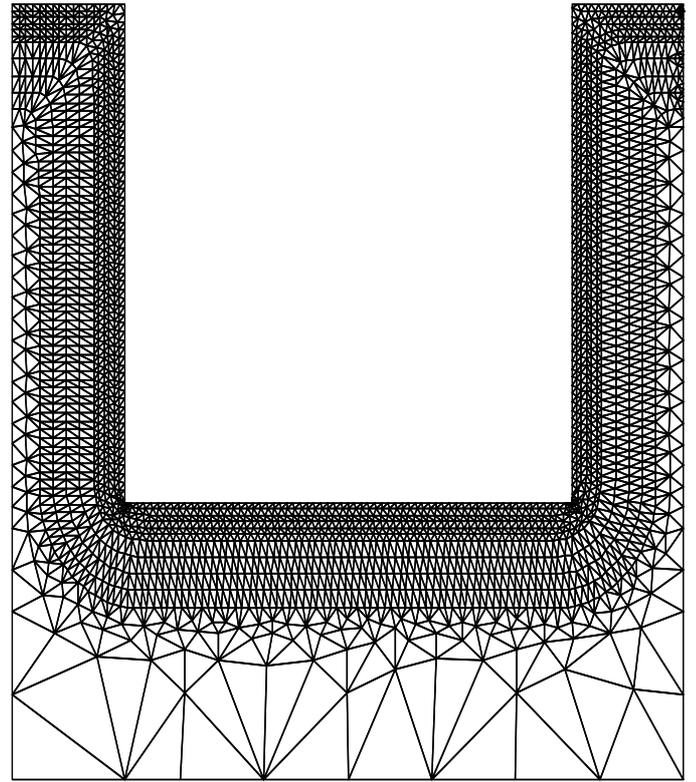


Figure 9: The triangulated grid of simulation domain in Fig. 8.

REFERENCES

- Bulucea, C. and R. Rossen, 1991, "Trench DMOS Transistor Technology for High Current (100A Range) Switching". *Solid-State Electron.*, 34(5):493–507.
- Dharmawardana, K. and G. Amaratunga, June 1998, "Analytical Model for High Current Density Trench Gate MOSFET". In *Proc. of the 10th International Symposium on Power Semiconductor Devices and ICs (ISPSD 1998)*, pages 351–354, Kyoto, Japan.
- Dharmawardana, K. and G. Amaratunga, December 2000, "Modeling of High Current Density Trench Gate MOSFET". *IEEE Trans. Electron Devices*, 47(12):2420–2428.
- Fang, T.P. and A. Piegl, 1993, "Delaunay Triangulation Using a Uniform Grid". *IEEE Computer Graphics and Applications*, pages 36–46.
- Heitzinger, C.; J. Fugger; O. Häberlen; and S. Selberherr, September 2002, "On Increasing the Accuracy of Simulations of Deposition and Etching Processing Using Radiosity and the Level Set Method". In *European Solid-State Device Research Conference (ESSDERC 2002)*, pages 347–350, Florence, Italy.
- Heitzinger, C. and S. Selberherr, June 2002, "On the Topography Simulation of Memory Cell Trenches for Semiconductor Manu-

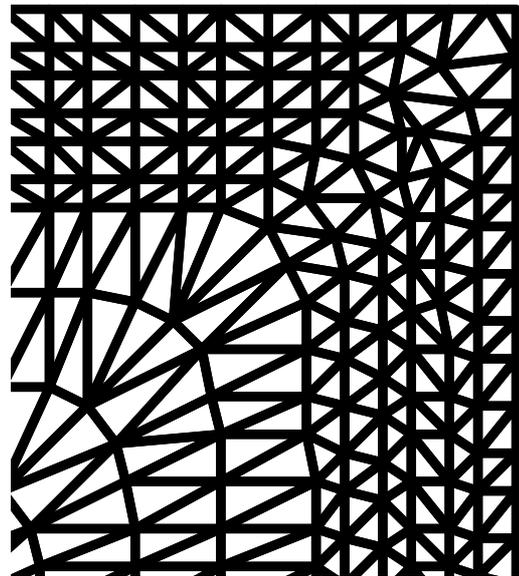


Figure 10: Fig. 9 is shown partly on a larger scale.

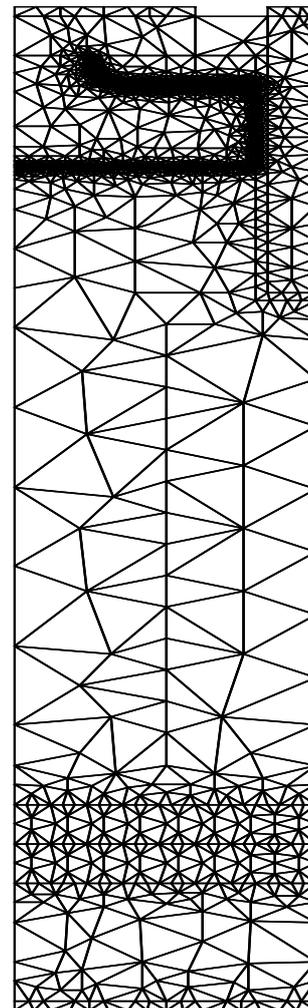
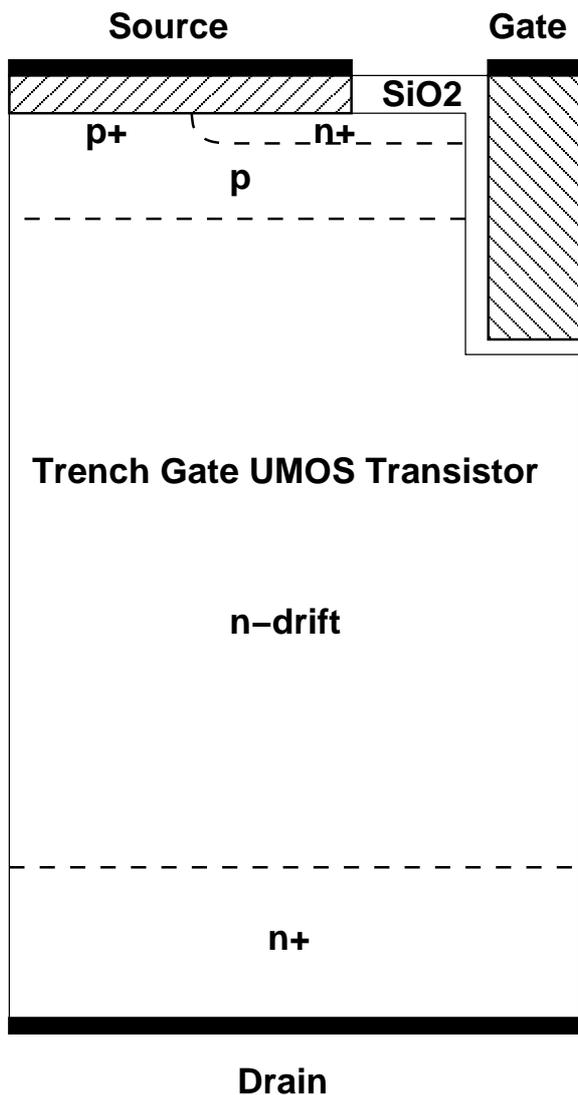


Figure 12: The grid generated for the device in Fig. 11.

Figure 11: Structure of TMOSFET. The half cell pitch of the device is $2.5\mu\text{m}$ and its n drift length is about $9.5\mu\text{m}$.

facturing Deposition Processes Using the Level Set Method". In *16th European Simulation Multiconference (ESM 2002): Modelling and Simulation*, pages 653–660, Darmstadt, Germany.

Knabner, P. and L. Angermann, 2000, *Numerik partieller Differentialgleichungen*. Springer, Berlin.

Sethian, J.A., 1994, "Curvature Flow and Entropy Conditions Applied to Grid Generation". *J.Comput.Phys.*, pages 440–454.

Sethian, J.A., 1999, *Level Set Methods and Fast Marching Methods*. Cambridge University Press, Cambridge.

Shenai, K., 1992, "Optimized Trench MOSFET Technologies for Power Devices". *IEEE Trans.Electron Devices*, 39(6):1435–1443.

Shewchuk, J.R., May 1996, "Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator".

First Workshop on Applied Computational Geometry (Philadelphia, Pennsylvania), pages 124–133. <http://www-2.cs.cmu.edu/quake/tripaper/triangle0.html>.

AUTHOR BIOGRAPHY



AHIREZA SHEIKHOLESAMI was born in Babol, Iran, in 1971. He studied electrical engineering at the University of Science and Technology in Tehran and "Technische Universität Wien", where he received the degree of "Diplom-Ingenieur" in 2002. He joined the "Institut für Mikroelektronik" in April 2002, where he is currently working on his doctoral degree. His scientific interest is focused on process simulation for semiconductor manufacturing.

PERFORMANCE PROTOTYPING - GENERATING AND SIMULATING A DISTRIBUTED IT-SYSTEM FROM UML MODELS

ANDREAS HENNIG, ANJA HENTSCHEL and JAMES TYACK

Siemens AG, Corporate Technology, CT SE 1,

Otto-Hahn-Ring 6, 81739 München, Germany,

Andreas.Hennig@siemens.com, Anja.Hentschel@siemens.com, jamesahtyack@hotmail.com

ABSTRACT: In this paper, we present the concept of “performance prototyping” – the automatic generation and deployment of small components emulating the intended behaviour of real components under design into real IT-infrastructure and environments. Allowing far more effective and consistent production of prototypes than manual prototyping, performance prototyping enables the designer of systems and their infrastructure to assess the impact of various load scenarios, design choices and configuration alternatives very early in the project, and thus allows to synchronize infrastructure planning and system development closely. Rather than the “build first – tune, change & upgrade later” approach, performance prototyping enables to design, plan and build hard-, soft and middleware in closer coordination and to meet performance targets in fewer cycles.

The basic concepts of the UML-based notation of performance aspects is presented which was designed to be compatible with current UML-tools and fit into their normal usage in development practises. We then discuss the interaction and differences of performance prototyping (“in-vivo performance simulation”), performance prediction in dedicated methods and tools (“in vitro” performance simulation) and load-testing as well as the differences to manual prototyping and benchmarking. A method of integrating performance prototyping into commercial UML-tools is presented, particularly with view on the challenges of generating multi-target and multi-protocol prototypes that interact across targets, platforms and protocols in the way prescribed by the model. We then describe the model, prototype, experiments and findings based on a JSP example before the conclusion of the paper.

KEYWORDS: Software Performance Engineering, UML Modelling, Performance Annotation, Performance Prototype Generation, Deployment, Simulation, Load Testing, Benchmark

INTRODUCTION

Larger IT-Systems or products, particularly if distributed and networked often have complex interactions between the various **hardware** (HW) entities (hosts, network nodes and links, peripherals), the different layers of operating system, execution environments and server processes (like web, servlet, application or database-server) which in this paper we will summarily call **middleware** (MW), and finally the main behavioural software components implementing the **business logic** of the system (SW). With the trend to standardized off-the-shelf products with standard interfaces and protocols, the division between the teams responsible for “SW-development” and “HW planning, installing configuration and operation” tends to be somewhere within the middleware layers, where the “SW-team” focuses on functionality and interfaces, the “HW-team” on configuration issues.

For most large systems (e.g. ERP systems, intra/internet-portals, online shops, information and control systems...), performance is a central criteria and critical success factor. With increasing expectations of the users to perceived performance, an unsatisfactory performance might - and frequently does - endanger the system's/product's/project's success irrespective of functionality and design. Since these systems are often business critical and/or highly image critical, insufficient performance can incur heavy costs (e.g. compensation, penalties, superfluous hardware, loss of market shares and value), delay the going-live, and reduce the system's benefit (e.g. through lack of user acceptance and retention, sub-optimal decisions based on out-of-date information). Performance problems can derive from a variety of sources, including sub-optimal configuration, insufficient computational power, inefficient implementations of individual modules and design flaws. The earlier lie within the responsibility of the HW-team and can be rectified by tuning or – although more expensive – by additional HW. The later are not only more difficult

to detect, they are also caused much earlier in the project and are therefore far more difficult to correct in time if detected towards the release date – apart from the much higher cost.

A reason behind the numerous performance failures of IT-Projects (and the ensuing mutual accusations) is the far-too-late assessment of performance, which derives partly from lack of performance-awareness, partly from the restrictions found in the predictive methods that could be used to measure and control progress in terms of performance goals. Ideally, for large and performance-critical systems, there should be the role of an overall “performance engineer”, who gathers performance assumptions and requirements, predicts overall final performance based on the current implementation progress, coordinates and mediates between the conflicting interests of HW and SW teams and executes in-development and pre-release load-test to substantiate development and release decisions.

We group the methods predicting the live performance coarsely into “benchmarking”, “simulation”, “prototyping” and “load-testing” (ref. Fig. 1):

“**Benchmarking**” employs small standardized activities (e.g. integer or floating point operations, memory or disk access...) and measures how many a given system can execute per second. While these “synthetic” benchmarks accurately describe one performance aspect of a given HW (and thus help to compare between a larger number of HW/MW alternatives), they - inherently – do not measure the performance in terms of the transaction types of the intended system (for the purpose of this paper, we

include specialized application-specific benchmarks under prototypes, below). Benchmarks thus provide a basis for HW-choices, but can only rarely be used for good estimates of the new system’s performance.

“**Simulation**” builds an abstracted model of the infrastructure (HW and MW), the behaviour of the business logic as well as the expected load from internal sources and users. The resulting performance aspects are obtained using various methods (e.g. queuing networks, stochastic methods, discrete event simulation...) of the software performance engineering domain (SPE, [Smith90]). Since the models are built and evaluated in an environment completely separate from the physical system (somehow “in vitro”), Simulation can be applied even before the first HW is purchased. However, a common problem of simulation lies in the need to build complex models (where particularly the MW-models depend on product release-cycles in a fast paced industry), to ascertain numerous model parameters and to – often manually – transfer the design of the intended system into a suitable representation. The resulting uncertainties and potential inconsistencies as well as the effort and time required restrict the use of simulation to (parts of) systems with clear boundaries that can be abstracted easily and reliably and have focused performance inquiries.

Manual “**Prototyping**” is normally performed in the early phases to establish suitability of a platform/middleware/ technology under consideration for the intended purpose, i.e. often functionality and interoperability orientated. The prototypes can then be load-tested, but since the effort required to manually develop prototypes restricts their

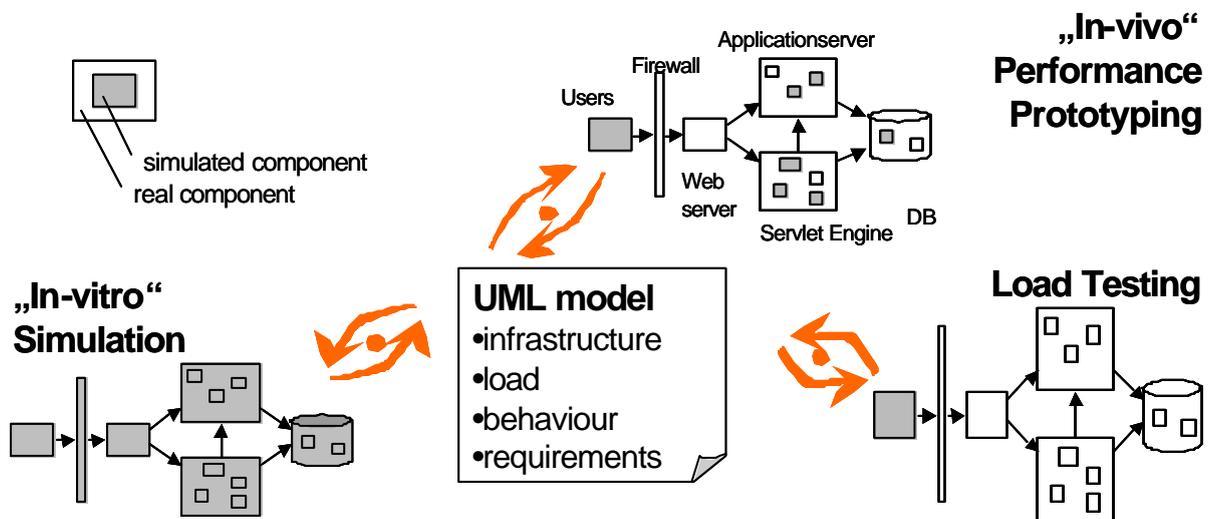


Fig. 1: One model throughout performance development lifecycle – integrating prediction, prototyping and acceptance testing by varying real and simulated components

comprehensiveness and variability, the results are at best a basis for extrapolation. Manual Prototypes differ from benchmarks since they measure small application-specific activities, but suffer from the same necessary restriction to few aspects. Furthermore, manual prototypes also bear the danger of errors, inconsistencies and incompatibilities.

“**Load-testing**” transforms the expected user behaviour into small simulated “virtual users” which are deployed onto real infrastructure (sometimes called “load-generators” or “load-injectors”), which exercise a real system or prototype with real and realistic load. Load-tests are either used as gruelling acceptance tests or during development to test performance aspects of individual modules or prototypes. Load-testing a finished system is the least predictive method discussed – it is however, predictive in terms of the anticipated user behaviour, which might vary greatly from the behaviour of real users.

We therefore propose automated “**Performance Prototyping**” as a means to overcome the above limitation of flexibility, coverage, efficiency and application-specificness. For this, models of the system’s intended business logic and planned infrastructure are maintained in UML. Since these documents normally need to be produced during development in some way, they only need to be annotated with some additional information like resource consumption, performance requirements and deployment and access locations.

This furthers

- a) understanding and consistency between the development teams and the performance engineer,
- b) reduces the effort required to build the main model (ideally now through the teams themselves),
- c) provides a simple and concise notation of performance aspects and
- d) permits to automatically generate and deploy comprehensive prototypes ready for load-test.

Unfortunately, the UML is currently not always used to document the entire system, particularly the infrastructure and HW/MW aspects thereof. But even if the few relevant HW parts need to be transformed into UML based on the input of the HW-team, above benefits still apply. From the model, a performance prototype is generated automatically and can be deployed on real target systems, where it can be tested “in-vivo”. Due to the fast cycle times with automatic performance prototypes, various alternatives in HW, MW or SW can be investigated with affordable effort.

In the remainder of the paper, we shortly present the information required for performance prototypes and how and where they can be modelled in UML based on a JSP example. We then describe the architecture of the prototype generator and how the required flexibility can be achieved. After the presentation of some experimental results, we give an outlook on synergies and interactions between performance prototyping, simulation and load-testing before the conclusion of the paper.

MODELLING

PERFORMANCE PROTOTYPES

A Performance prototype requires the description of the characteristics that are (or could be) performance-relevant. Programming language and execution environment certainly affects the performance (e.g. C++ being faster than VBA, or tomcat generally being faster than JServ) as does the interaction of components (a calls b calls c). The precise values within the requests (e.g. a lookup-key) or the responses (e.g. the retrieved data) do hardly affect performance, while the size, encoding and protocol of requests and responses are likely to have an impact. Performance-relevant information consists of infrastructure, load, behaviour and requirements information.

We model **infrastructure** information in deployment diagrams as the obvious representation in UML (see also [Williams98], [Dimitrov02], [Mirandola00], [Petriu99]). The diagram describes computational resources, their connection and – optional – the number of instances of a component, which we call its multiplicity according to the corresponding UML attribute. In Fig. 2, we show a sample deployment of two webserver hosts, both hosting a JSP-engine; webserver1 hosts additionally a database. A LAN component (used to represent a bus-topology in UML) connects the servers to two types of clients, which differ in the numbers of browsers running on it. The multiplicity of Client1 indicates, that there are \$NumClients instances (e.g. 20 in different network locations) in the system.

In addition to the core SPE notation as proposed in [Hennig02], performance prototyping requires additional information about the real hosts and servers used (e.g. IP numbers and ports of the webserver). Mainly, this is the “access” information, which denotes how a component is addressed for requests. In the deployment diagram, the access-path can be annotated as the tagged value “spe.ppr.access” of the nodes or objects. For webserver, this would simply be the URL of the JSPs themselves, which could include parameters,

port, username and password as well [RFC 1738, 1808]. For a database, the access-path would for example contain the JDBC connection information. Since performance prototyping aims to automatically deploy the components, further information is needed to specify where and how to deploy the component. The tagged value “spe.ppr.upload” therefore contains a URL that indicates the deployment destination. In the case of a JSP component spe.ppr.upload points to a file:// location or the URL to a cgi-script into which the JSP-source code can be uploaded.

The **behaviour** of a prototype system as well as the **load** placed onto it is described as the generation and exchange of messages, requests and responses, which we model in the sequence diagram. For the discussion concerning the use of UML in general and the use of state vs. sequence diagram in particular, see [Hennig01] and [Hennig02]. In Fig. 3 a simple interaction pattern (“workflow”) is depicted, several instances (“jobs”) of the same or different workflows can occur concurrently.

The load is generated by the actor, representing multiple users that execute the same workflow with given arrival and think times. In the example in Fig. 3, the webbrowser is modelled to submit various http-request per “click” of the actor (e.g. for nested or consecutive http like redirecting, frames, included image). Since the “browser” corresponds to different client machines in the deployment diagram, we indirectly model the network region, where the load

should originate.

Parameters passed along the http-requests will inform JSPs which step of which workflow they are expected to execute. The generated JSP code contains the information how to execute a specified step (i.e. how much computation is needed, how large the response will be, which other components need to be called). Resource usage can be modelled flexibly (in a spe.use.{resourcetype} tagged value) and currently includes but is not limited to time delay, cpu consumption, memory usage, I/O-volume and various types of semaphores. Since cpu-consumption depends on the speed of the hosting hardware, we specify it in number of iterations of classical benchmark operations like dhrystone, whetstone or of less formal but expandable operations like string-operations or heap-sorting.

Performance **requirements** like the maximum permissible response time for certain requests or the time to complete an entire workflow can be denoted as numerical expressions based on timestamps collected during the execution of the jobs. The timestamps as well as additional job-specific variables (e.g. a randomly chosen think-time of the simulated user) can also be used to gather workflow-related statistics. Infrastructure-related statistics like cpu-usage can be gathered and evaluated using network and performance management tools (e.g. based on SNMP, rstat or the windows performance monitor)

GENERATING PERFORMANCE PROTOTYPES

After modelling infrastructure, load, behaviour and requirements in various UML diagrams, an experiment definition diagram is used to specify the subset of “investigated diagrams” and overall parameters like a scalability factors for number of clients (e.g. \$numClients in Fig. 2), the workload or an overall think time.

A series of scripts can then be started directly from the UML-Tool (currently TogetherJ from Togethersoft) which controls the prototyping cycle depicted in Fig.4. The prototyping is integrated transparently in order to ensure user acceptance and achieve high impact by frequent use of the method through seamless integration of the end-to-end process. The selected diagrams and required information is extracted from the UML-tool and stored in an intermediate XML representation. From this experiment description, a converter produces the prototype parts for the various target platforms and deploys them into their respective environment. The User behaviour (the load characteristics from the

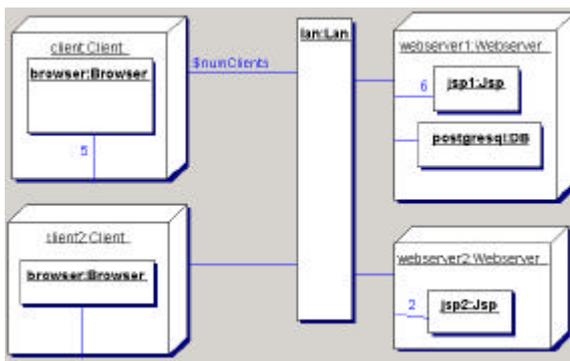


Fig. 2: UML deployment model of the prototype

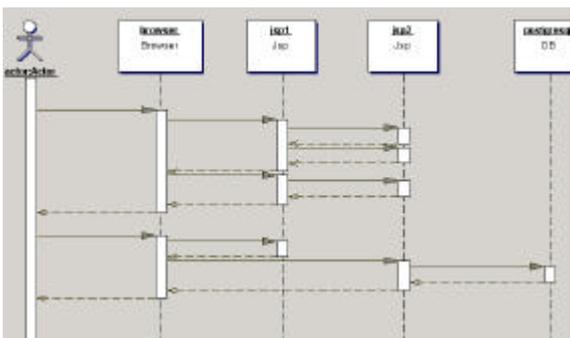


Fig. 3: UML behavioural model of the prototype

actor and in our example also the html-nesting logic

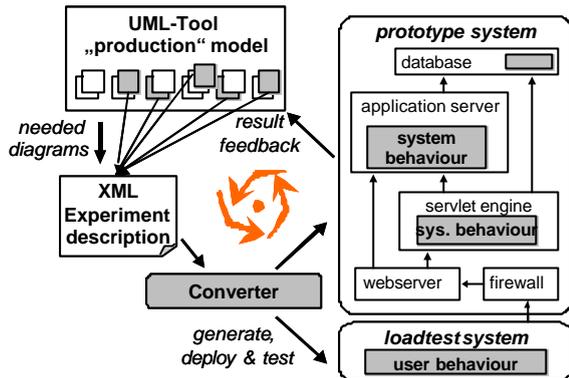


Fig. 4: End-to-end performance prototyping cycle from UML-tool

of the browser) results in a script for a commercial load-testing tool (LoadRunner from Mercury Interactive). From the behavioural description of the JSPs, the source code is generated in JSP syntax and uploaded into the JSP directories of the respective servlet-engines. Since in our example, the database request is a simple SQL statement and not a stored procedure, there is no need to generate code for the database as we can include the statement into the generated JSP code. The used tables, however, need to exist in the database.

The results obtained from internal statistics (e.g. response times) and network monitoring could be fed back into the UML model. Since commercial monitoring tools often provide specific analysis modules (e.g. drill-down or regression), this step might be performed in a specialized separate tool. After analysing and interpreting the data, the UML-Model can be updated and modified accordingly and the cycle started again.

The challenge of generating the prototypes lies in the potentially heterogeneous target platforms (programming languages, execution environment) and the communication protocol they use. Ultimately, each implementation platform should be able to issue requests to any other (sensible) type of platform using a number of possible protocols. Over these protocols, the control information of the prototype (e.g. workflow name, instance and current step) needs to be transmitted without altering the protocols. While flexible protocols like http, where additional parameters can easily added to the URL

without interferences, easily accommodate for this, more rigid protocols like SOAP or RMI will be more challenging.

EXAMPLE PROTOTYPE & EXPERIMENT

For experimental evaluation, we used the above simple behaviour and varied the deployment configuration by altering the host on which the JSP and database components were deployed to (ref table 1). Both hosts run under Linux, dax is a dual-processor server, ibex a single-processor workstation.

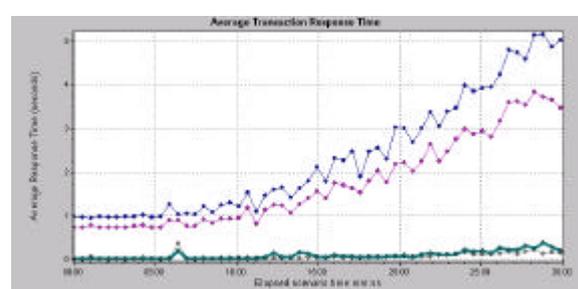
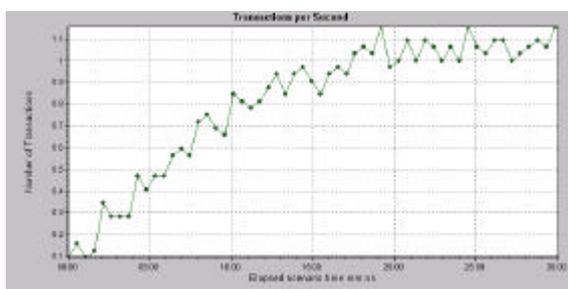
	JSP1	JSP2	DB
Test A	Dax	Dax	Dax
Test B	Dax	Ibex	Dax
Test C	Dax	Ibex (tuned)	Dax
Test D	Dax	Dax (tuned)	Dax

Table 1: testing deployment variations

In the sequence diagram, we specified the first call from the browser to jsp1 to be resource intensive (e.g. for analysing user authorization) as well as the processing of the database results in jsp2. We defined a think time of 3 seconds on average, which represent the time a user would need between clicks in the browser. For tests C and D, we assumed a scenario where a proposed tuning measure is expected to improve database processing by 60%, but since it would entail large modification efforts, an impact analysis should be carried out before any decision is taken. Resource consumptions are therefore adjusted in the sequence diagram, the prototype newly generated, deployed and tested to provide the answer in short time.

Each load test ran for 30 minutes and increased the load every 2 minutes by one additional simulated user. Fig 5. shows the achieved rate of fully completed transactions per second (TPS, number of finished workflow instances/s) of Test A. The system went into saturation after 18 minutes with an approximate capacity of 1.05 TPS caused by 10 concurrent simulated users (Fig. 5). Further users did not increase the transaction rate but only resulted in increased response times due to shared use of the CPU resources.

The response times for the first user (on a basically idle system) where in sum 1.8s without the prescribed think times, but rose to 5.5s at the saturation point of the system. Fig. 6 shows the



response times of the four constituent requests from the browser. The result for the further tests is shown in table 2.

	Capacity		Response time	
	TPS	Users	“idle”	saturated
Test A	1.05	10	1.8s	5.5s
Test B	0.35	5	2.8s	5.9s
Test C	0.58	7	1.9s	6.6s
Test D	1.33	13	1.3s	4.7s

Table 2: Performance Measurement of the prototype

We can see, that – not surprisingly – deploying parts of the application on ibex did not improve the overall capacity of the system. The proposed improved database processing step, however, would significantly improve the overall performance in terms of capacity (by 26%) and response times (28%-36%) and should therefore be attempted.

Once the initial model was built, each complete performance prototyping cycle, i.e. evaluation of a further design variation, was completed in less than an hour. The generation and deployment of the prototype itself took about two minutes, the remainder being spent on running and analysing the load-test. This allows assessing even small design choices for their impact on performance. Tests C and D also demonstrated a possible further application of performance prototyping – setting performance targets for each step of the intended workflows based on the expected impact on the entire system. This “budgeting” of resources and planning of performance could help to coordinate the viewpoints of owner, designer, developer and operator of a distributed system.

OUTLOOK AND CONCLUSION

In this paper we presented the ideas and principles behind performance prototyping as well as the integration and deployment concepts. In our example, we showed a basic web-application and demonstrated, how a simple and fast evaluation of a performance prototype could be used to assess the suitability of different design variants.

We are currently working to expand the method to include further platforms (e.g. EJB, ASP, .NET) and protocols (e.g. SOAP, RMI) to support prototypes of more heterogeneous systems.

In our view, the major obstacles in the way of widespread application of software performance engineering are

- a) insufficient familiarity of SW-developers with SPE methods,
- b) separate, potentially inconsistent and contradicting notation and interpretation of SPE and SW-models and
- c) the large time and effort needed to obtain results.

We expect that by overcoming these obstacles, the concepts and methods of SPE would bring a large benefit into SW-engineering. SPE could then contribute more than currently towards SW-products and systems that have better performance at lower development costs and shorter development time. With our UML-based notation we aim to contribute towards a more intuitive modelling of performance aspects in standard UML in mainstream tools. Our works around simulation [Hennig02] and performance prototyping as SPE methods show the flexibility and wide range of the notation. The possibility to evaluate different scenarios fast, consistently and efficiently allows for close interaction of predictive methods like simulation, benchmarking, load testing and performance prototyping. Simulation will be invaluable for extrapolation in the dimensions of scalability, reliability and optimisation. Benchmarking can provide basic measures; performance prototyping can assess specific infrastructures for specific load scenarios. Projecting the findings onto larger planned server farms or networks could again be the contribution of simulation based on the parameters and findings obtained through multi-varied performance prototyping.

At the ESM 2003 we will give a presentation of the integrated end-to-end process of performance prototyping.

REFERENCES

- [Dimitrov02] Dimitrov E., Schmietendorf A., Dumke, R.: “*UML-based Performance Engineering Possibilities and Techniques*”, IEEE Software, Darmstadt, p. 74-83, Vol 19/1, Jan/Feb 2002
- [Hennig02] Hennig, A. R. Wasgint; “Performance Modeling of Software Systems in UML-Tools for the Software Developer”, in *Proceedings of European Simulation Multiconference ESM’2002*, Darmstadt, Germany, 2002
- [Hennig01] Hennig, A.; Eckardt, H., 2001, “*Challenges for Simulation of Systems in Software Performance Engineering*”, in Proc. ESM’01, Prague, p. 121-126, 2001
- [Mirandola00] Mirandola, R., Cortellessa, V.; 2000, “*UML Based Performance Modeling of Distributed Systems*”, UML 2000 - 3rd Int. Conf., York, UK, 178-193, 2000
- [Petriu99] Petriu, D., Wang, X. „*From UML descriptions of High-Level Software Architectures to LQN Performance Models*“, Proc AGVTIVE’99, Springer Verlag, LNCS 1779, p47-62, 1999
- [RFC1738]: Berners-Lee, T., Masinter, L., and M. McCahill, Editors, "Uniform Resource Locators (URL)", December 1994.
- [RFC1808]: Fielding, R., "Relative Uniform Resource Locators", June 1995.
- [Smith90] Smith, C.U., 1990. “*Performance Engineering of Software Systems*”, ISBN 0-201-53769-9, Addison-Wesley, Reading, US, 1990
- Togethersoft Corporation,
<http://www.togethersoft.com/>
- Mercury Interactive Corporation,
<http://www.mercuryinteractive.com/>
- [Williams98] Williams, L.G., Smith C.U., “*Performance Evaluation of Software Architectures*”, WOSP 1998, p 164.177, 1998

From UML to Performance Measures - Simulative Performance Predictions of IT-Systems using the JBoss Application Server with OMNET++

Andreas Hennig, Dean Reville and Michael Pönitsch
Siemens AG, Corporate Technology, CT SE 1,
Otto-Hahn-Ring 6, 81739 München, Germany,

Andreas.Hennig@siemens.com, dean_ntu@hotmail.com, Michael.Poenitsch@siemens.com,

KEYWORDS

Software Performance Engineering, UML Modelling, Software Development, Performance Annotation, Simulation, JBoss, J2EE, LoadTest

ABSTRACT

In this paper, we argue the case for thorough performance engineering already in the early development phases of complex IT-systems, particularly web-based ones on the example of the Open Source Application Server JBoss. We show the need for a fast and efficient modelling of web-architectures, shortly recall a proposed UML notation and conversion framework [Hennig02], report progress of our end-to-end integration of simulation into a commercial UML-Tool and demonstrate its benefit on the example of a JBoss-based application. The abstractions chosen for the JBoss EJB Application Server model in the OMNET++ Simulator are described and the predicted performance is compared to values observed in load tests on the finished system.

INTRODUCTION

Most of today's complex IT-systems consist at least in parts of web-technologies – increasingly even in their core functionalities, not merely the user front end. Apart from the reduced number of core standards and mechanism (e.g. http, SOAP, WebServices, J2EE) as well as the (hopefully) improved interoperability, it is the wide availability of server software (commercial and public domain), engineering tools and development know-how that advances their use. Also the fast pace of developing the initial (typically highly presentable) increments is in favour of the web, but often conceals the long way to complex distributed systems with good performance.

For the single and local user that typically develops and tests the system, everything works smoothly and transparently, but the “going live” almost immediately exposes them to a large, wide and highly critical audience: the www-users and the competitor's web-sites “one click away”. Unfortunately, typical failures in distributed systems do not arise during the test of individual components by individual users but occur during integration, system and deployment test with many

concurrent users and transactions. Unless located and corrected in time they may result in cost overruns and missed deadlines. The prediction of performance properties of software systems (e.g. capacity, speed, stability, reliability, scalability), particularly distributed ones like web-applications, is therefore vital for substantiated design decisions at an early stage. Since these systems are often business critical and/or highly image critical, insufficient performance can incur heavy costs e.g. compensation, penalties, superfluous hardware, sub-optimal decisions based out-dated information, loss of market shares...

Simulation as a method of Software Performance Engineering (SPE, [Smith90]) has a long and successful track record of early and reliable performance predictions, which assess suitability of design choices and thus helps to ascertain time, cost and function targets. However, even if known, SPE methods often fail to keep up with the fast pace of development, mostly through high modelling efforts, but also through their high communication needs between the developer (or designer) and the modelling expert.

Based on the work of our group within Siemens Corporate Technology, which offers SPE consulting to the Siemens business units we are convinced that simulation makes valuable contributions, but needs simplified and more intuitive modelling in a “native language” of the developers and powerful and flexible pre-modelled abstractions of common infrastructure components and protocols to be sufficiently fast, trustworthy and effective. The currently most widespread “native” language of SW-developer would be the UML. One such component would be an J2EE application server, a central and common infrastructure element of current web-architectures.

In the following sections, we will recall the main elements of the UML-notation used; the one-click integration approach of the simulation into the UML-Tool before describing the application of the notation to an EJB application server and the JBoss model developed for OMNET++. We then compare the predications of a simple EJB system from the simulation to values measured in load-tests on real prototype systems before concluding the paper.

UML PERFORMANCE ANNOTATIONS

UML has established itself as the "native" modelling language of SW-development – despite all its shortcomings and ambiguities. It is therefore not surprising, that we among many others advocates of SPE (e.g. [Mirandola00], [Klein96], [Dimitrov00], [Xu03]) favour UML for SPE modelling as the most suitable way to integrate SPE methods and software engineering. However, we regard the SPE annotations as the “guests” in the “production model” and therefore formulated strict requirements to ensure acceptance of SPE annotations [Hennig01]: one single model, the same UML tool (commercial off-the-shelf, even if incomplete in support of UML standard), same interpretation of UML elements, same diagram types, no (or very few) additional diagrams, no interference with forward/reverse engineering, no SPE artefacts in generated code, no manual conversion steps...

This means in particular the use of sequence diagrams instead of state diagrams for representation of behaviour and resource consumption, since sequence diagrams are by experience not only available far earlier at less effort and require less detail; they are also more intuitive to developers, analysts and project owners. Using sequence diagrams directly without manual transformation ensures the consistency of team-model and SPE-model. The lack in precision (of modelling all possible behaviours) is more than compensated by the fact that it is modelled for SPE evaluation - even if only “typical” behaviours are modelled rather than “all possible”.

Use-case diagrams are used to aggregate behaviour and allow for usage variation through parameterisation. Actors in Sequence and Use-Case diagrams represent the load onto the system. Deployment diagrams are used as the obvious choice (like in [Williams98], [Dimitrov02], [Mirandola00], [Petriu99], unlike e.g. [Arief00]) to model HW and SW entities as well as their available resources and connections. UML-multiplicities model repeated occurrences of identical entities like servers in a server farm or a pool of clients. Performance measures and requirements can be modelled in sequence diagram, where the observed values can be recorded or requirement violations can trigger alerts.

Of the many possible diagrams in a production model, the experiment setup defines which deployment, use-case and sequence diagrams should be evaluated, together with setting parameter values to allow for variants. This way, one single production model can contain various experiments (“what happens in infrastructure scenario a, b, c; which user behaviour x, y, z”) or of abstraction levels (“strata”) in a consistent and non-interfering manner. Although creating special SPE-models for each experiment would separate individual inquiries more clearly, it would render consistency over various fast-paced iterations virtually impossible.

SIMULATOR INTEGRATION INTO UML-TOOLS

In order to achieve a transparent end-to-end process for the user, we devised and implemented the following integration concept (Fig. 1). In the UML-CASE tool (currently TogetherJ 4.2 [togethersoft]), the simulation cycle is initiated (“one-click”), for which the required diagrams are obtained and compiled into a XML document describing the entire experiment. The experiment description is complemented by optional default settings and information on available library modules like JBoss. The “converter” generates a simulation definition file from the experiment, which is then compiled into the network and behaviour parts of the simulator. The simulator is based on the freely available discrete event simulator OMNeT++ [Varga01], [Varga97] and contains the relevant core modules and specific SPE extensions (scheduler, workflow execution engine...) as well as pre-modelled modules like the representation of JBoss (see next sections). The statistics of performance observations collected during the execution of the simulator are compiled and fed back textually into dedicated tagged values in the UML-model (if requested). The entire cycle is controlled by a set of platform-independent scripts, which could also be used to run an entire series of experiments.

Instead of producing merely NED for OMNeT++ as simulation definition file, it is also possible for the converter to be expanded to produce definition files for different evaluation techniques, e.g. Queuing networks, Bottleneck Analysis [Eckard01] or Performance Prototyping [Hennig03].

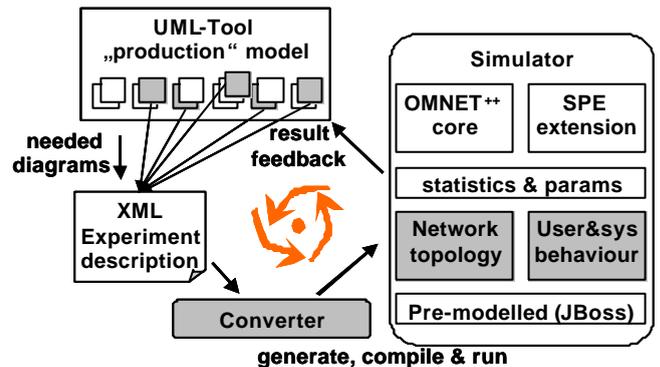


Fig. 1: End-to-end simulation cycle from UML-tool

In order to keep the simulation cycle short, we decided against using XMI [XMI02] directly instead of the XML experiment description, since the export and subsequent parsing and traversal of an entire production model with many unused diagrams would be too resource-intensive. In the future, we intend to use a programmatic interface (based on the Meta Object Facility MOF [MOF02]), which allows to query and update specifically identified elements of the model according to the MOF metamodel without having to go through its XMI representation.

MODELLING THE SAMPLE JBOSS APPLICATION

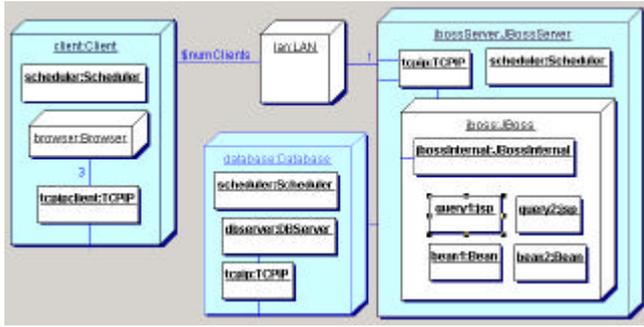


Fig. 3: EJB network model in UML deployment diagram

The infrastructure and network of the example application is modelled in the UML deployment diagram depicted in Fig. 3. It contains (on the right) the machine hosting the JBoss server with abstractions of a scheduler as provider of the resource “CPU”, a TCP/IP communication protocol stack and the JBoss server itself. The business logic modules, i.e. the JSPs and beans, are placed inside the JBoss server. The specific type of beans is specified by UML attributes (tagged values) of the beans: beantype (session or entity), statefulness (stateful and stateless) and persistence (container- or bean-managed). To avoid cluttering of the diagram by the internals of the JBoss (e.g. the servlet engine and containers for the different types of beans with their numerous interceptors), the UML diagram contains a module “JBossInternal” representing the internals. The diagram also shows the physical connections between the hosts and network nodes, as well as multiplicities indicating e.g. how many web browsers (3) are running per client host (\$numClient).

The intended behaviour of the application is modelled with the sequence diagram shown in Fig.4, where the actor “webUser” represents the load of one or more concurrent users with prescribed load pattern (e.g. gradual ramp-up or continuous, interarrival delay, think-times). Upon a single “click” the browser submits three http requests to JSPs query1 and query2, which serve as a front end to the beans. On one occasion, bean1 consults the database before sending the response back to the browser. We also

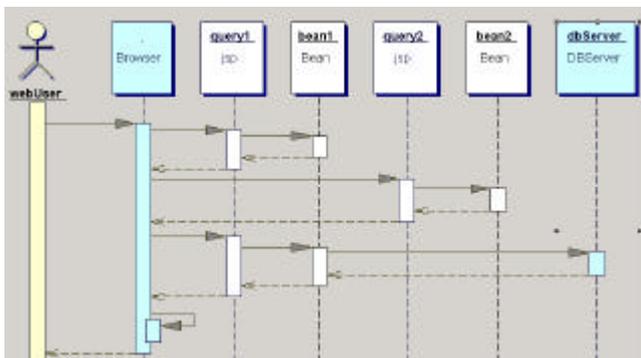


Fig. 4: EJB behavioural model in UML sequence diagram

specified how much CPU-time each step consumes. CPU-

consumption is specified either as CPU-seconds or in iterations of the core operation of standard or application-specific benchmarks like whetstone or heapsort. This allows simple scalability investigations, by adjusting the amount of provided CPU-resources in the scheduler. Requirements are specified in the sequence diagram as well, e.g. the combined start-to-finish time the three browser requests should be below a given threshold.

After modelling infrastructure, load, behaviour and requirements in various UML diagrams, an experiment definition diagram is used to specify the subset of “investigated diagrams” and overall parameters like a scalability factors for number of clients (e.g. \$numClients in Fig. 2), the workload or overall think time.

JBOSS SIMULATION MODULES IN OMNET++

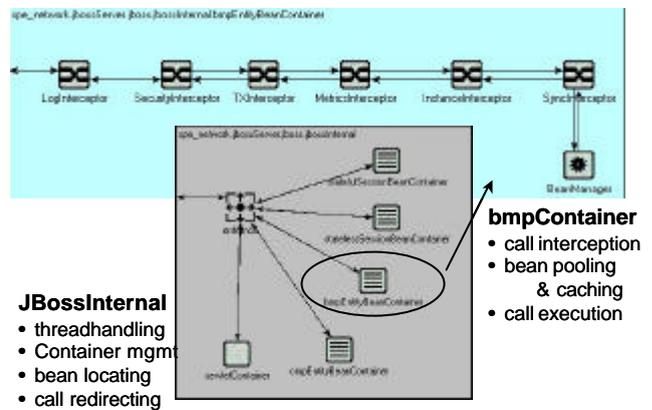


Fig. 5: JBoss internal modules in OMNET++

The option to use pre-modelled simulation modules allows us to develop complex infrastructure components separate from the UML-models of the target application. This helps avoiding unnecessary detail and clutter in the UML-diagrams, but also provides the means to build model libraries in the native programming environment of the simulation engine (C++ for OMNET++). It also enables us to use models from the growing OMNET++ user community to build larger scenarios more quickly. Typical examples for pre-modelled modules are communication protocols like TCP and infrastructure components like JBoss. Fig. 5 therefore shows the sub-modules of JBossInternal in terms of OMNET++. The “user” of the JBoss models would only use the more familiar UML representations in Fig 3 and 4.

The module “JBossInternal“ is responsible for modelling the thread-handling of the JBoss and for the overall management of the bean containers. When a request to a bean arrives at JBossInternal, the target container is located based on the type of bean (session or entity, container- or bean-managed persistence) or servlet. The call is then redirected towards the appropriate container, in Fig. 5 this is a “bmpContainer” for entity beans with bean-managed persistence.

Inside the container, the call traverses a series of configurable interceptors (e.g. for logging, security,

transaction processing) before arriving at the beanManager. The beanManager obtains the requested bean instance either from a pool or cache of beans, where the call will be processed. At this time, the information of the sequence diagrams is consulted to obtain the prescribed behaviour like resource consumption, timestamp collection and requirement evaluation. If nested calls are required (like the database request from bean1 in Fig 4.) they are executed according to condition and iteration specification given in UML. One of the most challenging aspects of the JBoss model was the fact that the entire sequence of containers, interceptors and managers is executed within the same java thread – which could clearly be seen in traces generated from a simple manual prototype application. This meant that all simulation modules had to reuse a specific existing thread to prevent loss of simulation accuracy due to excessive context switches.

EXPERIMENTS AND RESULTS

For experimental evaluation, we used the above simple behaviour and varied the load on the system through the number of simulated users. The load was increased every 2 minutes by one additional simulated user until approximately 10 minutes after a clear saturation of the system had been reached. In the sequence diagram, the call to bean2 was specified to be resource intensive (the equivalent of performing 3500 heapsorts on arrays of 1000 floating point values) e.g. for analysing user authorization, the calls to bean1 are less demanding (1500 heapsorts). The think time was 5 seconds on average, which represents the time a user would need before the first click in the browser.

In order to compare the simulation results with real installations of JBoss, we built a prototype of above specifications and deployed it onto two different hosts. Both hosts run under Linux, dax is a dual-processor server, ibex a single-processor workstation. For reasons of simplicity, scaling of the simulation model to the reference hosts was done through adjustment of the CPU-capacity of the server in UML diagram only.

For the time being, this ignores other influences like network

bandwidth, i/o speed and latencies, which could distort findings significantly.

Unfortunately, we experience a software incompatibility, which forces us to upgrade underlying parts of the system. Rather than presenting misleading low-quality data, we give an overview on the type of data and investigation we will present. For the final paper / camera-ready copy to be submitted for the ESM, we will obtain additional measurements and run further experiments to calibrate the model more precisely and then verify the accuracy of the prediction on a larger sequence of interactions and further examples. We apologize the inconvenience.

Taking the prototype measures on dax as an example, the system went into saturation (ref. markers in Fig. 6) after 60% of the experiment time with an approximate capacity of 1.9 TPS, which was caused by 23 concurrent simulated users. Further users did not increase the transaction rate but only resulted in increased response times due to shared use of the CPU resources. The response time for the first user (on a basically idle system) was 0.2s without the prescribed think time, but rose to 1.1s at the saturation point of the system (ref Fig. 7). Table 1 summarily lists the achieved or predicted rate of fully completed transactions per second (TPS, number of finished workflow instances per second).

		Capacity		Response time	
		TPS	Users	“idle”	saturate d
simulation	ibex				
	dax				
prototype	Ibex	0.6	10	1.0s	4.6s
	dax	1.9	23	0.2s	1.1s

Table 1: Comparison of simulation and prototype results

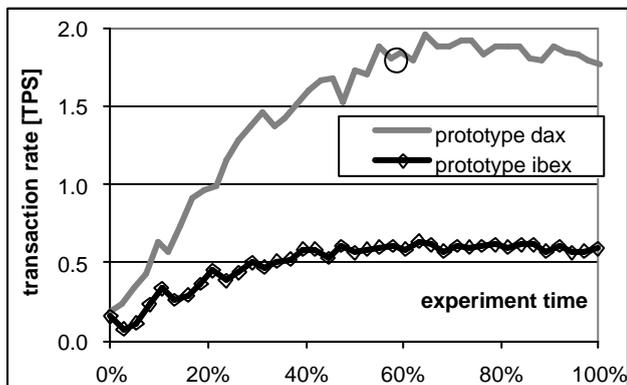


Fig. 6: Transaction rates of simulation and prototype

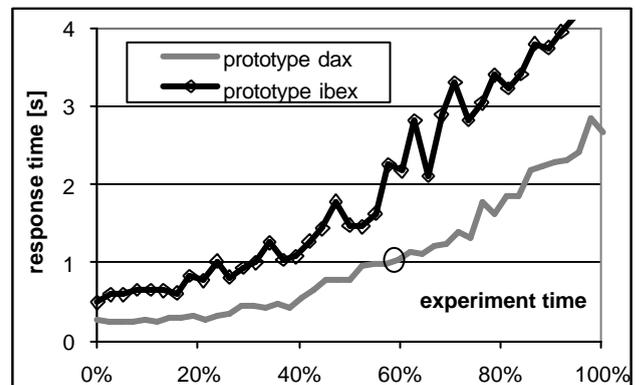


Fig. 7: Response times of simulation and prototype

OUTLOOK AND CONCLUSION

The simulation model presented in this paper provides a performance engineer with a simple and intuitive way of noting infrastructure, load, behaviour and requirements of a distributed system, which uses a JBoss application server. By splitting the model into an application (UML) and infrastructure level (native OMNET++), we achieve sufficient expressive and simulative potential without requiring excessive detail.

We currently work on integrating our method with models from the OMNET++ community, most importantly the TCP/IP models described in [Kaage2001]. Another aspect is the calibration of the model by efficient determination of model parameters on various infrastructures by means of benchmarking, prototyping (manual and automatic) and load testing. Particularly the possible interaction between Simulation and Prototyping is very promising.

In [Hennig03], we present a way to use the notation, conversion methodology and UML-models described here to automatically generate and deploy performance prototypes. The efficient combination of corresponding prototypes and simulation models opens new opportunities to predict system performance faster and more accurately with affordable effort and time.

REFERENCES

- [Arief00] Arief, L.B., Speirs, N.A. "A UML Tool for an Automatic Generation of Simulations Programs", WOSP 2000, p 71-76, 2000
- [Dimitrov02] Dimitrov E., Schmietendorf A., Dumke, R.: "UML-based Performance Engineering Possibilities and Techniques", IEEE Software, Darmstadt, p. 74-83, Vol 19/1, Jan/Feb 2002
- [Dimitrov00] Dimitrov E., Schmietendorf A.: "UML-basiertes Performance Engineering", "in Performance Engineering in der Softwareentwicklung (PE 2000)", Darmstadt, p. 41, 2000
- [Hennig01] Hennig, A.; Eckardt, H., 2001, "Challenges for Simulation of Systems in Software Performance Engineering", in Proc. ESM'01, Prague, p. 121-126, 2001
- [Hennig02] Hennig, A. R. Wasgint; "Performance Modeling of Software Systems in UML-Tools for the Software Developer", in *Proceedings of European Simulation Multiconference ESM'2002*, Darmstadt, Germany, 2002
- [Hennig03] Hennig, A., Hentschel, A. and Tyack, J., "Performance Prototyping - Generating and Simulating a distributed IT-System from UML models" submitted to *European Simulation Multiconference ESM'2003*, Nottingham, UK, 2003
- [Kaage01] Kaage, U., Kahmann, V., Jondral, F., „An OMNet++ TCP/IP MODEL“, in Proc. ESM'01, Prague, p. 409-413, 2001
- [Klein96] Klein, M.H.: "State of Practice report: Problems in the Practice of Performance Engineering". Technical Report, Pittsburg, Pennsylvania: Software Engineering Institute, 1996
- [Mirandola00] Mirandola, R., Cortellessa, V.; 2000, "UML Based Performance Modeling of Distributed Systems", UML 2000 - 3rd Int. Conf., York, UK, 178-193, 2000
- [XMI02] OMG, XMI 1.2, 2002 , <http://www.omg.org/technology/documents/formal/xmi.htm>
- [MOF02] OMG, MOF 1.5 RTF, 2001/2002, in Revision, http://www.omg.org/techprocess/meetings/schedule/MOF_1.5_RTF.html
- [Petriu99] Petriu, D., Wang, X. „From UML descriptions of High-Level Software Architectures to LQN Performance Models“, Proc AGVTIVE'99, Springer Verlag, LNCS 1779, p47-62, 1999
- [Smith90] Smith, C.U., 1990. "Performance Engineering of Software Systems", ISBN 0-201-53769-9, Addison-Wesley, Reading, US, 1990
- [Togethersoft] Togethersoft Corporation, <http://www.togethersoft.com/>
- [Varga01] Varga A., "The OMNET++ Discrete Event Simulation System", in Proc. ESM'01, Prague, p. 319-324, 2001
- [Varga97] Varga, A. OMNET++ Homepage, <http://www.hit.bme.hu/phd/vargaa/omnetpp.htm>, 1997
- [Williams98] Williams, L.G., Smith C.U., "Performance Evaluation of Software Architectures", WOSP 1998, p 164.177, 1998
- [Xu03] Xu, Z, Lehmann, A., "Automated Generation of Queuing Network Model from UML-based Software Models with Performance Annotations", Technical Report #2002-06, Universität der Bundeswehr, München, 2002

META-MODELLING OF DATA FLOW PROCESSES WITH META-MODELLING TOOL ATOM³

ANDRIY LEVYTSKYI and EUGENE J.H. KERCKHOFFS

*Faculty of Information Technology and Systems, Mediamatica Department
Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands
Email: a.levytskyi@cs.tudelft.nl*

Abstract: this paper illustrates how meta-modelling is used to support designing and executing data flows in a web-based simulation environment (in our case the home-made so-called NCSE environment [Levytskyi and Kerckhoffs, 2000a]). Although simple, the data flow considered has a significant leverage in the real-world scenarios typical for web-based environments. Based on the definition of Data Flow Diagrams (DFD), we specify a DFD metamodel in the Entity-Relationships formalism with the meta-modelling tool ATOM³ [de Lara and Vangheluwe, 2002a] and use it to generate a visual modelling tool tailored according to the proposed DFD metamodel. Finally, the paper illustrates how a DFD model created with this modelling tool is transformed into a textual code, a job description for the NCSE execution controller.

keywords: Data Flow Diagrams, Metamodel, Transformation, Code Generation, Web Environment

1. INTRODUCTION

The emergence of the world-wide web (WWW) and its popularity in the simulation community gave birth to the concept of *web-based simulation* [Fishwick, 1996], which now includes (among others) activities that deal with the use of the WWW as infrastructure to support distributed simulation execution and encompass research in tools, environments and frameworks that support the distributed, collaborative design and development of simulation models [Page, 1998].

Within this domain, several years ago we started a Collaborative Simulations project in which a generic web environment is developed to support simulation and modelling components in multidisciplinary collaborative projects [Levytskyi and Kerckhoffs, 2000a]. The environment's functionality is similar to that of the DLR-IMF Virtual Laboratory [DLR-IMF]. The practical application of our prototyped environment lies in the so-called NanoComp project, which investigates computing systems based on quantum devices; therefore the environment is named NanoComp Simulation Environment (NCSE).

NCSE is *based* on two major types of remote objects called *resources* [Levytskyi and Kerckhoffs, 2001]: conventional tools and models, which are maintained by the collaborative groups that own them. The environment *provides*: (i) an infrastructure that connects remote resources to their respective web-façades (proxy objects accessible from the web) via a distributed object middleware; (ii) centralised access control (via a controller) to remote resources; and (iii) on-line

services, such as registration, discovery and processing of resources (i.e. simulation of a registered model with an integrated simulation tool). These web-façades are containers for metadata that describe properties of the remote counterpart tools and models, thus enabling the above-mentioned services. Since 2002, NCSE includes meta-modelling capabilities with the assistance of ATOM³ (A Tool for Multi-formalism and Meta-Modelling).

ATOM³ is a visual tool for meta-modelling and model-transforming. Meta-modelling refers to modelling formalism concepts at a meta-level, and model-transforming refers to automatic converting, translating or modifying a model of a given formalism into another model of the same or different formalism [Vangheluwe et al, 2002]. The tool's meta-layer allows a high-level description of models, based on which ATOM³ can automatically generate a tool tailored to the family of those models.

In NCSE, ATOM³ is used as Meta-CASE Tool (and the topic of this paper is an example of such a use) to develop meta-models for various formalisms supported by the environment. Given these metamodels, ATOM³ can be used as a conventional modelling tool for the supported formalisms. Finally, we employ the model-transforming capabilities (a) to generate job descriptions for the NCSE controller (which is discussed in this paper) and (b) given a formalism's metamodel, to synthesize code for the formalism's components of the NCSE environment.

In this paper we illustrate how meta-modelling is used to support designing and executing data flows in the NCSE environment. Although simple, the data flow considered has a significant leverage in the real-world scenarios typical for web-based environments. In section 2 we provide a definition of Data Flow Diagrams that will serve as specification for the DFD metamodel presented in section 3. Based on this metamodel, AToM³ can generate a completely new DFD modelling tool. Section 4 describes how to construct a transformation that, given a DFD model, generates a respective textual code for an external solver: the NCSE controller. An example of model-transforming is given in section 5. We conclude the paper with final remarks.

2. DFD DEFINITION

Data Flow Diagrams (DFD) present the flow of data through a system [Gane and Sarson, 1979]. The focus is on how data is processed by a system in terms of inputs and outputs. The building constructs of Data Flow Diagrams are *Data Flow*, *Data Store*, *Process* and *External Entity*. Figure 1 shows their respective graphical notations as proposed in [Gane and Sarson, 1979].

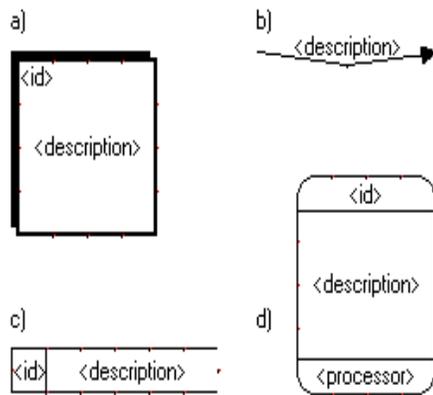


Figure 1: DFD notation.

External entities (Figure 1a) are data objects outside the context of the modeled system. External entities are sources and sinks (destinations) of the system's inputs and outputs. Each is given an alphabetic identifier.

Data flow (Figure 1b) is a pipeline through which packets of data of known composition flow. The arrowhead indicates the direction of the data flow. Each data flow must have a label describing the data.

Data stores (Figure 1c) are repositories of data inside a system. It is a data queue as opposed to

data flow. Each is identified by "D" and an arbitrary number.

Process (Figure 1d) transforms an incoming data flow into an outgoing data flow. Each is given a numerical identifier, physical reference (in the lower part of the process box) and is described with an imperative sentence containing an active verb e.g. "CONVERT data".

Additionally, there are general rules that a valid DFD diagram should comply with. Some of them are:

- Data flow connects other DFD constructs.
- No alteration of data can take place within a data flow.
- An external entity cannot be connected to another external entity.
- Data stores receive inputs and outputs only from processes.

There is much more to say about DFD (levels and types of Data Flow Diagrams, more rules and recommendations), but the definition provided here is sufficient for our purposes.

3. METAMODEL

A metamodel of a given formalism specifies the syntax aspect of the formalism by defining the language constructs and how they are built-up in terms of other constructs.

To construct a DFD metamodel we used Entity Relationships (ER) diagrams extended with constraints, a default meta-formalism of AToM³. Constraints provide a view on how a construct can be connected to another construct to be meaningful, and thus specify static semantics of the formalism. In this paper constraints are expressed in Object Constraint Language [OCL, 1997].

Important properties of each construct are *Cardinality*, *Attributes*, *Constraints*, and *Appearance*. Cardinality determines the possible number of incoming and outgoing connections of a construct. Additionally, we employ Constrains to control what constructs can connect to what constructs. We populate each construct's Attributes property (of collection type) with a minimum set of regular attributes that supports the semantics of the construct alone and in combination with other constructs. Finally, we define the Appearance property of each construct in accordance with the notation presented in Figure 1.

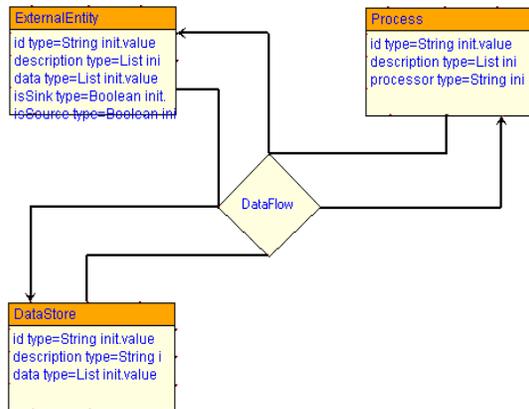


Figure 2: DFD Metamodel.

The metamodel in Figure 2 was constructed in AToM³ according to the DFD definition provided above and shows how DFD constructs can be combined together. In the following, we describe each element of the metamodel in more details:

EXTERNALENTITY:

Cardinality:

self-to-dataflow: (1: 0..*)
 dataflow-to-self: (1: 1)

Attributes:

id: string = 'a'
 description: string
 data: sequence
 isSource: boolean
 isSink: boolean

DATASTORE:

Cardinality:

self-to-dataflow: (1:1..*)
 dataflow-to-self: (1:1)

Attributes:

id: string = 'D'
 description: string
 data: sequence

DATAFLOW:

Cardinality:

self-to-destination: (1:1)
 source-to-self: (1:1)

Attributes:

description: string
 data: sequence

Constraints:

```

DataFlow ::= CONNECT(...)
post: self.Source.metaclass ->
  forAll(s |
    self.Destination.metaclass ->
      forAll(d |
        not s = d =
          'ExternalEntity'))
post: self.Source.metaclass ->
  forAll(s |
    self.Destination.metaclass ->
      forAll(d |
        Set{s,d}
        Set{"Process", "DataStore"}))
  
```

PROCESS:

Cardinality:

self-to-dataflow: (1:1..*)
 dataflow-to-self: (1:1)

Attributes:

id: string
 description: string
 processor: string

Along with the properties defined for each DFD construct, we also extend the global properties for the metamodel itself with attributes, such as *title*, *subject*, *description*, *author* and *version*. They can be used for basic documentation of models specified in this DFD formalism.

All global properties and regular attributes are to be filled-in by the end-user of the DFD modeling tool to be generated at the lower meta-level.

Finally, the flexibility and elegance of the meta-modeling concept allows us to easily adapt the DFD formalism as defined in section 2 to our needs. For example, to match the capabilities of the controller, we introduce two new general rules for our DFD:

- Do not allow branching.
- Do not allow loops on the same Process.

AToM³ allows preventing branching by tuning the Cardinality property of elements. The loops are avoided by the following constraint:

```

DataFlow ::= CONNECT(...)
post: self.Source -> forAll(s |
  self.Destination -> forAll(d |
    s <> d implies s.id <> d.id))
  
```

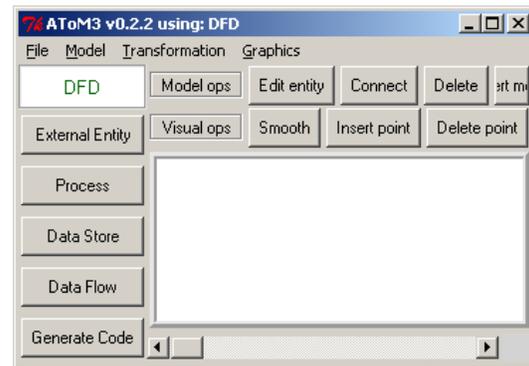


Figure 3: Generated DFD modeling tool.

Given our metamodel, we can now generate in AToM³ a meta-specification, which, when loaded into the meta-level of AToM³, turns it into a new modeling environment for the modeled DFD formalism. A part of this meta-specification is a specification of the User Interface. This specification is a model in its own right and can be edited in AToM³ at any time under a so-called "Buttons" formalism. By default, this specification creates a button for every construct of the formalism. In addition, we created one extra button, which on click applies the code generation transformation to the model on the tool's canvas. An instance of the generated DFD modeling tool is shown in Figure 3.

4. CODE GENERATION TRANSFORMATION

Model transformation is related to dynamic semantics of a formalism, which defines the meaning of well-formed constructs. This meaning can be described in a number of ways, e.g.: formalism transformation, model optimization, code generation and simulator specification.

This section describes a code generation transformation that, given a DFD model, generates a corresponding textual job description for the NCSE controller. The controller is a custom built Process-Interaction (PI) solver based on the operational semantics of π Demos [Birtwistle and Tofts, 1994].

In AToM³ model transformations are specified through Graph Grammars, and consist of *Initial Action*, *Final Action* and *Transformation rules*. Each rule consists of *Left Hand Side* (LHS) and *Right Hand Side* (RHS) graphs, and *Condition*, *Action* and *Priority* properties.

The *Initial Action* of the transformation iterates through all the elements of the current model (objects on the tool's canvas) to augment them with temporary attributes to be used in the conditions specified below. Attribute *isVisited* helps to distinguish the elements that have been already processed from those that have not yet. Attribute *isCurrent* is used to mark a DataFlow that leads to the element whose code has to be generated next. It also creates the job data structure:

```
{'source': '', 'sink': '', 'body': [ ]}
```

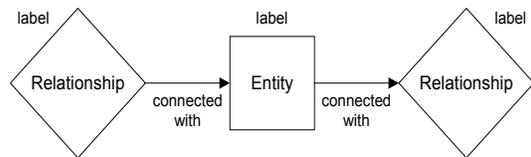


Figure 4: Subgraph match pattern.

We designed the rules to match a pattern shown in Figure 4, where the relationship element is a DataFlow, and the entity can be an instance of any other DFD component. Either the left or right relationship can be omitted. Present elements are labelled with consequent numbers. In the following we briefly describe each rule:

RULEPROCESS (priority 1) locates a Process and rewrites the model as shown in Figure 5. Its *action* generates code using proper controller commands (**get**, **hold**, **put**) to access, use and release the physical entity implementing the process, and marks element 1 as not current, element 2 as visited, and element 3 as current.

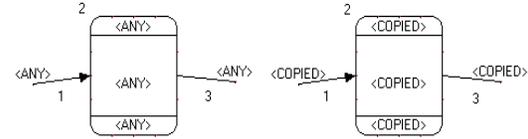


Figure 5: LHS and RHS subgraphs for processes.

```
Action
pre: LHS.element1.isCurrent = 1
     and LHS.element2.isVisited = 0
post: RHS.element1.isCurrent = 0
     and RHS.element2.isVisited = 1
     and RHS.element3.isCurrent = 1
```

RULESOURCEEXTERNAL (priority 2) locates a source ExtEntity and rewrites the model as shown in Figure 6. Its *action* updates the 'source' field of the job description with the URL value of the data object of element 1 and marks element 1 as visited and element 2 as current.

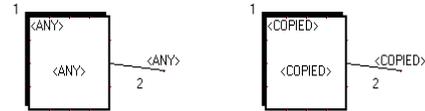


Figure 6: LHS and RHS subgraphs for source externals.

```
Action
pre: LHS.element1.isSource = 1
     and LHS.element1.isVisited = 0
post: RHS.element1.isVisited = 1
     and RHS.element2.isCurrent = 1
```

RULEDATASTORE (priority 3) locates a DataStore and rewrites the model as shown in Figure 7. As semantics of this entity in the controller's context is currently not defined, the *action* only marks element 1 as not current, element 2 as visited, and element 3 as current.

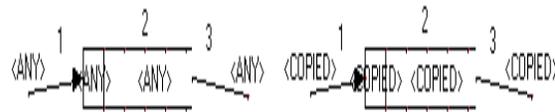


Figure 7: LHS and RHS subgraphs for data sources.

```
Action
pre: LHS.element1.isCurrent = 1
     and LHS.element2.isVisited = 0
post: RHS.element1.isCurrent = 0
     and RHS.element2.isVisited = 1
     and RHS.element3.isCurrent = 1
```

RULESINKEXTERNAL (priority 4) locates a sink ExtEntity and rewrites the model as shown in Figure 8. Its *action* updates the 'sink' field of the job description with the URL value of the data object of element 2 and marks element 1 as not current and element 2 as visited.

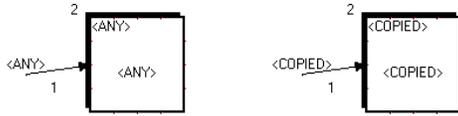


Figure 8: LHS and RHS subgraphs for sink externals.

Action
pre: LHS.element1.isCurrent = 1
 and LHS.element2.isSink = 1
 and
 LHS.element2.isVisited = 0
post: RHS.element1.isCurrent = 0
 and RHS.element2.isVisited = 1

The *Final Action* prints the job data structure into an output file. As the last step, it iterates through all the elements on the tool's canvas removing temporary attributes *isVisited* and *isCurrent*.

5. MODEL-TRANSFORMING

Figure 10 shows a DFD model created with the generated modeling tool.

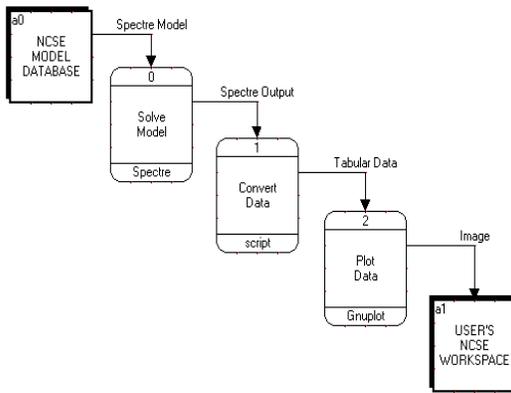


Figure 9: A model in the DFD formalism.

Source external entity **a0** contains a reference to a model registered in the NCSE model base. Process **0** refers to a simulation tool that can solve the model concerned. Process **1** is a script that converts the output of process **0** into the input for process **2**. Process **2** refers to a visualization application that produces diagrams from the input data. Finally, sink external entity **a1** refers to the modeler's workspace at NCSE.

Model-transforming in AToM³ can be launched in a variety of ways, e.g. by clicking the button, which we created in the graphical user interface for the code generation transformation.

During execution of a model transformation, AToM³'s Graph Rewriting Processor (GRP) iterates through the list of rules sorted by their priority in an ascending order and tries to apply the current rule to the model. If the rule makes a match (LHS pattern is found and conditions are met), it is

executed and the GRP repeats trying each rule again from the beginning of the list. This continues until there are no rules anymore that can be applied, then GRP considers the model transformation as completed [de Lara and Vangheluwe, 2002].

The result of our model-transforming is a valid job description for the controller (see Figure 10).

```
#
# This code is automatically generated.
#

__version__ = 'Revision: 0.01 $'[11:-2]
__author__ = 'A. Levytsky'

# A job description for NCSE controller
job = {
  'body': ["getR ('Spectre')",
          'hold ()',
          "putR ('Spectre')",
          "getR ('script')",
          'hold ()',
          "putR ('script')",
          "getR ('Gnuplot')",
          'hold ()',
          "putR ('Gnuplot')",
          "close()"
          ],
  'source': 'scheme://host:port/sourcepath',
  'sink' : 'scheme://host:port/sinkpath'
}
```

Figure 10: Generated textual code for execution.

At this point the synthesized code can be passed to the NCSE controller for execution. The controller will create a new job (and add it to the pool of already existing jobs) that will provide the data as input for process **0**, and so on until the output of process **2** is placed in the environment's cash and the output's URL is stored in the user's workspace. More details on the controller and job execution can be found in [Levytsky and Kerckhoffs, 2000b].

6. FINAL REMARKS

In this paper we demonstrate how the concept of meta-modelling could be used to easily extend an existing simulation environment with new functionality, namely dataflow modelling and execution. The meta-modelling tool AToM³ plays an important role in this (even though currently AToM³ can only be used locally and not from the web) and is primarily used as Meta-CASE Tool to develop meta-models for various concepts used in the environment, and as code generator. The most important is that meta-modelling and AToM³ indeed enable us to adjust NCSE to different situations.

ACKNOWLEDGEMENT

The research reported in this paper is done in the framework of the NanoComp project, sponsored by TU-Delft.

We would like to thank the Modelling, Simulation and Design Lab (MSDL) of the School of Computer Science of McGill University (Montreal, Canada), and especially Hans Vangheluwe and Juan de Lara, for providing and helping us with AToM³.

REFERENCES

- Birtwistle G. and Tofts C. 1994, An operational semantics of process-oriented simulation languages: Part 1 pDemos. *Trans. Soc. Comput. Simul.*, 10(4), Dec. 1994, pp. 299-333.
- de Lara J. and Vangheluwe H. 2002, "AToM3: A Tool for Multi-Formalism Modelling and Meta-Modelling". In: *European Conferences on Theory And Practice of Software Engineering ETAPS02, Fundamental Approaches to Software Engineering (FASE)*. Lecture Notes in Computer Science 2306, Springer-Verlag, pp. 174 - 188.
- DLR-IMF. Virtual Laboratory, a repository of online-executable scientific software: http://vl.nz.dlr.de/VL/S_qb4Mm21B/portal/
- Fishwick P.A. 1996, "Web-Based Simulation". In: *Proceedings of the 1996 Winter Simulation Conference*, pp. 772 – 779.
- Gane C. and Sarson T. 1979, *Structured Systems Analysis: Tools and Techniques*. Prentice-Hall, Englewood Cliffs, USA
- Levytskyy A. and Kerckhoffs E.J.H. 2000a, "Towards a Prototype Web-Based Collaborative Simulation Environment", SCS: paper of the 5th Euromedia Conference, May 2000, pp. 60 – 66.
- Levytskyy A. and Kerckhoffs E.J.H. 2000b, "A simulation-based controller for a distributed collaborative environment". In: *D.F. Moeller (ed.): Simulation in Industry, Proceedings of ESS2000 (12th European Simulation Symposium, Hamburg, Germany, September 28-30, 2000)*, pp. 88-95.
- Levytskyy A. and Kerckhoffs E.J.H. 2001, "Integration of Simulation Tools and Models in a Collaborative Environment". In: *Proceedings of 2001 European Simulation Interoperability Workshop* (London, UK, June), Simulation Interoperability Standards Organisation, pp. 407-415.
- OCL (1997) Object Constraint Language Specification, version 1.1, September 1.
- Page E. H. 1998, "The rise of Web-based simulation: implications for the high level architecture". In: *Proceedings of 1998 conference on Winter simulation* (Washington, D.C., United States), pp. 1663 – 1668.
- Vangheluwe H., de Lara J. and Mosterman P.J. 2002, "An introduction to multi-paradigm modelling and simulation". In: *Proceedings of the AIS'2002 Conference (AI, Simulation and Planning in High Autonomy Systems)*, Lisboa, Portugal, April 2002, pp. 9 - 20

AUTHORS' BIOGRAPHIES



Andriy Levytskyy graduated from Chernivtsi State University, Ukraine and holds an MSc-degree in Computer Science. Currently, he is a PhD student at Delft University of Technology, Faculty "Information Technology and Systems", Department "Mediamatica", Group "Knowledge-based Systems".



Eugene J.H. Kerckhoffs holds an MSc-degree from Delft University of Technology (1970, Physical Engineering, thesis on analogue and hybrid computer simulation) and a PhD-degree from the University of Ghent (1986, Computer Science, thesis on parallel continuous simulation). Currently, he is an associate professor at Delft University of Technology (Faculty "Information Technology and Systems", Department "Mediamatica", Group "Knowledge-based Systems"). He was also chairholder of the SCS Chair in Simulation Sciences at the University of Ghent, Belgium.

SYSTEM DYNAMIC SIMULATING MODELLING OF DRIVING SYSTEM “ANCHOR WINDLASS DRIVEN BY ASYNCHRONOUS MOTOR” (BSVPAM)

ANTE MUNITIĆ, MARIO ORŠULIĆ, MAJA KRČUM, JOŠKO DVORNIK

Split College of Maritime Studies
University of Split
Zrinsko-frankopanska 38,
21000 Split, Croatia
e-mail: munitic@pfst.hr

Abstract

System dynamic simulating modelling is one of the most appropriate and successful scientific dynamics modelling methods of the complex, non-linear i.e. natural, technical and organisational systems. Investigation of behaviour dynamics of the ship's propulsion system as a typical example of complex, dynamic technical systems requires application of the most efficient modelling methods. The aim of this essay is to present the efficiency of application of the system-dynamic simulating modelling in investigation of behaviour dynamics of the BSVPAM propulsion system. The anchor windlass and its driving asynchronous motor shall be presented by mental - verbal, structural and mathematical computing models. The System Dynamics Models are, in essence, continuous models because the realities are presented by the set of non-linear differential equations, i.e. "equations of state". They are at the same time discrete models, because they used basic time step for counting i.e. discrete sampling DT, which value is determined in total accordance with "SAMPLING THEOREM" (Shannon and Kotelnikov). With the choice of basic time step DT it is possible to do computer modelling of continuous simulation models on digital computer, which is very suitable for education of the marine students and engineers, because they can study complex dynamics behaviour of marine systems and process.

Keywords

System Dynamics, Modelling, Asynchronous Engine, Windlass, Continuous and Discrete Simulation

1. INTRODUCTION

The System Dynamics Modelling is in essence special, i.e. "holistic" approach to the simulation of the dynamics behaviour of natural, technical and organisation systems, and it contains quantitative and qualitative Simulation Modelling of various nature realities. The concept of optimisation in System Dynamics is based on belief that the manual and iterative procedure, i.e. optimisation by the method "retry and error" can be successfully executed using heuristic optimisation algorithm, with the help of digital computer, and in complete coordination with System Dynamics Simulation Methodology. This simulation model BSVPAM is small part of scientifically macro project called: Intelligent Computer Simulation of the Model of Marine Processes.

2. SYSTEM- DYNAMIC SIMULATING SYSTEM MODELS “ANCHOR WINDLASS DRIVEN BY ASYNCHRONOUS MOTOR”

2.1. System dynamic model of the anchor windlass

Anchoring is an operation by which the ship is fixed to the ship's bottom. This is performed by the anchor arrangement consisting of: anchor, anchor chains, stoppers, chain locker and windlass. Some elements of the anchor arrangement are also used for the mooring of a ship. All ships are provided with the bow anchor arrangement and some of them with the stern anchor arrangement. Ships of less size may be provided with anchor windlasses driven by the internal combustion engines while on tankers where such drive may cause explosion windlasses are driven by steam. Windlasses on other ships are mostly driven by electric motors and recently are also hydraulically driven. Electric drive of windlasses shall be performed by: - the alternative three phase electric motor, in Leonard's connection, alternative three phase electric motor directly coupled with overlapping of two or three pairs of pole.

Anchor windlass consists of driving electric motor with stopper, reduction gear and main shaft unit laid

in solid bearings. Reduction gear where safety-sliding coupling is located includes a few pairs of front gears with associated shafts and bearings. High-speed rotating shafts are laid in rolling bearings while the main shaft is fitted in sliding bearings. Chain locker is situated in the main shaft having the belt brake and tightening drum. Claw clutch is located between chain locker and reduction gear enabling the independent operation of the tightening drum from the chain locker. Reduction gear is lubricated by oil sump while main shaft and other sliding surfaces are grease lubricated. The basic equations of anchor windlass are:

$$\frac{ds}{dt} = \omega r_0 \quad (1)$$

$$F_t = (G_1 S + G_s) g \quad (2)$$

$$F_{uz} = \varphi g A S \quad (3)$$

$$F_u = F_t - F_{uz} \quad (4)$$

$$M_{\tau} = F_u r_0 \quad (5)$$

$$S = S_0 - \int v_s dt \quad (6)$$

According to the basic equations the mental verbal model of ship anchor windlass may be developed or structural and anchor windlass course diagrams, respectively.

Table 1: Marks and mode of records in Dynamo language

Marks	Description	Dynamo language
M _v	Winch torque	MTVA
G _s	Anchor weight	GS
G ₁	Chain weight	GL
S	Chain length	SM
φ	Seawater density	GU
G	Gravity	GR
A	Anchor and chain area	POVRSR
S	Chain length	S
V _s	Anchor lifting speed	VS
F _T	Loading force	FU
F _{uz}	Buoyancy force	FZ

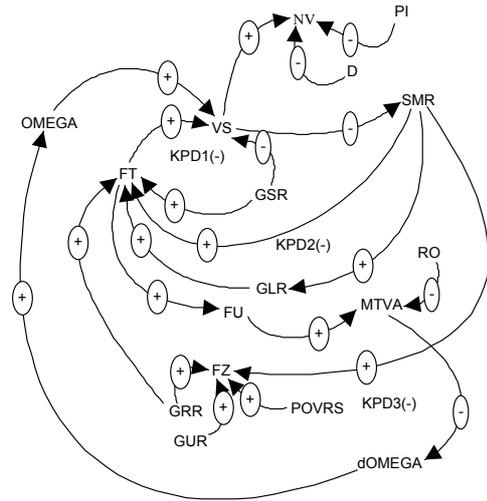


Figure 1: Structural simulation model of the anchor windlass

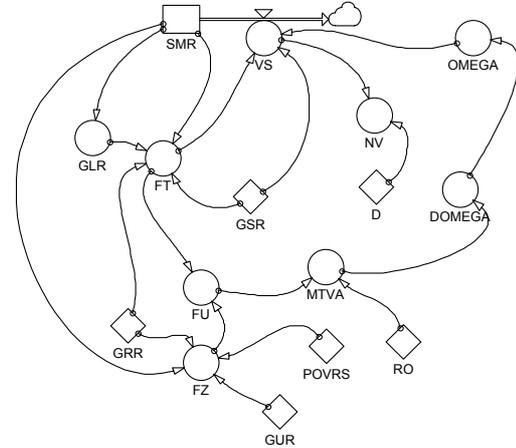


Figure 2: Structural simulation model in DYNAMO symbolic-flow diagram

Three feedback loops are present in the concerned anchor arrangement system (KPD).

KPD1 (-): VS=> (-) SMR =>(+)FT=>(+)VS; which has self-regulating dynamic character (-) because the sum of negative signs is odd number.

KPD2(-):VS=>(-)SMR=>(+)GLR=>(+)FT=>(+)VS; which also has self-regulating dynamic character .

KPD3(-)SMR=>(+)GLR=>(+)FT=>(+)FU=>(+)MTVA=>(+)dOMEGA=>(+)OMEGA=>(+)VS;which has also self-regulating dynamic character.

Within KPD a few cause and effect relations are acting (UPV) for which the following dynamic relations are valid:

“ If the anchor and chain VS lifting speed is increasing, chain length SMR is reducing what results in the negative sign of cause-effect relation”; by increasing chain and anchor VS lifting speed, the number of revolutions of shaft NV is also increased resulting in positive sign of UPV.”

By increasing the relative anchor mass GSR, chain and anchor VS lifting speed is reduced resulting in

negative sign of UPV”. “By increasing of relative chain length SMR loading force is increasing as well as the total chain weight GL and also buoyancy force FU resulting in positive sign UPV.”

“By increasing the loading force -FT as well as speed of rotation of asynchronous motor, the anchor-VS and chain lifting speed is increased resulting in positive sign of the observed UPV.” “By increasing gravity-G the loading force-FT and buoyancy force-FU are increased and accordingly observed UPV has positive sign.” “By increasing loading force-FT total force is increased and consequently by increasing total force winch torque-MTVA is increased and thus the observed UPV has a positive sign.”

“By increasing buoyancy force-FU total force is decreasing resulting in a negative sign UPV.” “By increasing chain and anchor-A area as well as seawater density buoyancy force is increased resulting in positive sign of the observed UPV.”

2.2. System dynamic model of asynchronous motor

The following cause and effect relation is applicable to the first equation:

$$\frac{d\psi_{ds}}{dt} = u_{ds} - \frac{1}{T'_s} \psi_{ds} + \frac{k_r}{T'_s} \psi_{dr} + \omega_k \psi_{qs} \quad (7)$$

Variation speed of system - d condition is decreasing what results in negative sign of the observed cause and effect relation.” “By increasing rotor linkage factor -Kr, system condition variation is also increased resulting in positive sign of the observed UPV.” “By the increase of stator - Ts transient time constant, system condition variation is reduced resulting in negative sign of the observed UPV.” “If product is increasing and if stator voltage variation in axis d - Uds is increasing then system condition variation speed is also increasing resulting in a positive sign of the observed UPV.”

On the basis of the specified model given in the form of cause and effect relation of system elements, the mental verbal model of equation of asynchronous motor condition may be determined and thus a structural model and continuity diagram of the mentioned equation may be elaborated.

In this short paper, it is impossible to give a complete model (27 equations) of the asynchronous motor, complete model has been

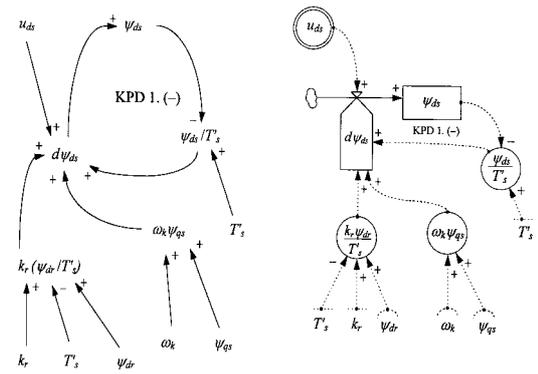


Figure 3: Structural diagram and continuity diagram of the first differential equation of the asynchronous motor condition

presented in IASTED 1998., Pittsburg, USA.

2.3. Computer simulating model BSVPAM

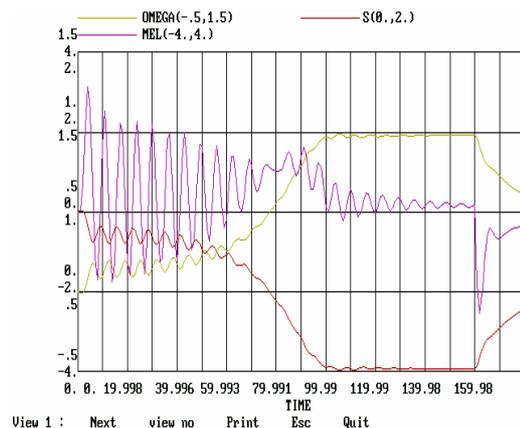
PARAMETERS OF SHIP'S ANCHOR WINDLASS:	
C G=7000	TOTAL NOMINAL WEIGHT OF ANCHOR WITH CHAIN (kg)
C GS=3000	NOMINAL ANCHOR WEIGHT (kg)
K GSR=GS/G	RELATIVE VALUE OF ANCHOR WEIGHT
K GL=4000/SM	NOMINAL CHAIN WEIGHT PER LENGTH UNIT (kg/m)
K GLR=GL/G	RELATIVE VALUE OF CHAIN WEIGHT PER LENGTH UNIT
C SM=100	CHAIN LENGTH (m)
C GR=9.81	GRAVITY (M/s*s)
K GRR=GR/SM	RELATIVE GRAVITY
C GU=1.25	SEAWATER DENSITY (kg/m*m*m)
K GUR=GU/SM	RELATIVE SEAWATER DENSITY
C VOLUM=1000	CHAIN AND ANCHOR VOLUME (m*m*m)
K POVRSR=VOLUM/SM	CHAIN AND ANCHOR AREA
R VS.KL=CLIP(STEP(OMEGA.K,0),0,FT.K,GSR)	ANCHOR LIFTING SPEED
C D=1	
A NV.K=VS.KL/(D*3.14)	SHAFT NUMBE OF ROTATION
L SMR.K=SMR.J+DT*(-VS.JK)	
N SMR=100	
A MTVA.K=FU.K*RO	WINDLASS TORQUE
C RO=0.1	CHAIN LOCKER DIAMETER
A FT.K=CLIP(STEP((GLR*SMR.K+GSR)*GR,0),0,GLR*SMR.K*GR,GSR)	LOADING FORCE
A FZ.K=CLIP(STEP(GUR*GRR*POVRSR*SMR.K,0),0,SMR.K,15)	BUOYANCY FORCE
A FU.K=FT.K-FZ.K	TOTAL FORCE
SAVE SMR,MTVA,FU,FT,FZ,VS	
PARAMETERS OF ASYNCHRONOUS MOTOR:	
C Rs=0.0141	STATOR TRANSFORMED OPERATING RESISTANCE
C Rr=0.0934	ROTOR TRANSFORMED OPERATING RESISTANCE + 5Rr
C Lcs= 0.286	STATOR TANSFORMED INDUCTANCE
C lcr= 0.1	ROTOR TRANSFORMED INDUCTANCE
C Lm=3.32	TRANSFORMED MUTUAL INDUCTANCE
C TCS=20.3	STATOR TRANSIENT TIME CONSTANT
C Tcr=3.11	ROTOR WITH 5R TRANSIENT TIME CONSTANT
C Ks=0,965	STATOR LINKAGE FACTOR

C Kr=0.95 ROTOR LINKAGE FACTOR
 C Lsig=0.12 STATOR LEAKAGE INDUCTANCE
 C Lsigr=0.175 ROTOR LEAKAGE INDUCTANCE
 C SIGMA=0.083 LEAKAGE FACTOR (SIGMA=1-Ks*Kr)
 CH=57.6 INERTIA CONSTANT
 I DIFFERENTIAL EQUATION OF CONDITION:
 R dPSId.KL=Uds.K-(PSId.K/Tcs)+
 OMEGak.K*PSIqs.K+(Kr*PSIdr.K)/Tcs
 DPSId= VARIATION SPEED OF LINKAGE FLUX
 PSId (Wb/s)
 Uds= STATOR VOLTAGE IN AXIS d (V)
 Tcs= STATOR TRANSIENT TIME CONSTANT
 OMEGak=OMP STATOR ROTATION
 SYNCHRONOUS SPEED (rad/s)
 PSIqs=STATOR LINKAGE MAGNETIC FLUX IN
 AXIS q (Wb/s)
 Kr= ROTOR LINKAGE FACTOR
 PSIdr= ROTOR LINKAGE MAGNETIC FLUX IN
 AXIS d (Wb)
 L PSId.K=PSId.J+DT*(dPSId.JK)
 N PSId=0
 PSId=STATOR LINKAGE FLUX IN AXIS d (Wb)
 DPSId=VARIATION SPEED OF STATOR
 LINKAGE FLUX IN AXIS d (Wb/s)
 A Uds.K=STEP(1,0)+
 CLIP(1,0,FT.K,GSR+1e-20)+STEP(-1,0)
 A OMEGak.K=1
 OMEGak=OMP STATOR ROTATION
 SYNCHRONOUS SPEED (rad/s)
 II DIFFERENTIAL EQUATION OF CONDITION:
 RdPSIqs.KL=Uqs.K-(PSIqs.K/Tcs)-
 OMEGak.K*PSId.K+(Kr*PSIqr.K)/Tcs
 DPSIqs=VARIATION SPEED OF STATOR
 LINKAGE MAGNETIC FLUX IN AXIS q (Wb/s)
 Uqs= STATOR VOLTAGE IN AXIS q (V)
 PSIqs=STATOR LINKAGE MAGNETIC FLUX IN
 AXIS q (Wb/s)
 Tcs= STATOR TRANSIENT TIME CONSTANT
 OMEGak=OMP STATOR ROTATION
 SYNCHRONOUS SPEED (rad/s)
 PSId= STATOR LINKAGE FLUX IN AXIS d (Wb)
 Kr= ROTOR LINKAGE FACTOR
 PSIqr=ROTOR LINKAGE MAGNETIC
 FLUX IN AXIS q (Wb)
 L PSIqs=0
 PSIqs=STATOR LINKAGE MAGNETIC FLUX IN
 AXIS q (Wb/s)
 DPSIqs= VARIABLE VARIATION SPEED PSIqs
 (Wb/s)
 A Uqs.K=0
 Usq=STATOR VOLTAGE IN AXIS q (V)
 A Uas.K=SQRT(Uds.K*Uds.K+Uqs.K*Uqs.K)
 Uas=VECTOR SUM OF VOLTAGE
 COMPONENTS IN AXES q AND d
 III DIFFERENTIAL EQUATION OF CONDITION:
 R DpsiDR.kl=Udr.K-(PSIdr.K/Tcr)+
 (OMEGak.K-OMEGA.K)*PSIqr.K+Ks*PSId.K/Tcr
 A Udr.K=0
 DPSIdr=ROTOR VARIATION SPEED OF
 LINKAGE MAGNETIC FLUX IN AXIS d (Wb/s)
 Tcr=ROTOR SA 5R TRANSIENT TIME
 CONSTANT
 Ks=STATOR LINKAGE FACTOR
 OMAGak=OMP STATOR SYNCHRONOUS
 ROTATION SPEED (rad/s)
 L PSIdr.K=PSIdr.J+DT*(dPSIdr.JK)
 PSIdr= ROTOR LINKAGE MAGNETIC FLUX IN
 AXIS d (wb)
 N PSIdr=0
 IV DIFFERENTIAL EQUATION OF CONDITION:
 R dPSIqr.KL=Uqr.K-(PSIqr.K/Tcr)-
 (OMEGak.K-OMEGA.K)*PSIdr.K+Ks*PSIas.K/Tcr
 A Uqr.K=0
 DPSIqr=ROTOR VARIATION SPEED OF
 LINKAGE MAGNETIC FLUX IN AXIS q (Wb/s)
 Tcr= ROTOR SA 5R TRANSIENT TIME CONSTANT

Ks= STATOR LINKAGE FACTOR
 OMEGak= OMP STATOR SYNCHRONOUS
 ROTATION SPEED (rad/s)
 LPSIqr.K=PSIqr.J+DT*(dPSIqr.JK)
 PSIqr= ROTOR LINKAGE MAGNETIC FLUX IN
 AXIS q (Wb)
 NPSIqr=0
 V DIFFERENTIAL EQUATION OF CONDITION:
 RdOMEGA.KL=(1/(2*h))*(Ks/Lcr)*(PSIqs.K*PSIdr.K-
 PSId.K*PSIqr.K)-(1/(2*H))*MT.K
 DOMEGA=VARIATION SPEED OF ANGLE SPEED
 (rad/s(s))
 H= INERTIA CONSTANT
 Ks= STATOR LINKAGE FACTOR
 Lcr= ROTOR TRANSFORMED INDUCTANCE
 L OMEGA.K=OMEGA.J+DT*(dOMEGA.JK)
 OMEGA= ANGLE SPEED (rad/s)
 N OMEGA=0
 VI EQUATION OF ELECTROMAGNETIC TORQUE:
 A Mel.K=PSId.K*Iqs.K-PSIqs.K*Ids.K
 VII EQUATION OF LOADING TORQUE:
 A MT.K=STEP(MTVA.K*KOPT,0)
 C KOPT=1
 VIII ADDITIONAL CURRENTS EQUATIONS:
 A IDS.K=(1/Lcs)*(PSId.K-Kr*PSIdr.K)
 AIqs.K=(1/Lcs)*(PSIqs.K-Kr*PSIqr.K)
 AIqs.K=(1/Lcs)*(PSIqs.K-Kr*PSIqr.K)
 Alas.K=SQRT(Ids.K*Ids.K+Iqs.K*Iqs.K)
 A ldr.K=(1/Lcr)*(PSIdr.K-Ks*PSId.K)
 A lqr.K=(1/Lcr)*(PSIqr.K-Ks*PSIqs.K)
 A lar.K=SQRT(ldr.K*ldr.K+lqr.K*lqr.K)
 IX ADDITIONAL SLIP AND NUMBER OF
 ASYNCHRONOUS MOTOR REVOLUTION EQUATIONS:
 S S:K=(OMEGak.K-OMEGA.K)/OMEGak.K
 N S=1
 SAVE dPSId,PSId
 SAVE dPSIqs,PSIqs
 SAVE dPSIdr,PSIdr
 SAVE dPSIqr,PSIqr
 SAVE dOMEGA,OMEGA,OMEGak
 SAVE Uds,Uqs,Uas,Ids,Iqs,Ias
 SAVE ldr,lqr,lar
 SAVE Mel,MT,S
 SPEC DT=.01,LENGTH=180,SAVPER=1

2.4. The Results of Simulation

Graphical figure of the simulation results of the BSVMP:



View 1 : Next view_no Print Esc Quit
 Figure 4: Diagram of loading torque, electric torque and slipping

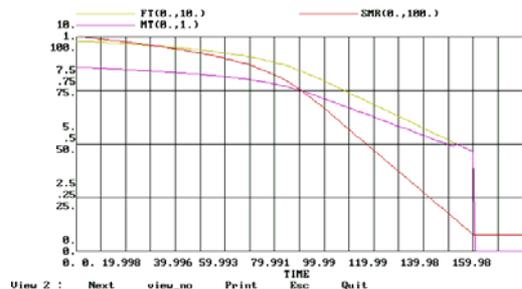


Figure 5: Diagram of loading force, buoyancy, chain length and speed

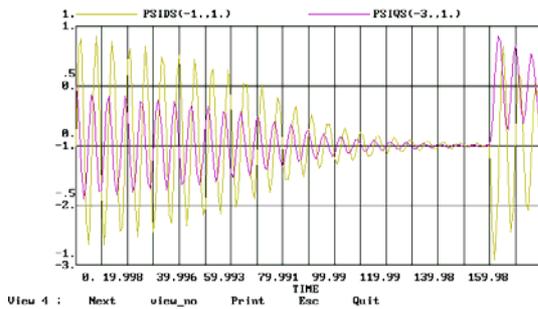


Figure 6: Stator magnetic fluxes

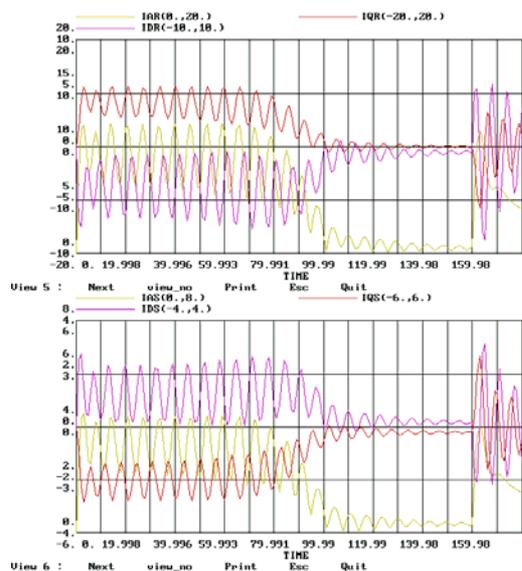


Figure 7: Stator and rotor currents

3. CONCLUSION

System Dynamics is such scientific methodology that provides the simulation of the most complex systems. In the shown example the methodology evidently indicates to the high quality of the simulations of the complex dynamical systems and it gives the opportunity to every student or engineer interested to by the same methodology modulates, optimises and simulates any scenario of the existing realities. Furthermore, the users which use this simulation methodology of the continuous models on a digital computer, create a possibility to themselves of the newest knowledge's in the behaviour of the dynamical systems. The Methodology is also significant because it doesn't

contain only a computer type of modelling, but it clearly determinates the metal, structural and mathematical modelling of the same system realities. Based on our long-term experience in the application of the dynamical methodology of simulating and in this short presentation we provide every expert in need with the possibility to acquire additional knowledge about the same system in a quick scientifically based way of exploring the complex systems.

4. REFERENCES

- Forrester, Jay W. 1973/1971. "Principles of Systems", MIT Press, Cambridge Massachusetts, USA.
- Jadric, M. and Francic, B. 1996. "Dinamikaelektričnih strojeva."(in Croatian), Manualia Universitatis Studiorum Spalatiensis, Graphics, Zagreb, Croatia.
- Munitic, A. and Milic L. and M.Milikovic, 1997. "SystemDynamics computer Simulation Model of the Marine Diesel – Drive Generation Set 1997. AutomaticControl System." IMACS World Congress on Scientific Computation, Modelling and Applied Matematchs, vol.5, Wiessenschaft & Technik Verlag, Berlin.
- Munitic,A., I.Kuzmunic, M. Krčum, 1998. "System Dynamic Simulation Modelling of the Marine Synchronous Generator", IASTED, Pittsburg. pp.372.-375
- Munitic, A. 1989. "Application Possibilities of System Dynamics Modelling." System Dynamics, Edited by Susan Spencer and George Richardson, Proceedings of the SCS Western Multiconference, SanDiego, California, A Society for Computer Simulation International, San Diego, USA.
- Munitić A., Milić L., Bupić, M.,Oct.18- 20, 2001. "System Dynamics Simulation Modeling and Heuristic Optimization of The Induction Motor", Simulation Symposium, Marseille.
- Richardson, George P. and Pugh III Aleksander L. 1981. "Introduction to System Dymanics Modelling with Dynamo.", MIT Press, Cambridge, Massachusetts, USA.



Ante Munitić received his first B.Sc. in Electric and Energetic Engineering in 1968, and his second in 1974, his M.Sc. degree Organisation System and Cybernetics Science (Operational Research) in 1978, and his Ph.D. of Organisation Science (System Dynamics), in 1983.

He is currently a full Professor of Computer and Informatic Science at the University of Split. He has published over 100 papers on system dynamics modeling and simulation, operational research, marine automatic control system and the Theory of Chaos. He has published two books: "Computer Simulation with help of System Dynamics" and "Basic Electric Energetic and Electronics Engineering".



Mario Oršulić received his B.Sc, M.Sc. and Ph.D. in mechanical engineering from Faculty of Mechanical Engineering University of Rijeka in 1968, 1984 and 1988 respectively. He is currently an associative professor at the University of Split, College of Maritime Studies. He is author or co-author of a number of bibliographical units (scientific and professional conference and journal papers, research projects, text books, etc.).

His research interest is in Marine Engineering, auxiliary marine engines and technical mechanics theory and practise.

educative system of Croatia". In June 2002. he has enrolled postgraduate study of engineering at Faculty of Mechanical Engineering and Naval Architecture. He has published 20 scientific papers on System Dynamics Simulation Modelling



Maja Krčum graduated from the Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, University of Split on March 1981. She received a graduate degree (M Sc.) at the Faculty of Electrical Engineering, University of Zagreb in 1996. In 1997 she was appointed Head of Department, also working as a tutor and counsellor. She has participated in a number of both national and international conferences where her papers and lectures were generally acknowledged as an active and valuable contribution towards the development of her profession.

Her primary interest lies in the field of shipboard propulsion systems, with a special emphasis on electrical propulsion and its numerous applications (simulation methods).

She is also a member of several national and international societies (e.g. IEEE, EMAR...).



Joško Dvornik finished elementary and high Maritime school. In school year 1996/97 he enrolled Maritime University in Split, Marine engineering Department, completed all theoretical and practical subjects included in school program on time, and passed all exams. He graduated in 2000. year on theme "*Application on computer simulation dynamics of behaviour of ship propulsion system: windlass – asynchronous engine*", with excellent degree as a first student in his class.

Since December 2001. year he has worked as younger assistant at Maritime University in Split on scientific project titled "Computer simulation model of maritime

MODELLING AND DISCRETE-EVENT SIMULATION OF COMPLEX SYSTEMS USING RAINBOW

Angelo Furfaro, Libero Nigro, Francesco Pupo

*Laboratorio di Ingegneria del Software
Dipartimento di Elettronica Informatica e Sistemistica
Università della Calabria, I-87036 Rende (CS) - Italy
Email: a.furfaro@deis.unical.it {l.nigro,f.pupo}@unical.it*

Abstract: This paper describes a modelling language –Rainbow– based on Coloured Petri Nets, which was designed for modelling and simulation of complex systems. The formalism uses Java as the net annotating language. The timing model permits different policies to be associated with places which affect the token binding process. A graphical tool was achieved in Java which supports editing, debugging and simulation of CPN models. Large models can be simulated on top of a Time Warp based distributed executor. The practical use of Rainbow is demonstrated through a scalable simulation model.

Keywords: Modelling, simulation, complex systems, coloured Petri nets, Java

1. INTRODUCTION

Coloured Petri (CP) nets (Jensen 1992-98) are a well-known class of high-level nets that extend ordinary Petri nets (Murata, 1988) by allowing tokens to carry arbitrarily complex data, and arcs to be annotated with input predicates (influencing the enabling of a transition) or output functions (stating the production rule of tokens when a transition fires). Declarations and net inscriptions can be expressed by means of mathematical notations or by using an ordinary high-level programming language.

The work described in this paper focuses on the development of a CP-net dialect -Rainbow– which was especially designed for supporting modelling and simulation of large systems, in a centralized or distributed setting. Key features of Rainbow are:

- the use of Java as the net annotating language. Colour sets of places, arc inscriptions and guards (of arcs and transitions) can directly be programmed in Java
- a timing model which accommodates both unordered and ordered places. Unordered places support classical non deterministic token selection. Ordered places can work with different token selection policies, e.g., FIFO-strict and FIFO-random, which restrict the choice of tokens during the binding process, on the basis of colours and time.

A totally portable Java-based graphical tool was achieved which enables editing, debugging and simulation of Rainbow models on a single workstation. A distributed executor based on a Time Warp mechanism (Beraldi and Nigro, 2001)(Beraldi *et al.* 2002) was implemented which supports distributed simulation over a networked system. Details of the distributed executor are described in a recent paper (Furfaro *et al.*, 2002b).

This paper summarises the Rainbow modelling language and associated general timing model. The implementation status of the project is then clarified. After that, a scalable simulation model, together with some experimental results, are presented to demonstrate the practical use of Rainbow. Finally, directions of on-going work are outlined in the conclusions.

2. THE RAINBOW MODELLING LANGUAGE

The following provides a brief and informal description of Rainbow. The formalism relies on Java as the net programming language. With respect to similar modelling languages and tools (e.g. Renew (Kummer *et al.*, 2002)), types (classes), functions and so forth are expressed in Java and not using a syntax which

requires mapping and translation in Java. Rainbow hosts only basic net constructs and focuses on time management.

2.1 Places

To each place is associated a class (*colour set*) whose instances are the admitted tokens (or colours). Place classes are extensions of the ColourSet base class. A few colour set classes, corresponding to primitive data types, are predefined so as for them to be immediately reused: ColourInt, ColourFloat, etc. Tokens in a place form a *multi-set*. A parameterless *initialization function* can be assigned to a place to provide its initial marking.

2.2 Arcs

Can be input or output. Input arcs connect places to transitions. The input places of a transition constitute the transition *preset*. Output arcs connect transitions to output places (transition *postset*). Both input and output arcs can be annotated by *arc inscriptions*, e.g. a *variable* or a *function*. More in particular, input arcs are normally decorated by a variable, which will be bound to a colour from the emanating place. In alternative, a function can be attached to an input arc, checking for the existence of suitable tokens in the relevant place of the preset. Input arc functions can be replaced by *arc guards*. A guard is a function which returns true if the token bound to the arc variable satisfies a certain selection criterion. By default, guards evaluate to true if missing. Output arc inscriptions regulate the generation of tokens at transition firing. An output arc inscription can be the same variable of an input arc, or a function which generates specific output tokens.

2.3 Transitions

A *binding element* is a pair (t,b) consisting of a transition t and a *binding* b . A binding is an assignment of values to all the variables involved with the transition, i.e., the variables used in the arc inscriptions relevant to the transition. Transition t is enabled in a marking M if there exists at least a binding for t . A *guard* can be associated with a transition for controlling the binding/enabling process. For the transition to be enabled, all the input arc and transition guards must evaluate to true. An enabled transition can fire. Firing a binding element (t,b) withdraws tokens from the preset of t according to the binding b , and generates tokens in the postset according to the t output arc inscriptions.

2.4 Timing aspects

A Rainbow model has a time notion (Jensen 1992-98) expressed by the value of a global clock (*model time*). In addition, tokens (i.e., colours) are time stamped. The time stamp of a colour reflects its generation time. Time stamped colours are components of *timed* multi-sets. The following is an example of a timed multi-set:

$$4'[a]\{1@49 \quad 2@50 \quad 1@52\} + 2'[b]\{1@49 \quad 1@50\}$$

The multi-set has four tokens of colour *a*, one with time stamp 49, two with time stamp 50 and the last one with time stamp 52, and two tokens of colour *b* respectively with time stamps 49 and 50.

To be acceptable, a binding element must be *time enabled*. A binding element is time enabled if it is composed of *ready tokens*. A token is ready if the global clock is greater than or equal to the token time stamp. Normally, the choice among ready tokens in a place is non deterministic (*unordered place*). Would there be multiple ready tokens for a given binding element, any one such a tokens can be selected to participate in the binding element. A time enabled binding element is characterized by its *enabling time*, i.e., the maximum value of the time stamps of the tokens involved in the binding element.

The global clock is automatically advanced when no binding element is time enabled at current time. In these cases, a binding element with minimum enabling time is chosen and the global clock adjusted to this value to ensure progress in model behaviour.

As in Generalized Stochastic Petri Nets (Marsan et al. 1984), Rainbow permits both *timed* and *untimed* (or immediate) *transitions* to be used in a model. Binding elements involving immediate transitions are always selected before binding elements of timed transitions. Immediate transitions can be assigned priority and probability values useful for conflict resolution (Marsan et al. 1987)(Ferscha 1994). The set of binding elements of immediate transitions having the highest priority is determined in the first place. Then, the actual binding element is selected in the set by a random choice according to transition probabilities.

Timed transitions are associated with a *delay* which affects the generation of tokens at transition firing. Firing a transition *t* at time τ is an instantaneous event whose effect is the creation of tokens in the postset of *t*, all time stamped with the value $\tau + \text{delay}$. The delay of a timed transition can be deterministic or stochastic. A *delay function* can be attached to a timed transition in order to constrain the delay value on the basis of the selected binding.

2.5 Token selection policies

The set *P* of places of a Rainbow model is partitioned in two subsets: $P = rP \cup oP$, where *rP* denotes a set of classical unordered (or random) places, *oP* is a set of *ordered* (or *queue*) places (Bause 1993)(Poses). In an ordered place colours are ranked by ascending time stamps.

One of different *token selection policies* can be associated to an ordered place: FIFO-strict, FIFO-random, LIFO-strict, LIFO-random. According to FIFO-strict, a binding element with a queue place can only occur with tokens at the queue's head (oldest tokens). FIFO-random flexibly allows a binding to occur with the first matching colour starting from the head of the ordered token list. In a similar way are defined the LIFO-strict and LIFO-random policies which visit the token ordered list starting from the youngest tokens.

FIFO policies are the most natural in many simulation models. Figure 1 shows a typical scenario. Place W contains tokens

representing units of work, which are assigned for processing to a given machine. Each machine can process one unit of work at time. For simplicity, the colour of a unit of work coincides with the corresponding machine number. Place M holds the machines available at current time. Transition $t_{process}$ models the task of a machine which processes a unit of work. Near to each place is indicated its current marking. The global clock is assumed to be 10. Finding a binding for $t_{process}$ means binding a machine colour to variable *m* so that the function *f*(*m*) returns a colour which is contained in *W*. For the purposes of the example *f*(*m*) can be the identity function: it just returns a colour from *W* equal to its argument. Function *f*(*m*) could be replaced by annotating the arc $W \rightarrow t_{process}$ with a variable, e.g. *n*, and introducing a transition guard which checks that *n* and *m* are corresponding colours. Table 1 depicts the available binding elements at current time under FIFO and/or Random policies.

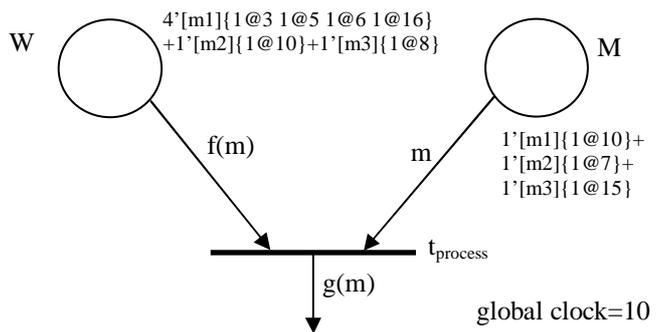


Figure 1. A typical scenario for token selection policies

W-Rand, M-Rand	b1: $\{m1@3\}_W \{m1@10\}_M$, enabling time: 10 b2: $\{m1@5\}_W \{m1@10\}_M$, enabling time: 10 b3: $\{m1@6\}_W \{m1@10\}_M$, enabling time: 10 b4: $\{m2@10\}_W \{m2@7\}_M$, enabling time: 10
W-FIFO-Strict, M-Rand (or FIFO-Rand)	b1
W-FIFO-Rand, M-Rand (or FIFO-Rand)	b1, b4

Table 1. Binding effects when applying different token selection policies

If the Random policy is adopted for both W and M places, four binding elements as possible for $t_{process}$. In particular, two bindings can be fired, one at a time and in any order: one chosen among b1, b2 or b3, and the other being b4. Generated tokens are controlled by the arc function *g*(*m*). All such tokens are time stamped by $10 + \delta(t_{process})$, where $\delta(t_{process})$ is the (estimated) delay of $t_{process}$. The two firings occur at the same time horizon (global clock=10) to express the parallelism (infinite server semantics (Ferscha, 1994) of $t_{process}$) with which physical machines (e.g., m1 and m2) process distinct units of work.

The random policy does not force tokens in W or M to be processed according to their arrival time. Constraining work units to be processed in the arrival order is the responsibility of FIFO policies. However, FIFO-Strict for W would forbid, at current time, to fire other bindings except but b1. In addition, if the m1 colour in M is ready at a time greater than the global clock, no binding would then be available at current time for $t_{process}$, although machine m2 is ready from time 7 and b4 is potentially ready for firing. FIFO-Random for W and Random or FIFO-Random for M, would constrain machines m1 and m2 to process the available units of work having minimum time stamp (see b1 and b4 bindings).

The design of the timing model of Rainbow purposely separates time management from functional aspects of a net model captured by arc inscriptions. From this point of view, an input arc

inscription can only express requirements for colour selection. The use of token time stamps and the system time advancement rule are under implicit control of the underlying executor which has responsibility in applying place selection policies.

3. IMPLEMENTATION STATUS

An implementation of Rainbow was achieved in Java through a graphical tool. The following are some points of the developed tool:

- it allows editing, debugging, simulation and analysis of CPN models. Both step-by-step execution and checkpoints (e.g., desired markings in selected places) are supported
- it hosts both coloured and non coloured nets. Non coloured nets rest on tokens which consist of the time stamp only
- it allows graphically to distinguish between unordered (default) and ordered places (split circles). A property of an ordered place concerns its selection policy
- it hosts an executor which is devoted to sequential simulation of a model. The executor uses Java reflection for accessing and invoking user-defined model functions.

Distributed simulation of a Rainbow model can be required by the computationally very expensive (in time and space) task involved with binding element processing. The critical factor is *binding calculation*. Building the bindings corresponding to a transition t requires in general exhaustively enumerating all the possible assignments of values (according to colours and time stamps available in the preset of t) to variables involved with t . A variable can be used alone on an arc or as a function parameter of an input or output arc of t . The same value of the variable must consistently be used in all its occurrences in a binding. Binding calculation is responsible for identifying all the candidate bindings existing at current time for any transition. Among alternative bindings, a random choice eventually selects the binding to fire. Ordered places and associated token selection policies obviously can speedup the relevant binding calculation process, since they restrict the possible proposed bindings.

Distributed simulation is currently dealt with externally to the Rainbow graphical tool and depends on a specialized version of the executor built on top of an agent-based Time Warp mechanism (Furfaro *et al.*, 2002b). Key points of the distributed executor are the following:

- it allows a large model to be partitioned into a collection of subnets/LPs allocated for execution one per physical processor of a networked system. The Rainbow tool makes it possible to visually decompose a net model into cuts and to save them on disk as part of the model data representation. Actually, model data representation can be archived according either to standard Java serialization or XML and associated DTD. The model data representation is parsed by a *director* agent which configures and controls the distributed simulator
- it benefits from the features of Temporal Uncertainty Time Warp –TUTW– (Beraldi and Nigro, 2001)(Beraldi *et al.*, 2002) which permits temporal uncertainty to be exploited in general distributed simulations. TUTW adopts an event delivery strategy where the occurrence time of an event is specified by a time interval and not a punctual timestamp. All of this augments the model event parallelism (events having overlapping time intervals are concurrent) and has the potential of improving the simulation performance since the control engine is given some flexibility in the event resolution, i.e., choosing the actual time stamp of events at dispatch time. Temporal uncertainty allows to relax in part

the synchronization constraints. TUTW, though, is able to keep causality among concurrent events using Lamport “happens-before” relationship. For many simulation applications, experiments have shown that TUTW is capable of improving performance of the distributed simulator with respect to the case temporal uncertainty isn’t used, without necessarily compromising the accuracy of the results.

4. A SIMULATION MODEL

The following describes a complex and scalable simulation model with the goal of illustrating the practical use of Rainbow and its graphical tool. The model is based on a non coloured TPN model proposed by Zuberek (Zuberek, 1999)(Zuberek, 2002) for studying the influence of long-latency memory accesses in distributed-memory multithreaded multiprocessors (DM-MM). The simulation model was actually experimented for exploring the effects on the cpu utilization of component heterogeneity vs locality of memory references. All of this can be accomplished without changes in the model topology.

4.1 A multiprocessor multithreaded model

A DM-MM system with $n \times n$ processors (or nodes) interconnected by a bi-dimensional torus-like switching network is assumed (see Figure 2). Each processor can communicate directly with its four neighbours. An outline of the node architecture is portrayed in Figure 3).

Each node has a local memory and two network interfaces allowing concurrent send/receive operations. Any processor can issue a memory request which can be directed to local memory or to the memory module of some remote node, which can be reached through the interconnecting network according to a suitable path, e.g., one with shortest distance. Through the outbound interface is routed all the outgoing traffic concerning remote memory requests originated in this node, or the results of memory operations asked by remote processors to the memory module in this node. Through the inbound interface occurs all the incoming traffic consisting of remote originated requests to the memory of this node, as well as the results of remote request operations which come back to the originating nodes.

Each processor has a queue of ready threads. Whenever a long-latency memory operation is started at this processor, a *context switch* is accomplished as follows: first the current thread is suspended, then the memory operation is forwarded to the relevant memory module (local or remote); finally, processor execution is resumed by selecting another thread, if there are any, from the ready queue, and transferring the control to its next instruction. When the result of this memory request is received, the corresponding thread changes its status from “suspended” to “ready” and it is added to the ready queue waiting for dispatch.

4.1.1 Model parameters

The runlength of a thread, ℓ_i , represents the number of instructions executed, on the average, between context switches. This parameter is directly related to the probability that an instruction raises a long-latency memory access. Two other important parameters are p_ℓ and $p_r=1-p_\ell$ that is respectively the probability that a memory access is to local memory or is directed to a remote memory node.

The values of p_ℓ and p_r control the amount of switching network traffic and congestion vs local node memory accesses. Finally, the (average) number of available threads, n_i , influences the utilization of system components and the overall system performance. In the Rainbow model of Figure 4, the value of this parameter is assumed to not change with time.

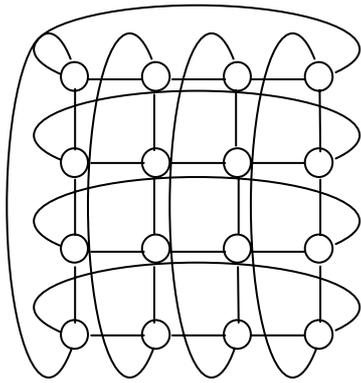


Figure 2. Switching network

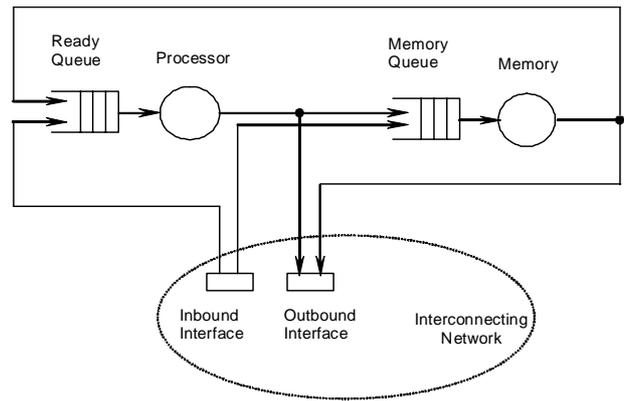


Figure 3. Node architecture

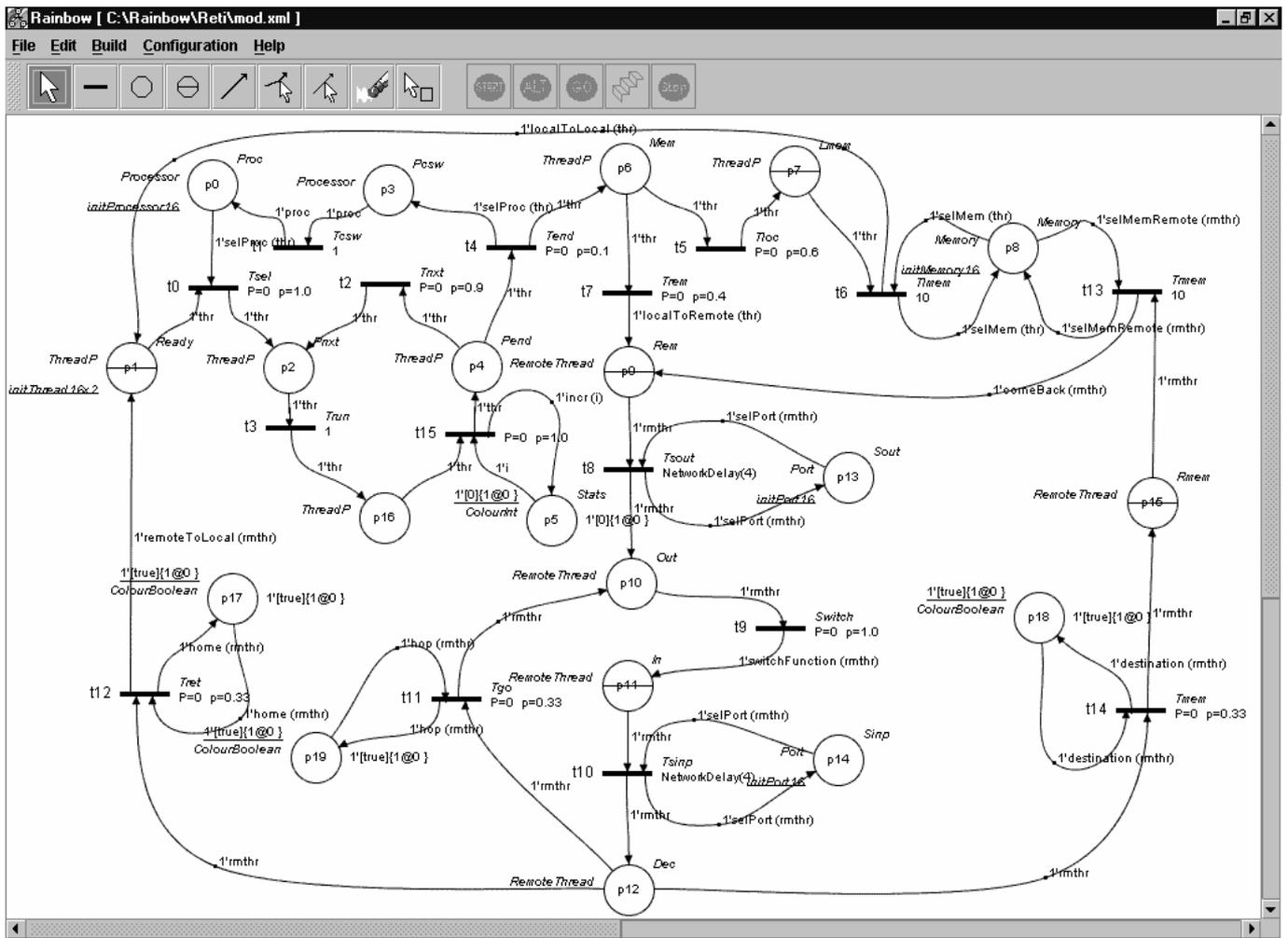


Figure 4. A DM-MM Rainbow model

4.1.2 A DM-MM Rainbow model

The model shown in Figure 4 is logically organized into four sections: a processing subnet (places p0 to p4 and p16, transitions t0 to t4), a memory subnet (places p6 to p9 and p15, transitions t5, t6 and t13), a switch subnet (places p9 to p14, p7 to p19, transitions t7 to t12, t14), a statistics subnet (place p15 and transition t15). Component replications in the physical system (Figure 2) are achieved by colour replications in the fixed model topology. For instance, all the available processors are initially represented by colours in the ready queues of the various processors are mapped on to colours in the FIFO-Random place p4 (or Ready), all the switch board colours are kept in p13 and p14

places and all the memory modules are represented by colours in the p8 (Memory) place. Scalability of the model is automatically ensured by adjusting the initial number of colours in places p0, p4, p8, p13 and p14.

When a processor is available in p0 for processing a thread from p1, a thread token is generated in p2. Transition t3 (Trun) simulates the execution of one instruction. Its delay is 1. All other delays in the model are expressed in terms of number of instructions. Thread execution is simulated by the loop p2-p16-p4 and transitions Trun, t15 and Tnxt. Transition t15 serves only for statistical purposes. At each firing of t15 (or equivalently of Trun) the counter in p5 gets incremented. Place p4 is a free-choice. Immediate transitions Tnxt and Tend represent respectively the

execution of a non memory accessing instruction or a memory request which implies a context switch. In the latter case, a processor token is deposited in *Pcsw* (or *p3*) with the timed transition *Tcsw* modelling the actual thread context switch. At the end of the context switch the processor becomes again available for processing the next thread from its ready queue. Probability of *Tnxt* (and then of *Tend*) mirrors the (average) runlength of a thread.

Place *Mem* (*p6*) is a free-choice. Immediate transitions *Tloc* and *Trem* represent respectively an access to local memory or an access to the memory module of a remote processor. The probability of *Tloc* (and then of *Trem*) captures the locality of memory accesses vs remote memory accesses. Local requests are held in the FIFO-Random place *Lmem* waiting for the memory (place *Memory* or *p8*) to be available. An actual memory operation is modelled by *Tlmem* timed transition. Remote thread requests are maintained in the *Rmem* place and served by *Trmem* timed transition. Firing frequencies of *Tlmem* and *Trmem* are function of token multiplicity respectively in places *Lmem* and *Rmem*. After being served, a remote thread request is routed into *Rem* for it to engage the coming back to home path through the interconnection network.

The outbound/inbound interfaces of processor nodes are respectively modelled by *Tsout* and *Sout* and *Tsinp* and *Sinp*. The switching network is represented by places *Out* and *In* and transition *Switch*. A remote memory request which has been transmitted through the switch is received in the *Dec* place from which it can proceed (next *hop*) in the network (transition *Tgo*), or it just arrived at the destination node (*Tmem* transition) or at its home node (transition *Tret*). In the physical system such decision depends on the current position within the transmission path toward the target node which is the remote node during forward movement or the home node during backward movement.

The design of the DM-MM model was driven by the desire to reproduce “as close as possible” the behaviour of the actual system. This in turn motivated the adoption of FIFO-Random policy for ordered places *Ready*, *Rem*, *In*, *Lmem* and *Rmem*, and the introduction of suitable colour sets and arc inscription functions. For brevity, the following only provides an informal description of colour sets and arc functions. The colour set is indicated at the left of a place in Figure 3.

A processor is encoded by its *id* (an int colour). A thread keeps the processor number to which it is assigned (*ThreadP* colour set) and the time at which it enters the ready queue. A *Memory* colour is identified by its processor number too. A long-latency memory request is modelled by a *RemoteThread* colour. A remote thread moves along the network and carries such information as: the originating processor, the destination processor, *shortest_path* to destination, and current processor position in the switching network. The path to destination is concretely expressed by the number of hops to be taken respectively at North, East, South and Ovest from current position. The *localToRemote* function of arc *Trem-Rem* transforms a local thread into a remote thread by choosing a destination node and a path to follow for reaching it. The *remoteToLocal* function on the arc *Tret-Ready* converts a remote thread to its local representation. The *selPort* function selects the in/out switch port of the current processor a remote thread is passing through. It is ensured that *Tsout* and *Tsinp* have a single server semantics.

The choice among *Tmem*, *Tgo* and *Tret* transitions from *Dec*, is made by checking current position of the remote thread. In the case of an intermediate position, the *hop* function on the arc *p19-Tgo* enables *Tgo* and the thread proceeds for a next hop in the interconnection network. Similarly, *destination* function on *p18-Tmem* and *home* function on *p17-Tret* respectively enable *Tmem* or *Tret* in the case the current position of the remote thread coincides

with the target node (destination or home node). Only one transition among *Tret*, *Tgo* and *Tmem* can be enabled at a same time. Places *p17*, *p18* and *p19* hold a boolean colour which always is true. Only in the case the corresponding checking function (*home*, *hop* or *destination*) returns the true colour is the transition enabled. After a firing of *Tret*, *Tgo* or *Tmem*, the true colour is reconstructed in its corresponding place.

The *switch* function on the arc *Switch-In* is responsible of updating the current position of a remote thread after a hop. The *localToLocal* function on the arc *Tlmem-Ready*, saves in the thread token the time when it was generated (its time stamp) by *Tlmem*. Such information allow estimating thread waiting time in the ready queue.

Functions *selMem* on arc *Memory-Tlmem* and *selMemRemote* on arc *Trmem-Memory* select the memory colour of the processor respectively specified by local or remote thread. It is ensured that a memory module is always handled according to the single server semantics.

The *NetworkDelay* function attached to *Tsout* and *Tsinp* transitions is an example of a function which can set the delay of a timed transition according to model parameters. The function receives as an argument the number of switching boards which are assumed to be of low speed. *NetworkDelay* provides the context for a customization of the built-in *getDelay* function which in the case of *Tsout* and *Tsinp* was redefined in order to receive the remote thread selected from *Rem*. *getDelay* can thus regulate the actual delay of *Tsout* or *Tsinp* according to the managed remote thread, and its current position.

4.1.3 Simulation experiments

The DM-MM model was experimented in a case for evaluating the influence of component heterogeneity on the cpu utilization vs the probability of local/remote memory accesses. Heterogeneity can occur in the computing power, switching board and memory module of processors. In the following, for demonstration purposes, some executions carried out by considering only switch heterogeneity are reported. Different runs refer to varying the amount of node switches which introduce additional communication delays. A low speed switching board is assumed to double the transmission delay.

Figures 5 to 7 depict the estimated model cpu utilization for a DM-MM with 16 processors, and a probability p_ℓ that a memory access is local (see transition *Tloc* in Figure 3) respectively of 0.4, 0.6 and 0.8. In the experiments, the simulation time was 10^4 , the thread number n_t was varied from 2 to 10, and the number of low speed switching boards was varied from 0 (homogeneous case of high speed boards) to 16 (homogeneous case of low speed boards). Moreover, the runlength of threads was set to 10 (probability of *Tnxt* 0.9 and of *Tend* 0.1).

Figures 5 to 7 confirm that the critical factor on the cpu utilization is the p_ℓ value. When p_ℓ is 0.8 a good cpu utilization is achieved even when processors are loaded with a small number of threads.

5. CONCLUSIONS

Rainbow is a formalism based on Coloured Petri Nets (Jensen, 1992-98). It was designed for supporting modelling and simulation of complex systems. Prototyping tools were achieved which allow to experiment with simulation models both in a centralised and a distributed framework on top of a Time Warp mechanism (Beraldi and Nigro, 2001)(Beraldi *et al.*, 2002). A key factor of the Rainbow project is the adoption of Java both as the net annotating language and as the tools implementation language. All of this simplifies the use of the modelling language and makes the achieved tools totally portable almost on every platform.

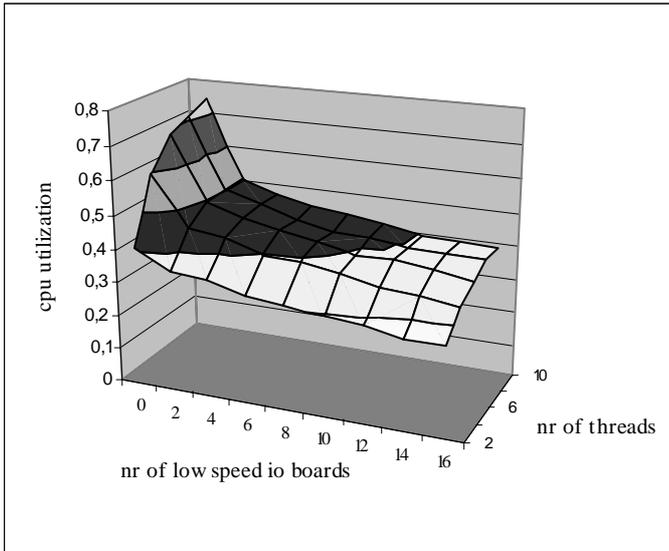


Figure 5. Cpu utilization, 16 processors, $p_t=0.4$

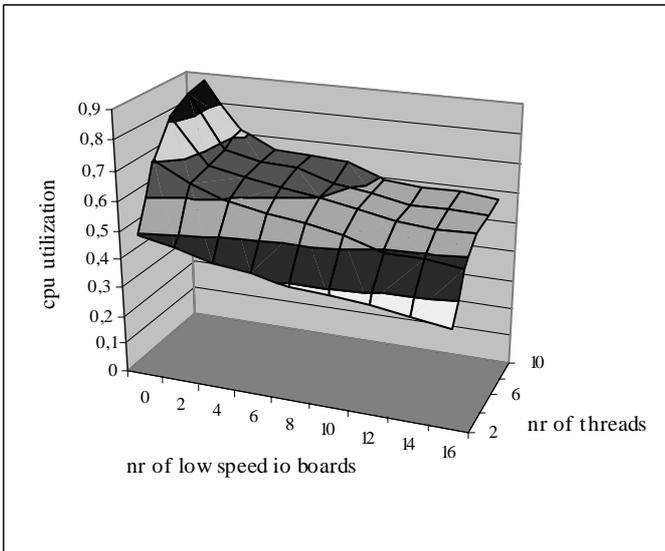


Figure 6. Cpu utilization, 16 processors, $p_t=0.6$

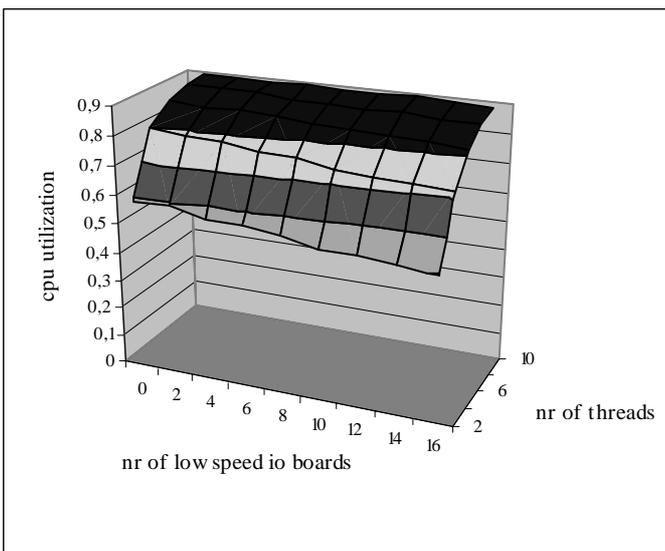


Figure 7. Cpu utilization, 16 processors, $p_t=0.8$

On going and future work is geared at

- optimising the Rainbow executor by improving the binding calculation process which critically affects the simulation

performance. From this point of view the aim is to replace the actual linked-list representation of colour multi-sets in places by more efficient data-structures and algorithms (Mortensen, 2001)

- extending the graphical tool with aspect-oriented features (CACM, 2001)(Furfaro *et al.*, 2002a), i.e., the possibility of adding to a model a crosscutting specification (*monitor*) useful for monitoring and analysing the simulation (Wells, 2002). A monitor would catch selected event occurrences in the model and make necessary book-keepings for statistics computation. Aspect-oriented monitors would be transparently attached to a model by avoiding explicit subnets for statistical computations to be introduced

6. REFERENCES

- (Bause, 1993) F. Bause. Queuing Petri Nets: A formalism for the combined qualitative and quantitative analysis of systems. *Proc. of the Int. Workshop on Petri Nets and Performance Models*, pp. 14-23, Toulouse, October, 1993.
- (Beraldi and Nigro, 2001) R. Beraldi, L. Nigro. A time warp mechanism based on temporal uncertainty. *Transactions of the Society for Modelling and Simulation International*, **18**(2), June, pp. 60-72, 2001.
- (Beraldi *et al.*, 2002) R. Beraldi, L. Nigro, A. Orlando, F. Pupo. Temporal Uncertainty Time Warp: An agent-based implementation. *Proc. of 35th Annual Simulation Symposium*, 14-18 April, San Diego, CA, pp. 72-79, 2002.
- (CACM, 2001) *Communications of the ACM*. Aspect-oriented programming. **44**(10), pp. 29-99, October, 2001.
- (Ferscha 1994) A. Ferscha. Concurrent execution of timed Petri nets. *Proc. of 1994 Winter Simulation Conference (WSC94)*, Lake Buena Vista, Florida, USA, pp. 229-236, 1994.
- (Furfaro *et al.*, 2002a) A. Furfaro, L. Nigro, F. Pupo. Aspect oriented programming using actors. *Proc. of 22nd IEEE Int. Conference on Distributed Computing Systems Workshops, Aspect Oriented Programming for Distributed Computing Systems (AOPDCS 2002)*, Vienna, Austria, 2-5 July, pp. 493-498, 2002.
- (Furfaro *et al.*, 2002b) A. Furfaro, L. Nigro, F. Pupo. Distributed simulation of Timed Coloured Petri Nets. *Proc. of Sixth IEEE Int. Workshop on Distributed Simulation and Real-Time Applications (DS-RT 2002)*, 11-13 October, Fort Worth (Texas), IEEE Comp. Society, pp. 159-166, 2002.
- (Jensen *et al.*, 1996) K. Jensen, S. Christensen, P. Huber and M. Holla. (1996). Design/CPN. A reference manual. Computer Science Department, University of Aarhus. Online: <http://www.daimi.aau.dk/designCPN/>, 1996.
- (Jensen, 1992-98) K. Jensen. *Coloured Petri Nets - Basic concepts, analysis methods and practical use*. Vol. 1, 2, 3. EATCS Monographs on Theoretical Computer Science. Springer-Verlag, 1992-98.
- (Jensen *et al.*, 1996) K. Jensen, S. Christensen, P. Huber and M. Holla. Design/CPN. A reference manual. Computer Science Department, University of Aarhus. Online: <http://www.daimi.aau.dk/designCPN/>.
- (Kummer *et al.*, 2002) O. Kummer, F. Wienberg, M. Duvigneau (2002). Renew-User Guide. <http://www.informatik.uni-hamburg.de/TGI/renew/renew.html>.
- (Marsan *et al.*, 1984) M.A. Marsan, G. Balbo, G. Conte. A class of generalized stochastic Petri nets for the performance evaluation of systems. *ACM Transactions on Computer Systems*, **2**(2), pp. 93-122, 1984.
- (Marsan *et al.*, 1987) M.A. Marsan, G. Balbo, G. Chiola and G. Conte. Generalised Stochastic Petri Nets revisited: random switches and priorities. In *Proc. of the 2nd Int. Workshop on Petri Nets and Performance Models*, pp. 44-53, IEEE-CS Press, 1987.
- (Mortensen, 2001) K. H. Mortensen. Efficient data structures and algorithms for a coloured Petri nets simulator. In: Kurt Jensen (Ed.): *3rd Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools (CPN'01)*, pp. 57-74. DAIMI PB-554, University of Aarhus, August 2001.
- (Murata, 1989) T. Murata. Petri nets: properties, analysis and applications. *Proceedings of the IEEE*, **77**(4), pp. 541-580, 1989.
- (Poses) Poses on-line: <http://www.gpc.de>
- (Wells, 2002) L. Wells. Performance analysis using Coloured Petri Nets. *Proc. of MASCOTS 2002*, 11-16 October, Fort Worth (Texas), pp. 217-221, 2002.
- (Zuberek, 1999) W.M. Zuberek. Performance modeling of multithreaded distributed memory architectures. *Proc. of 2nd Workshop on Hardware Design and Petri Nets*, Williamsburg, VA, pp. 63-82, 1999.
- (Zuberek, 2002) W.M. Zuberek. Approximate simulation of distributed-memory multithreaded multiprocessors. *Proc. of 35th Annual Simulation Symposium*, 14-18 April, San Diego, CA, pp. 107-114, 2002.

SIMULATION OF SELF ORGANIZING STRUCTURES USING NEURO MECHANICAL NETWORKS

MAGNUS SETHSON

*Dept. of Mechanical Engineering
Linköping University
Sweden*

PETTER KRUS

*Dept. of Mechanical Engineering
Linköping University
Sweden*

MATTS KARLSSON

*Dept. of Biomedical Engineering
Linköping University
Sweden*

Abstract: The neuro mechanical network consists of a large number of one-dimensional elements connected into a topological graph of intelligent actuators in 2 or 3 dimensions. This forms a self actuating mechanical network that can be trained to perform certain tasks. In the analysis and training of such networks the time domain simulation of the network performance becomes important. Even though the basic components hardly exist in hardware at present, the study of such networks gives us interesting models to design and analysis the mechanisms of the near future using current technologies and engineering tools. The neuro mechanical network has a meaning also at a micro or even macro level in order to realize highly robust flexible actuator systems. Another potential use is for design of more conventional system, requiring a minimum of components. Furthermore it can be used as an explanatory model for some of the mechanics found in very complex biological systems, e.g. heart muscles. The key to success in design such networks will be the training of the neurons handling the information propagation through the structure. To be able to evaluate its dynamic behaviour, time domain simulation techniques are used. Some preliminary results of such simulations and their general implementation are presented in this paper.

Keywords: Simulation, neuro mechanical networks, large scale systems

Address: Magnus Sethson, Campus Valla Bld. A, IKP/MekSys, SE-58183 Linköping, SWEDEN, magse@ikp.liu.se

1. INTRODUCTION

By studying the body tissues of humans and animals one get a clear view of its cellular structure. Also muscle fibres reveals a ordered pattern of fibres, beneficial for its primary function or load direction. The structures in nature are built up by small basic elements that form a larger functional component. This might become the concept of future manufacturing technologies as well. Layered manufacturing and 3D-printing machinery is examples of that.

Several different types of basic elements might be used to form a working structure and by the mixture of different kinds of elements one get multi functional components that reflect the basic characteristics of the foundation elements. This is indeed an interesting approach to future mechanical design. However, it is still difficult to analysis the properties of a system or network of such elements, even if the overall structure has a general behaviour that is easy to model. The internal interaction among all elements may be hard to solve in detail. Computer tools for such analysis are needed. In fact the computer selection, simulation and optimization tools are essential to the development of selforganizing structures. In the same way that nature has evolved the designs of all species, future computer tools will be able to select and evolve the interior of self actuating structures of

advanced complex robots and machinery. This scenario is sometimes analogues to VLSI design within the field of electronics, where most of the present highly integrated designs would not exist without computational tools for both analysis and synthesis.

1.1. Project Presentation. This project, called Neuro Mechanical Networks (NMN), tries to evaluate the properties of such networks for use in engineering systems design, [3][8]. In figure 1 is a simple layout of a neuro mechanical network presented. In general, the used approach introduce an actuating element containing several energy conversion processes and sensing capabilities, see section 2.3. By establishing software for simulation, selection and training of the neuro mechanical network one get an engineering tool that support the future study of the application of the network, its topology, and the characteristics of the building elements.

It is worth to emphasize that this work relates to the computational synthesis of systems of large numbers of small elements. Similar structures have been studied in statistical mechanical since the late eighties [4][6]. However, in this work entropy studies of the general behaviour of the network of elements is not studied, even though that is most likely interesting for the understanding of neuro mechanical networks.

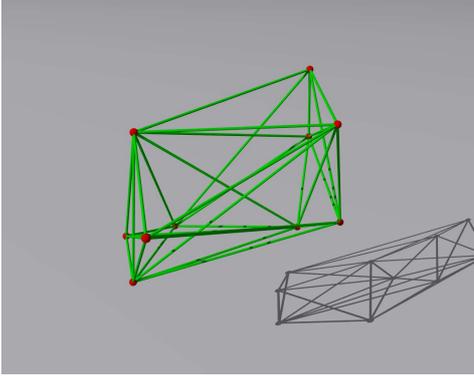


Figure 1: This is a typical layout of a neuro mechanical network. The spheres represent the nodes and the rods represent actuators. The topology is never changed during simulation.

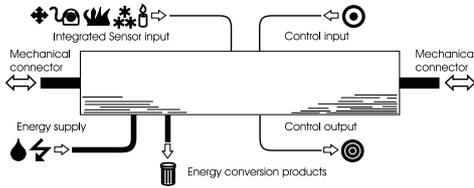


Figure 2: An idiomatic figure of a general actuator. It contain signal processing, sensing capabilities and energy conversation processes. Even though this is an element of the future it is most interesting to study for the understanding of more conventional designs.

The computational tools for this development process may use formal algebraic methods or optimizing functionality to fulfil the goal of the design. This is similar to electronics design tools making use of both analytical tools for circuit layout and different kinds of optimizing strategies for handling non-linearity phenomena and dynamics. The creation of mechanical actuation systems may in the future be dependent on small engineering elements in large numbers and computational synthesis tools. Such tools need to be able to configure the neuro mechanical network both in terms of topology and dynamical behaviour.

2. THE GEOMETRY OF NEURO MECHANICAL NETWORKS

The neuro mechanical network consists of a simple actuating element that performs either positional actuation, like a position servo, or a force actuation similar to a spring. In [3] a position servo approach has been adopted. This paper describes the force actuating approach.

Each actuator can be looked upon as a spring with and actuating element. Also a damping element is included. By tuning the basic properties of each actuators such as stiffness, actuation and damping one get a tool for creating more actuated bone-like structures with internal damping domains that dissipative energy from the system. The general layout of such actuator is shown in figure 2. The tuning process becomes very complex and computer power demanding. A tuning process that is similar to the selections of the fittest in nature is currently under study. There are several possible ways of tuning the individual actuator behaviours. In this work we use neural networks in every node controlling the actuation of every attached actuator. The actuator itself also has a neural network to balance the signals from each of its ends. The inputs to the neural network in the nodes are the actual actuation displacement of each actuator. The signal propagation velocity then becomes the same as the actuation velocity. The signal propagates with the same speed through the network as a mechanical wave of displacements in the actuators. This approach gives us one important benefit: It improves the numerical stability in the time domain simulation of the system. Another interesting approach uses the forces in the actuators as inputs to the nodes. This provides us with the ability to tune the actual rim or border of the neuro mechanical network. An actuator that has been tuned to no longer provide any force, having a low stiffness coefficient, does not transfer signals anymore. Therefore, the signal routes become consistent with the active mechanical structure and cavities in the structure may emerge in a natural way. The flow of signals is schematically shown in figure 3. By combining both displacement and force of the actuators into the neural networks of the nodes further generalization may be achieved. It is important that the time-domain simulation technique used supports all the proposed synapses of the neural network. That includes actuation length, sensors, position of elements and general signal fields. The trapezoidal integration rule for numerical integration seems to provide a good foundation for such requirement. In its most general form it may be formulated as in equation 1.

$$(1) \quad F[(n+1)T] = F[nT] + \frac{T}{2} (f[(n+1)T] + f[nT])$$

Notice that the actuating and damping element most likely produces energy conversion products like heat or gases. These rest products need to be

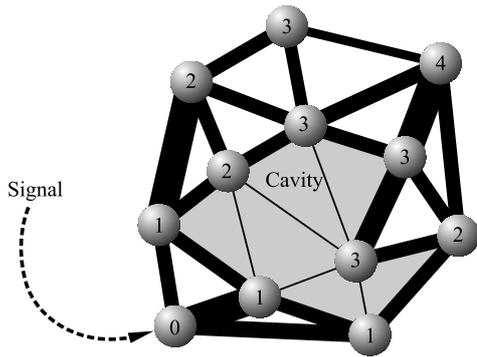


Figure 3: The general signal flow for controlling the neuro mechanical network. A signal is provided at the 0 node. The number in the node represents the time delay of the signal. Some of the actuators have just collapsed into a small line, unable to support any forces. Notice that no signals are routed through them.

transported away from the area. At the same time energy is required to be supplied into the system. These requirements for transport may either be inside the structure or parallel to it. This finding indicates a clear geometric dependency of the design problem. The actuator may not be modelled as a scalar symbolic element. Instead it needs to have length, thickness and position in space. The most general approach to an element finite in space will be the one-dimensional actuator connected at its ends towards other actuators. The connection may be free of friction and not supporting torque. The connecting joints will also host some of the controlling electronics for the neural signal propagation. By forming these simple actuators into a network we get a mechanical structure with large number degrees of freedom. The tuning of the actuators then becomes an engineering task making use of many traditional engineering disciplines in a new integrated way: the task is to make all these actuators respond to the environment and input signals in order to achieve some predefined objective. The time domain simulation technique becomes a natural choice for predicting the behaviours of a certain neuro mechanical network. One may compare this to our arms, which contain thousands of muscle fibres and still only have a few degrees of freedom.

2.1. Time domain simulations and optimizations. The application of time domain simulations for training the neurons is studied in this work. We also assume that the network perform some dynamic task like actuation or movement. Any static or quasi-static performance of the network will not be analyzed here. However, such analysis may very well be efficient and beneficial to the engineering process of tuning the neuro mechanical network.

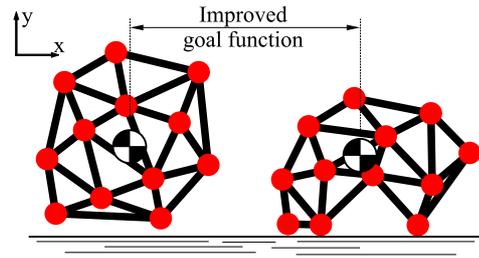


Figure 4: Example of a goal function and its direct relation to the structure. The network is trained to move along the x-axis. The simple objective then becomes to maximise the x-coordinate of the centre of gravity for the structure.

Such tuning will most likely rely upon algebraic methods and not relying on numerical algorithms.

A goal function is needed for designating a figure of merit on the actual configuration of a network. Since there are so many degrees of freedom, the goal function is formulated closely to the desired task of the network and not as an objective for each actuator. An example of that is shown in 4 where a desired movement along a certain direction is represented by the x-coordinate for the centre of gravity for the whole structure. The goal function then becomes very easy to formulate, it is simply to maximise the x-coordinate of the system. However, for large numbers of degrees of freedom this results in a huge set of possible solutions. There are several possible ways of dealing with this problem. One previously studied is the dimensional reduction of geometric models, especially triangular meshes [7]. Hierarchical geometric models created by applying simple rule sets may also be used. This is similar to cellular automata. In this work genetic algorithms have been selected in the first test designs of neuro mechanical networks. Its general characteristics seem to be beneficial to the design process at the current stage of the project. However this leads to a need for very large computational power. Other techniques may very well be used for network training. Still, in all possible approaches for training the network, a short simulation time becomes essential since each selection step may require thousands of simulations.

2.2. Neurons. The neurons of the network, both in the actuator and in the connecting nodes, are defined by the well known sigmoid function [2][1], see equations 2 and 3. s_0 in eq. 3 represent the input offset.

$$(2) \quad f(S_N) = \frac{2}{1 + e^{-\lambda(S_N - \Theta)}} - 1$$

$$(3) \quad S_N = \sum_{n=0}^N s_n w_n$$

Since the mapping of the neural network is directly related to the topology of the mechanical actuator network we often get a neural network that is recurrent, signal may very well propagate in a cyclic way through the network. This could lead to standing waves in the structure, a kind of muscular limit cycle. The limitation of signal propagation speed then becomes important for the stability of the network.

2.3. Actuator. Here, the actuator is described by a rod having a certain stiffness K , parallel to an activation element F_A that can introduce a force in either direction. The relative motion between the ends of the rod is damped by a damping element B . The stiffness is separated into two springs, one attached at each nodes of the actuator. However, in this work the stiffness and damping properties are moved into the nodes for numerical reasons. This simplifies the simulation algorithm used.

The actual displacement of the actuators is used as neuron inputs to the neural networks in the nodes. There is no limitation in actuation length of the actuators. However, to support the neural network with a well defined input signal from the actuators, displacements, are normalized against a pre-defined individual nominal actuation length. Typically, values of 80% to 100% of the initial length of the actuator have been used. This means that an actuator is supposed, but not limited, to work only in the range of 50% to 150% of the initial length.

2.4. Nodes. The nodes contain the major part of the simulation task and it represents the mass of the system. All forces from the actuators are summed up to form a total force vector for each node. This vector is then integrated twice to get the new updated position of the node. The trapezoidal rule is used for integration, see equation 1. The trapezoidal rule for integration has a special interpretation in transmission line modelling [5] in that it represent the time delay for a wave travelling from one end to the other of a inertial element. In this case it can be simplified even more if all summation is done using the same T of equation 1. We assume that all signal propagations between nodes takes the same time, independent of distance. The assumption is motivated by the fact that the simulations are not primarily used for obtaining the most accurate physical behaviour, but used merely

for selection and comparison between different simulations in an optimization scheme.

As in [5] the boundary conditions for the wave propagation elements is represented by characteristics of the form shown in equation 4 and 5. Normally the c and Z parameters are provided into the simulation component and the effort e and flow f are then solved for. In this work, however, the actuators are just calculating their length. The force acting upon the node is calculated by the node itself by using its current velocity and position. This approach gives us a numerical integration scheme that is computer memory linear. All calculations are done from two sequential lists of primitives, the actuators and nodes. This will improve the computational speed of the application. The memory storage requirement is also reduced since there is no need to save the c and Z variables, which simplifies even further the proposed numerical scheme. However, one should remember that the proposed simulation scheme does not represent an accurate physical model since the dynamics of the system becomes dependent upon the number of nodes and simulation time step.

$$(4) \quad e_1(t) = Z [f_1(t) + f_2(t - T)] + p_2(t - T)$$

$$(5) \quad e_2(t) = Z [f_2(t) + f_1(t - T)] + p_1(t - T)$$

Therefore the simple integration of the velocity and positional states of the nodes becomes as in equation 6 and 7.

$$(6) \quad \dot{x}_n = \dot{x}_{n-1} \frac{T}{2} [\ddot{x}_n + \ddot{x}_{n-1}]$$

$$(7) \quad x_n = x_{n-1} \frac{T}{2} [\dot{x}_n + \dot{x}_{n-1}]$$

The acceleration \ddot{x} is defined by the mass m of each node and the summed force F_n .

2.5. Numerical stability. As said before, it is beneficial for the computational efficiency if the simulation algorithm has a linear memory layout. This can be achieved almost completely by introducing some approximations to the time domain simulation model. Let assume that the node are described by the following equation of momentum.

$$(8) \quad F_A - Kx - B\dot{x} = m\ddot{x}$$

This may be transformed into the frequency s -plane.

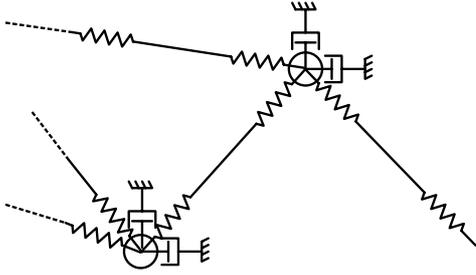


Figure 5: The general layout of actuators and nodes in the proposed simulation algorithm. Notice that the actuators do not include any dynamics.

$$(9) \quad F_A - KX - BsX = ms^2X$$

In solving equation 9 it becomes necessary to handle the numerical stability in a sensible way and at the same time keep the computational efficiency up. The proposed way of doing that tries to estimate an upper limit of the stiffness and damping factors for each node in the neuro mechanical network. The actuator stiffness may be defined in a traditional way.

$$(10) \quad \frac{\Delta F_A}{\Delta X} K = 1 + \frac{B}{K}s + \frac{m}{K}s^2 = 1 + \frac{2\delta}{\omega}s + \frac{s^2}{\omega^2}$$

Assuming optimal damping, that means real-valued roots to the polynomial in 10 we get an expression for B_{opt} .

$$(11) \quad B_{opt} = 2\sqrt{Km}$$

Even though the system might refer to a physical layout we apply a damping factor of B_{opt} to improve the numerical stability. The damping factor is applied in all dimensions of the nodes coordinates.

The actuators are attached to nodes by springs, see figure 5, that represent the stiffness of the actuator or rod connecting between two nodes. The worst case in terms of stiffness for all possible configurations of actuators and nodes is when all actuators line up in one direction. A damper is introduced in every nodes origin in each coordinate direction. Normally we get one, two or three dampers in an orthogonal arrangement. The least damped system will be found when all actuators of a node are aligned to a coordinate axis. Notice that the damping is not applied in the actuators themselves but directly at the nodes. The K in eq. 11 refers to the total stiffness of all attached actuators of a node.

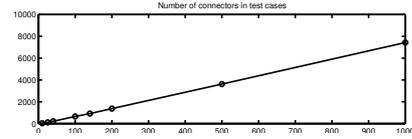


Figure 6: The number of actuators, A_N , scales almost linear with the number of nodes, N .

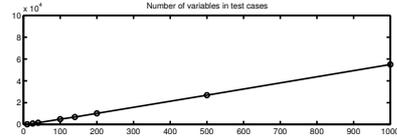


Figure 7: The number of design variables follows the number of connectors quite well, see figure 6. Still it is a function of topology, nodes and connectors.

Another important factor for stability is the time step T for the numerical integration. As the trapezoidal rule has a complete stability region the expression for the maximum T for marching the state integration becomes as follows.

$$(12) \quad T_{opt} \leq \min_N \left[\pi \sqrt{\frac{m}{K}} \right]$$

The two limiting values of B_{opt} and T_{opt} may preferably be updated before the first time step of each simulation. This requires the stiffness K of each spring in the nodes to remain unchanged during all of the simulation. This might seem a severe limitation, but it only applies to the maximum stiffness achievable in the interior of the actuators. In most cases that is a well known factor. It is also possible to have different B_{opt} and T_{opt} for all nodes. This has not been studied further in this work but may be an interesting approach for further improve the computational speed. In the same way the time step may be varying across the structure. Some parts of the structure are then only updated every second or third time step. Equations 11 and 12 then becomes important for the sectioning of the different numerical domains of the structure.

3. RESULTS

The characteristics of the simulation software is presented as diagrams in figure 6 to 9. The used genetic algorithm typically require a population size that is normally 10 to 50 times the number of design variables, N_d . This means that the computer memory storage requirement has a characteristic quadratic scaling to the number of design variables, $\propto N_d^2$. This is clearly seen in figure 8.

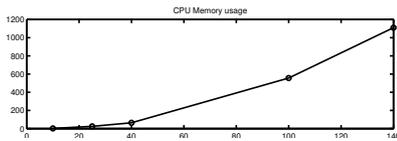


Figure 8: The amount of computer memory needed for the optimizations is growing exponential in a most unpleasent way. From the diagram one can make the conclusion that on 32-bit machine only systems with less than about 200 nodes can be trained. This underlines the need for a better implementation of the framework.

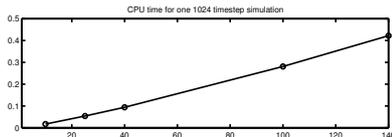


Figure 9: This is the simulation time for one 1024 simulation steps evaluation of the system. It is noticed that the simulation time does not drop even if the amount of used memory is higher than the available in the machine (768 Mb). This gives a hint that the memory problem of figure 8 is solvable.

4. DISCUSSION

There are several aspects of the current implementation of the simulation software for neuro mechanical networks. One is the memory requirement. This will need to be addressed in the future. Since most of computational time is required for the simulation process one can think of several improvements to the current approach. The comparisons between different solutions during the optimization may very well be performed at each time step of the simulation and then it becomes possible to abort a simulation that has already shown a worse goal function value than the currently best one. The use of multi functional optimization is also interesting. The optimization scheme very often requires a population size proportional to the number of design variables times some constant. Typically the memory requirement in these applications follows a $50N_a^2$ scaling.

Another interesting area to improve the simulation time further is to make use of different time step in different parts of the structure. The equations 11 and 12 gives a clear hint of that. At the same time this will most likely be quite difficult to handle if the stiffness of the actuators are one of the objectives of the training or adoption phase.

5. CONCLUSIONS

A first attempt to simulate networks of self organizing mechanical structures has been presented. The important feature of linear scaling with the number of actuators reveals a promising future application for this scheme as a synthesis design tool.

The exponential requirement for computer memory needs to be further studied. The use of bidirectional elements as the foundation element of the network has proved to give fast and accurate simulation results that can be used for optimization techniques and selection schemes. This forms an evolutionary design process to self-organizing structures. The memory requirement shows that it is possible to train a neuro mechanical network of about 1000 nodes on a regular 32-bit PC.

REFERENCES

1. Adries P. Engelbrecht, *Computational intelligence, an introduction*, Wiley, University of Pretoria, South Africa, 2002.
2. Simon Haykin, *Neural networks, a comprehensive foundation*, second edition ed., Prentice Hall, 1999.
3. Petter Krus and Matts Karlsson, *Neuro-mechanical networks, self-organising multifunctional systems*, PTMC2002 (Edge, ed.), vol. 1, CRC Press, Sept 2002, pp. 22–44.
4. L.D. Landau and E.M. Lifshitz, *Statistical physics: Part1*, 3rd ed., Pergamon Press, 1980.
5. Jonas Larsson, *User's guid to hopsan, an integrated simulation enviornment*, Department of Mechanical Engineering, Linköping Institute of Technology, August 2002, Available at hydra.ikp.liu.se.
6. G. Parisi, *Statistical field theory*, Addison-Wesley, Reading, MA, 1988.
7. Magnus Sethson, *Complex cavity analysis : analytical fluid-power models using cad information*, Dissertations, no. 576, Linköping studies in science and technology. Dissertations, June 1999.
8. Magnus Sethson, Matts Karlsson, and Petter Krus, *Neuro-mechanical networks as an architecture for system design*, Computational Synthesis, AAAI, March 2003.



Magnus Sethson, PhD. Assistant Professor at Division of Engineering Systems, Department of Mechanical Engineering, Linköping University, Sweden. magse@ikp.liu.se. Magnus research and engineering studies originates from his special interest in geometry and mechatronics. After a stay within the military aviation industry as a CAD-systems programmer he started his M.Sc. in the late of the eighties. In 1999 he presented his PhD. thesis on the topic "Complex Cavity Analysis" covering different automatic dimensional reduction schemes for CAD-systems. The current research interest includes evolutionary design of mechatronic systems, multi-actuator systems and their relation to geometric information systems. He is also lecturing in mechatronics and motion control systems. Current research projects focus on autonomous vehicle control and steer by wire systems. He is also a keen programmer within many technical areas and maintains some open source projects within genetic algorithms. In his spare time he is most often found behind the camera as an amateur photographer.

RAPID PROTOTYPING OF HUMAN INTERFACE TECHNOLOGIES USING SIMULATION

MARIA F. GRABOVAC, DAVID A. CRAVEN and JAMES W. MEEHAN

Defence Science and Technology Organisation, Australia
maria.grabovac@dsto.defence.gov.au

Abstract: This paper reports the use of simulation to perform rapid prototyping of advanced control and display technologies (direct voice input, 3-dimensional sound and helmet mounted display) for fast-jet aircraft. By integrating the new prototypic systems into a research simulator and making rapid changes to the direct voice input, experienced pilots were able to use the systems in simulated flight, gaining insights into, and providing expert comments on their future application. The rapid-prototyping approach was shown to be useful in establishing how the technologies might be employed in future cockpit systems.

keywords: simulation; rapid-prototyping; direct voice input; human interface technologies

Introduction

A research simulator was used to perform rapid prototyping of advanced control and display technologies for future fast-jet aircraft cockpits. Rapid prototyping is a process by which the prototypes are developed by making successive changes to models in a simulation environment. The technique is widely used in the design, construction and even testing of engineered systems, including automobiles and aircraft [Boeing, Caltech 1997], [Hardtke, 2001] and [Lind et al, 2000].

The general purpose of the study was to investigate principles for the interoperability of advanced aircraft control and display systems. The three systems involved in this study were: a voice-recognition system referred to here as direct voice input (DVI); a spatially-encoded auditory signal generator, incorporating 3-dimensional sound (3DS); and a visual helmet-mounted display (HMD). The DVI prototypic voice-recognition system enabled the pilot to use voice commands to select and control aircraft systems. The 3DS was a spatially-encoded sound profiling system that enabled localisation of sound played in the simulator. The visual symbology was projected on the HMD. The monocular display projected symbology in the line of sight of the pilot's right eye.

The objective of the study was aimed primarily at assessing the interoperability of the systems in simulated flight using rapid prototyping. This required integration of the systems into a flight simulator. The prototypic equipment provided for the simulation consisted of individual closed-system technologies. The core of the ensemble was non-modifiable, while the modifiable sections governed the display of the systems including aspects of HMD symbology, 3DS and DVI; new commands could be created or the functionality of

existing commands could be changed. The DVI was tightly coupled with both the HMD and 3DS; however, this paper focuses on the DVI system. Voice commands enabled 3DS and HMD displays to be executed on demand, hence the description of the rapid prototyping of DVI will also reveal aspects of its interoperability with the other prototypic systems.

The general objective of an input system operated by the pilot's voice is to enable the pilot to control the aircraft more intuitively. There are putative conceptual advantages to this form of control such as time efficiency, i.e. it can be faster to call up a command than it is to scroll through pages of menus using switchology. The second is the ability to control the aircraft or its subsystems while the pilot's hands (or eyes) are otherwise occupied.

Direct Voice Input

The systems were integrated in the simulator as is shown in Figure 1. Since the purpose of the exercise was to explore the possibilities of the technologies and make modifications in response to expert comments, the systems had to be integrated so that making changes "on the fly" was acceptable. The new systems were a stand-alone avionics set, having a single point of contact with the simulator via a TCP/IP connection over ethernet [Robbie, 2001]. The communications protocol was fixed before the system was flown in the simulator; however this was flexible enough to permit incorporation of new DVI commands as required. On the simulator side, the "HMI Suite Interface" module received commands from the DVI, and fed back information to drive the HMD and 3DS displays. The "HMI Suite Controller" was at the other end of this connection, where all of the processing for the new systems took place. The HMI suite could be taken off line, modified, and re-connected while the simulator was still being flown.

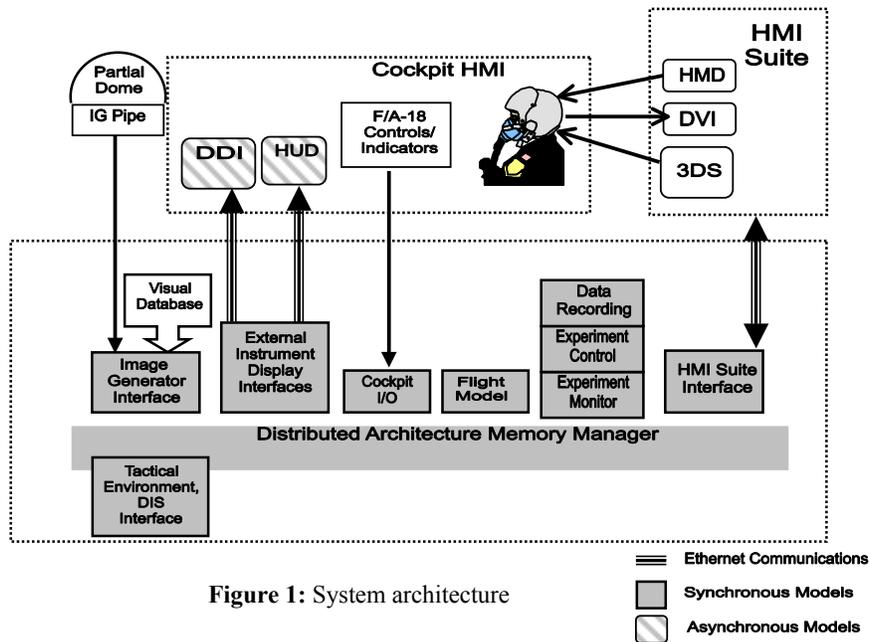


Figure 1: System architecture

DVI was tested in the simulator both as a stand-alone system and as part of the integrated suite in order to verify its functionality and test its compliance with performance specification.

Testing the system in the simulator revealed reduced reliability of the DVI compared with its performance in voice training. Among the problems identified were consistency in switchology, the voice-training environment, and software. A computer joystick was used in training the DVI, whereas a cockpit throttle communications switch was used in the simulation environment. The voice training was conducted at a computer console in a quiet location in the simulator facility, and the oxygen mask incorporating the microphone was held at the pilot's mouth. There was more background noise in the simulation environment, and the oxygen mask was mounted on the helmet, positioning it at a slightly different distance from the mouth. This may have resulted in a marginal change in the acoustical environment. The software was also tested to determine whether it performed better with or without noise-reduction. These potential causes of reduced recognition were tested systematically both in and out of the simulator.

It was noted that the voice-recognition software using noise reduction performed better overall than without noise reduction in both the simulation and training acoustic environments. It also seemed that the use of the cockpit button yielded more consistency in recognition rates than the use of the joystick, but this may have been due to the pilots' familiarity with the cockpit controls. In spite of these changes it was found that the recognition rate still did not reach the reliability level required. This

suggested that the execution of the algorithm in the software may have been involved. Recognising the link between the algorithm's pattern recognition phase and the need for a phonetically diverse syntax, the syntax structure was identified as a possible cause of errors. In order to understand the types of errors that were being made, it was necessary to examine the algorithm used in the system.

Three main algorithms used in voice recognition systems are: Dynamic Time Warping (DTW), Hidden Markov Models, and Neural Networks. The algorithm that was used in the DVI system was a speaker-dependent version of DTW. This type of voice recognition algorithm uses pattern-matching techniques to achieve voice recognition, and hence requires a syntax. The system needs to be trained in the syntax with the user's voice to create templates for recognition [Rabiner and Juang, 1993]. It was therefore necessary to assess the syntax, make changes and then examine the effect of the changes on the performance of the DVI system. This approach to "training" DVI fitted in very well with the intended rapid-prototyping approach of the study whereby rapid changes would be made to the systems involved, allowing pilots to assess their interoperability and potential during simulated flight. By enabling pilots to modify the functionality of some DVI commands, they were able to interact with the systems during flight, develop ideas and suggest new functions to implement and test "on the fly".

Pronunciation and Articulation

Other factors that influence a DVI syntax are articulation and pronunciation. Three different

accents were encountered during the study: French, New Zealand and Australian. There are differences in Australian and New Zealand pronunciation, particularly in the articulation of vowels, and there are other differences between English spoken by native and non-native speakers. This added to the complexity in establishing a syntax whose commands differed enough linguistically and phonetically that the commands would not be confused.

Example of syntactic development

In navigating an aircraft, the pilot flies in straight lines between predetermined points called waypoints. A waypoint has map co-ordinates that correspond with longitude and latitude. The aim is to turn the aircraft on a waypoint and fly on a new bearing to the next. In existing layouts the pilot is often required to select and update waypoints manually, but doing this by voice command to the flight computer would allow the pilot’s hands to remain on the throttle and stick controls. As navigation is not normally a time-critical function, voice commands are potentially well adapted to this function.

A typical command to navigate the aircraft could be “waypoint five”. In the DVI system, this call would display appropriate symbology on the HMD, such as an arrow pointing to the location of the waypoint. In the course of a mission, there would be many waypoints and the pilot would have to navigate from one to the next. The syntax for the DVI included navigational commands of the type “waypoint XX” (where XX represented numerals). This command would bring up the symbology for the nominated waypoint. The numerals were entered into the DVI syntax as English text and pronounced by the pilot accordingly (“one”, “two”, “three”, etc.). However, some commands (“waypoint nine” and “waypoint five”) were confused by the voice recognition system. These numerals were therefore changed to NATO-compliant phonetic translations, (“wun”, “too”, “tree”, etc.). This change was made to ensure uniformity in pronunciation during training, and again during flight. Hence, the numeral “five” changed to “fife”, and “nine” changed to “niner”. This transformation increased the number of syllables in “nine” and shortened the syllable length in “five”. It was observed that this change enabled the pattern-matching algorithm to use the length of the command during its pattern-matching stage as well as the spectral content of the commands. This increased the rate of recognition for waypoint calls. Pilots were then more confident to experiment with the calls and explore the potential of the systems more fully.

In addition to changes in the syntax, changes were made in display of functions resulting from DVI commands. An example was a command to display a heading tape, pitch, roll and altitude-above-ground in the HMD. Initially the heading tape displayed the heading of the aircraft regardless of the direction of the pilot’s head. Pilots suggested that the heading tape could be changed to take account of the direction the pilot was looking. The associated voice command, “call attitude” remained, but the functionality of the HMD symbology was altered accordingly.

DTW is a spectral-content-over-time pattern-recognition algorithm. An inherent problem in such algorithms is the incorrect recognition of words due to differing pronunciation, for example, when under stress, high g, or with a dry mouth. Depending on where emphasis is placed on a word, the spectral content can differ from one utterance to the next. Therefore, if a command were emphasised differently during training than when the command was used in flight, the next closest match may be returned instead of the correct one. The voice-recognition algorithm (speaker-dependent DTW) required the syntax to be trained to the individual’s voice prior to recognition. Pilots sometimes found it difficult to remember the exact intonation used in training. In some cases this led to mispronunciation and hence misinterpretation. To improve this, each pilot was asked to use a more “robot-like” manner of speaking when training, and during runs. An associated change made to the syntax involved the addition of the primer “call” to every DVI command. The primer served two purposes: the first was to give the pilot a constant word to use to initiate the command; the second was to lengthen the command so that the pilot could more easily adopt the same rhythm used in recording the training commands. Hence, commands such as “stores” became “call stores”, and “wingman” became “call wingman”.

Numerous such changes were made to the syntax to improve the performance of the DVI system during rapid prototyping. Figure 2 shows examples of the structure of the syntax before and after the changes.

Initial syntax	Final syntax
wingman	call wingman
waypoint one	waypoint wun
next waypoint	call next waypoint
82X	call 82X-ray

Figure 2: Examples of the syntax before and after changes

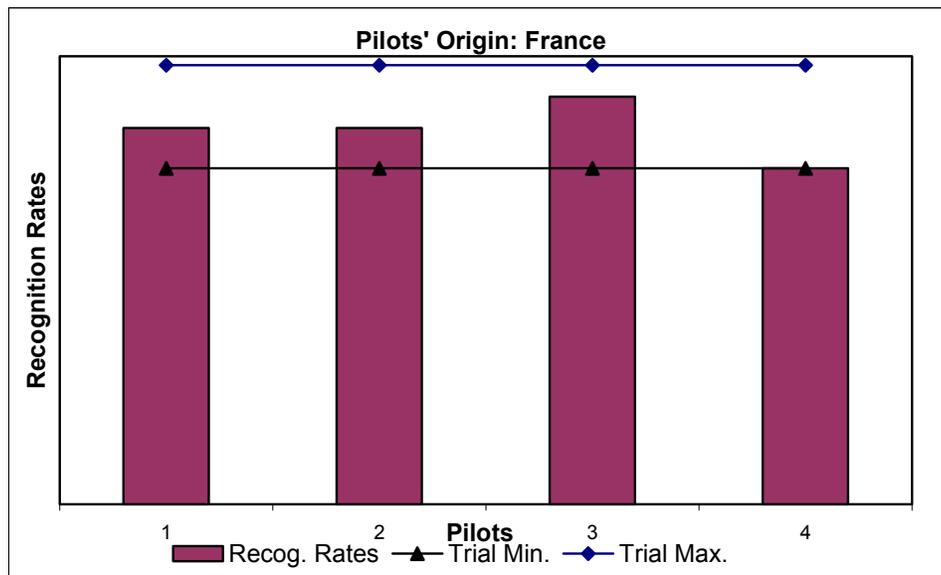


Figure 3: Recognition rates for French pilots, original syntax

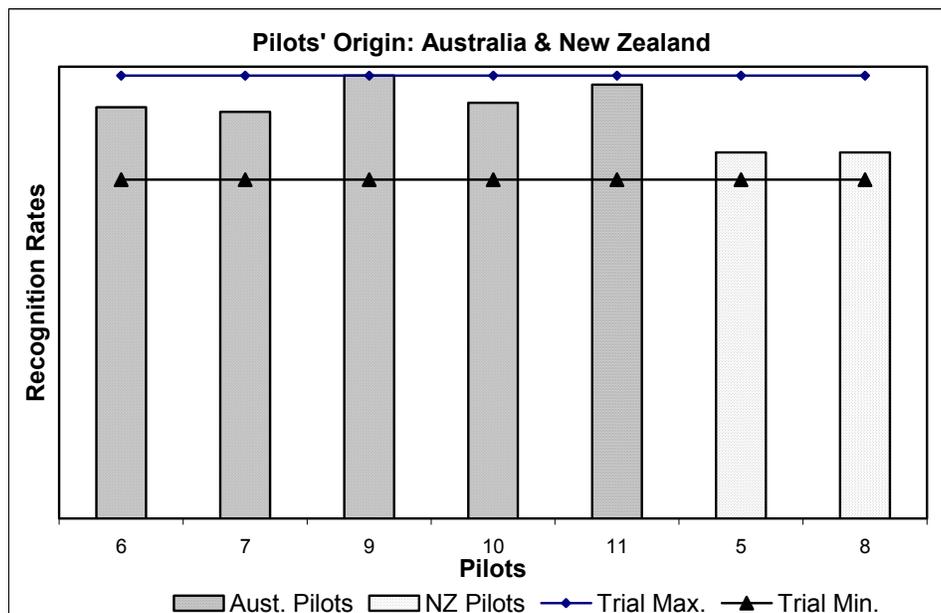


Figure 4: Recognition rates for New Zealand and Australian pilots, revised syntax

Pilot Nationality

The changes to the syntax may have impacted differently on pilots with varying nationalities. Although the study was not designed to investigate this aspect of DVI, some data were gathered that are interesting to compare. French pilots were required to use English throughout, and although fluency varied, all were proficient in English and experienced in the use of English in operational flight. Figure 3 shows recognition rates for the French pilots using the original syntax. The lower

than expected recognition rates prompted restructure of the syntax and investigation of the differences between the training and simulation conditions.

Interestingly, there was also a difference in DVI recognition rates between Australian and New Zealand “native” English speakers. Figure 4 shows the recognition rates for the New Zealand pilots after the syntax had been changed. Figure 4 also shows recognition rates for the Australian pilots that were consistently better than those for the

French and New Zealand pilots. Although the causes of these variable patterns in DVI performance are likely to be multi-factorial, it is worth noting that Australian and New Zealand pronunciation of vowels is the most noticeable difference in spoken English in the region. Hence, it is possible that both structure and content of a syntax in speaker-dependent voice recognition algorithms of this type can influence recognition rates.

Conclusion

This paper has described just one element of an extensive and complex prototyping activity. The DVI example presented shows that this was not a straightforward case of engineering, but that it drew on a number of different disciplines to perform a virtual test of physical prototypes, and a test of their potential use in future aircraft control and information display systems.

A critical element in such an endeavour is how the prototypic system is integrated in the test environment. The interplay between the test system characteristics, the test environment and the method of evaluation together impose limitations on what can be done. Hence, an optimum evaluation using simulation should be based on thorough understanding of the limitations of the engineered components and the scientific objectives of the evaluation itself. Failing to take full account of both will limit the ability to fully exploit the potential of this approach.

Simulation is being used much more extensively today in the design, prototyping, and testing of technologically advanced systems, such as those encountered in aeronautics. Simulation is less expensive than testing in the field. It is also safer and can even permit systems to be tested to destruction – which is neither possible nor desirable in the real world.

Acknowledgement

The prototypic DVI, 3DS and HMD systems employed in the work reported here were provided by Thales Avionics, Bordeaux, France, under a Technical Arrangement between the Délégation Générale pour l'Armement of France and the Defence Science and Technology Organisation of Australia.

References

1. Boeing, Caltech 1997, <http://www.cds.caltech.edu/conferences/1997/vecs/tutorial/Examples/Cases/777.htm>; *the use of dynamic simulations of aircraft flight*.
2. Hardtke F. 2001, "Rapid Prototyping for User-Friendly and Useful Human Machine Interfaces". Proceedings of SIMTECT 2001, Simulation Industry Association of Australia, Canberra, Australia, pp. 239-242.
3. Lind A, Hjorth CG, Hakansson U, Lorenzetto P. 2000, "Evaluation of mock-ups before manufacture of a shield block prototype by powder HIP". [Conference Paper] Elsevier. Fusion Engineering & Design, vol.49-50, Nov. 2000, pp.599-604. Switzerland.
4. Robbie, Andrew. 2001, "Design of an Architecture For Reconfigurable Real-Time Simulation". Proceedings of SIMTECT 2001, Simulation Industry Association of Australia, May 2001, Canberra, Australia, pp. 251-258.
5. Rabiner L, Juang BH. 1993, "Fundamentals of Speech Recognition", Englewood Cliffs, N.J. PTR Prentice Hall, c1993.

Biography



Maria Frances Grabovac obtained a Bachelor of Engineering (Electrical, Hons) and a Bachelor of Arts (French) in 2002 from The University of Melbourne, Australia.

Maria joined the Defence Science and Technology Organisation (DSTO), Air Operations Division as a Simulation Engineer in 2002. Her main areas of interest are head-mounted display symbology and speech recognition in simulated flight. She has worked on 'human-in-the-loop' simulation experiments such as the rapid-prototyping of human-machine interface technologies. Other areas of research include investigation of the effects of visual cues on height perception during simulated flight.

Maria is a Member of the Institute of Engineers Australia and a committee member of the Victorian Branch of the Institute of Electrical Engineers. Currently she is pursuing a Diploma in Modern Languages at Macquarie University.

DIGITAL AUDIO WATERMARKING: SURVEY

MIKDAM A. T. ALSALAMI* and MARWAN M. AL-AKAIDI**

* Computer Science Dept. – Zarka Private University, Jordan

** School of Engineering and Technology - De Montfort University, UK

email: mma@dmu.ac.uk

Abstract: Digital audio watermarking is a technique for embedding additional data along with audio signal. Embedded data is used for copyright owner identification. A number of audio watermarking techniques are proposed. These techniques exploit different ways in order to embed a robust watermark and to maintain the original audio signal fidelity. This paper makes a tutorial in general digital watermarking principles and focus on describing digital audio watermarking techniques. These techniques are classified according to the domain where the watermark is embedded.

Keywords: Digital watermarking, audio, copyright protection.

1. INTRODUCTION

As digital multimedia works (video, audio and images) become available for retransmission, reproduction, and publishing over the Internet, a real need for protection against unauthorized copy and distribution is increased. These concerns motivate researchers to find ways to forbid copyright violation. The most promising solution for this challenging problem seems to lie in information hiding techniques. Information hiding is the process of embedding a message into digital media. The embedded message should be imperceptible; in addition to that the fidelity of digital media must be maintained.

Information hiding is unlike cryptography. In cryptographic techniques significant information is encrypted so that only the key holder has access to that information, once the information is decrypted the security is lost. In information hiding, message is embedded into digital media, which can be distributed and used normally. Information hiding doesn't limit the use of digital data.

channel will notice the transmitted media, but he/she will never perceive the buried secret message inside this media. Figure 1.1 illustrates a simple steganographic system. In this system the message m is embedded into the Cover-object C (could be image, audio or video) to produce the Stego-object S that should has the same fidelity of C . The Cover-Object is only used for the Stego-object generation and is then discarded. The embedding operation is parameterized by the key k that is known for both ends of communication: sender and receiver. On receiver side the buried message is extracted from Stego-object in detection process. Embedding message should be perceptually and statistically undetectable for the warden. An ideal steganographic system would embed a large amount of information perfectly securely with no visible degradation to the cover-object.

Watermarking is very similar to steganography in that both seek to hide information in the Cover-object. However steganography is related to secret

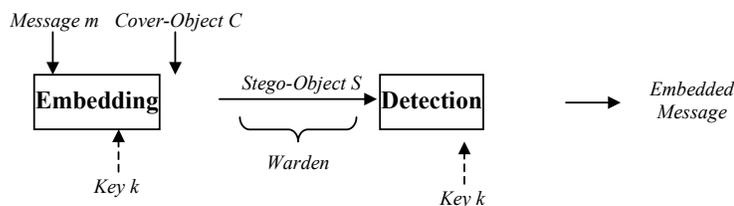


Figure 1.1 Steganographic System

Information hiding can be classified into two types of techniques: Steganography and Watermarking. The main purpose of steganography is to hide the fact of communication. The sender embeds a secret message into digital media (e.g. image) where only the receiver can extract this message. The warden of communication

point-to-point communication between two parties. Thus, steganography techniques are usually having a limited robustness and protect for the embedded information against modifications that may occur during transmission, like format conversion, compression or A/D conversion. On the other hand, watermarking rather than steganography principles

is used whenever the media is available to parties who know the existence of the embedded information and may have interest removing it. Thus, watermarking adds additional requirements of robustness. An ideal watermarking system would embed information that could not be removed or altered without making significant perceptual distortion to the media. A popular application of watermarking is to give a proof of ownership of digital data by embedding copyright statements.

This paper is organized as follow. Section 2 describes the modules of watermarking systems and the function of each module. Sections 3 and 4 are to explain the applications and requirements of digital watermarking. Section 5 covers digital audio watermarking techniques through subsections. Finally, conclusions and general work frame for audio signal are presented.

2. WATERMARKING SYSTEM MODULES

A watermarking system consists of three modules that are watermark signal generation module, watermark embedding module and watermark detection module. Watermark signal is generated by using a non-invertible function that takes, as an input, a watermark key. In some systems the host signal (cover-object) is taken into account when watermark is generated. This will help watermark generator in producing an imperceptible signal-dependent watermark.

Watermark embedding is performed in time domain or in transform domain (DFT, DCT, DWT, ...etc) using a suitable embedding rule (e.g. addition or multiplication). Finally, watermark is detection is performed by some sort of correlation detector or statistical hypothesis testing, with or without resorting to the original signal.

3. DIGITAL WATERMARKING APPLICATIONS

The requirements that watermarking system has to comply with are always based on the application. Thus, before we review the requirements and design considerations, we will present the applications of watermarking [Cox et al, 2002; Katzenbeisser and Petitcolas, 2000]:

3.1 Copyright protection

Copyright protection is the most important application of watermarking. The objective is to embed information identifies the copyright owner of the digital media, in order to prevent other parties from claiming the copyright. This application requires a high level of robustness to ensure that embedded watermark cannot be removed without causing a significant distortion in digital media. Additional requirements beside the robustness have to be considered. For example, the

watermark must be unambiguous and still resolve rightful ownership if other parties embed additional watermarks.

3.2 Fingerprinting

The objective of this application is to convey information about the legal recipient rather than the source of digital media, in order to identify single distributed copies of digital work. It is very similar to the serial number of software product. In this application a different watermark embedded into each distributed copy. In contrast the first application where only a single watermark is embedded into all copies of digital media. As well as copyright protection application of watermarking, fingerprinting requires high robustness.

3.3 Content Authentication

The objective of this application is to detect modification of data. This can be achieved with so-called fragile watermark that have a low robustness to certain modification (e.g. Compression).

3.4 Copy Protection

This application tries to find a mechanism to disallow unauthorized copy of digital media. Copy protection is very difficult in open systems; in closed system, however, it is feasible. In such systems it is possible to use watermarks to indicate the copy status of the digital media (e.g. copy once or never copy). On the other side, copy software or device must be able to detect the watermark and allow or disallow the requested operation according to the copy status of the digital media being copied.

3.5 Broadcast Monitoring

Producers of advertisements or audio and video works want to make sure that their works are broadcasted on the time they purchase from broadcasters. The low-tech method of broadcast monitoring is to have human observers watch the broadcasting channels and record what they see or hear. This method is costly and error prone. The solution is to replace the human monitoring with automated monitoring. One method of automated broadcast monitoring is to use the watermarking techniques. With watermarking we can embed an identification code in the work being broadcasted. A computer-base monitoring system can detect the embedded watermark, to ensure that they receive all of the airtime they purchase from the broadcasters.

4. PROPERTIES OF DIGITAL WATERMARKING

Watermarking systems can be characterized by a number of properties [Cox et al, 2002; Katzenbeisser and Petitcolas, 2000]. The relative importance of each property depends on the requirements of the system application. The properties being discussed in this section are

associated with watermark embedder, watermark detector, or both.

4.1 Embedding Effectiveness

The effectiveness of a watermarking system is the probability that the output of the embedder will be watermarked. The cover work is said to be watermarked when input to a detector result in positive detection. The effectiveness of a watermarking system may be determined analytically or empirically by embedding a watermark in a large number of cover works and detect the watermark. The percentage of cover works that result in positive detection will be the probability of effectiveness.

4.2 Fidelity

In general, the fidelity of a watermark system refers to the perceptual similarity between the original and the watermarked version of the cover work. However, watermarked work may be degraded in the transmission process prior to its being perceived by a person, a different definition of fidelity may be more appropriate. We may define watermarking system fidelity as a perceptual similarity between the unwatermarked and watermarked works at the point at which they are presented to a viewer.

4.3 Data Payload

Data payload refers to the number of bits a watermark embeds in a unit of time or works. For audio, data payload refers to the number of embedded bits per second that are transmitted. Different applications require different data payload. For example, Copy control applications may require a few bits embedded in cover works.

4.4 Blind or Informed Detector

We refer to the detector that requires the original, unwatermarked work as an informed detector. Informed detectors may require information derived from the original work rather than original work itself. Conversely, detectors that do not require the original work are referred to as blind detectors. Informed detector has a good performance in watermark extraction. However, this will result in a huge number of original works have to be stored.

4.5 False Positive Rate

A false positive is the detection of a watermark in a cover work that does not actually contain one. When we talk of a false positive rate, we refer to the number of false positives we expect to occur in a given number of runs of the detector.

4.6 Robustness, Security and Cost

Robustness refers to the ability to detect the watermark after common signal processing operations. Audio watermarking needs to be robust to temporal filtering, A/D conversion, time scaling, etc. not all applications of watermarking require all

the forms of robustness. This depends on the nature of application of watermarking system.

The security of a watermark refers to its ability to resist hostile attacks. Hostile attack is the process specifically intended to thwart the watermark's purpose. The types of attacks can fall in three categories: unauthorized removal, unauthorized embedding, and unauthorized detection.

The Cost of watermarking system refers to the speed with which embedding and detection must be performed and the number of embedders and detectors that must be deployed. Other issues include the whether the detector and embedder are to be implemented as hardware device or as software application or plug-ins.

5. DIGITAL AUDIO WATERMARKING

Watermarking digital media has received a great interest in the literature and research community. Most watermarking schemes focus on image and video watermarking. A few audio watermarking techniques have been reported. Digital audio watermarking is the process of embedding a watermark signal into audio signal. Audio watermarking is a difficult process because of the sensitivity of Human Auditory System (HAS).

The requirements mentioned earlier are common to both image and audio watermarking techniques. Despite their similarities, audio and still image watermarking systems exhibit significant differences. First of all, the fact that images are two-dimensional signals provides attackers with more ways of introducing distortions that might affect watermark integrity e.g. scaling, rotation or removal of rows/columns. Audio watermarking methods need not to deal with such attacks, as audio is a one-dimensional signal. Due to the difference between HAS and Human Visual System (HVS), different masking principles should taken into account in each case.

Digital audio watermarking techniques can be classified according to the domain where the watermarking takes place. The following sections will discuss audio watermarking techniques and classify them to four categories.

5.1 Frequency Domain Audio Watermarking

Audio watermarking techniques, that work in frequency domain, take the advantage of audio masking characteristics of HAS to embed an inaudible watermark signal in digital audio. Transforming audio signal from time domain to frequency domain enables watermarking system to embed the watermark into perceptually significant components. This will provide the system with a high level of robustness [Cox et al, 1997], because of that any attempt to remove the watermark will

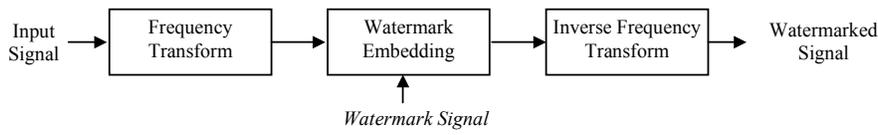


Figure 5.1
Watermarking in Frequency Domain

result in introducing a serious distortion in original audio signal fidelity.

The input signal is first transformed to frequency domain where the watermark is embedded, the resulting signal then goes through inverse frequency transform to get the watermarked signal as output as shown in Figure 5.1.

Watermark can be embedded into frequency domain components by mean of different methods, Cox and et al [Cox et al, 1997] proposed the use of spread spectrum technique in frequency domain. In spread spectrum communication, one transmits a narrowband signal over a much larger bandwidth such that the signal energy present in any single frequency is imperceptible. Similarly the watermark is spread over very many frequency components so that the energy of any component is very small and certainly undetectable. In this method the frequency domain of cover signal is viewed as a communication channel and the watermark is viewed as a signal that is transmitted through it. Attacks and unintentional signal distortions are thus treated as noise that the transmitted signal must be immune to. They claim that in order for the watermark to be robust, watermark must be placed in perceptually significant regions of the cover signal despite the risk of potential fidelity distortion. Conversely if the watermark is placed in perceptually insignificant regions, it is easily removed, either intentionally or unintentionally by, for example, signals compression techniques that implicitly recognize that perceptually weak components of a signal need not be represented.

Suppose that the watermark W consists of a sequence of real numbers, $W = w_1, w_2, \dots, w_n$. In order for W to be embedded into a cover signal, S , a sequence of values, $V = v_1, v_2, \dots, v_n$, is extracted from frequency spectrum of S , the watermark W will be embedded into V to obtain $V' = v'_1, v'_2, \dots, v'_n$. V' is then inserted back to S in place of V to obtain a watermarked signal S' . Only copyright owner knows the locations of V sequence values in frequency spectrum of S . This will ensure the security of the watermark. S' maybe altered, by intentional or unintentional attacks, to produce S^* . Given S and S^* , a possibly corrupted watermark W^* is extracted and compared to W . W^* is extracted by first extracting V^* from S^* and then generating W^* . Figure 5.2 depicts watermark embedding and extraction.

There are three natural formulae for computing V' :

$$\begin{aligned} v'_i &= v_i + \alpha w_i \\ v'_i &= v_i (1 + \alpha w_i) \\ v'_i &= v_i (e^{\alpha w_i}) \end{aligned}$$

α is scaling parameter (controls robustness and fidelity).

There are a number of ways that one can use to evaluate the similarity between two watermarks. A traditional correlation measure can be used, for example. Similarity of W and W^* can be measured by:

$$\text{sim}(W, W^*) = \frac{W^* \cdot W}{\sqrt{W^* \cdot W^*}}$$

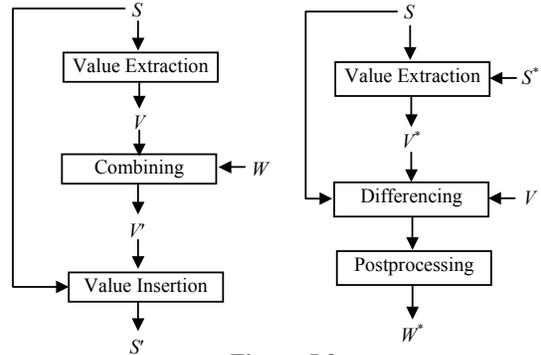


Figure 5.2
Watermark Embedding and Extraction

$$\text{Where } X \cdot Y = \sum_{i=1}^n x_i \cdot y_i$$

Another audio watermarking technique uses statistical algorithm works in Fourier domain [Arnold, 2000; Arnold, 2001]. This method is based on the patchwork algorithm [Bender et al, 1996] and doesn't need the original audio in detection process.

Audio signal is broken into frames; each frame is used to embed one bit. Each frame is transformed to frequency domain using DFT. Assume that the transformed frame contains $2N$ values, and then the embedding process works as follows:

1. Map a secret key and the watermark to the seed of random-number generator. Start the generator to pseudorandomly select two intermixed subsets $A = \{a_i\}_{i=1, \dots, M}$ and $B = \{b_i\}_{i=1, \dots, M}$ of equal size $M \leq N$ from the original set of audio signal frequency spectrum.
2. Alter the selected elements $a_i \in A$ and $b_i \in B$, $i=1, \dots, M$ according to the following embedding function:

$$a'_i = a_i + \Delta a_i \quad \& \quad b'_i = b_i - \Delta b_i$$

Δa_i and Δb_i are two patterns generated by the secret key. There are two patterns for 0 and another

two for 1. We have to select the correct patterns according to the value of the bit being embedding. The alterations of frequency domain coefficients have to be performed in a way that achieves inaudibility. Therefore, Δa_i and Δb_i are driven from psychoacoustics model. Thus, Δa_i and Δb_i are reshaped for each individual frame. For more information about psychoacoustics model see [Painter and Spanias, 2000].

In watermark detection process, hypothesis testing is used. We formulate test hypothesis, H_0 , and alternative hypothesis, H_1 , the appropriate test statistic z will be a function of the sets A and B with probability distribution function PDF $\varnothing(z)$ in the unwatermarked case and $\varnothing_m(z)$ in watermarked case.

H_0 : the watermark is not embedded; z follows PDF $\varnothing(z)$.

H_1 : the watermark is embedded; z follows PDF $\varnothing_m(z)$.

Two kind of error are incorporated in hypothesis testing:

$$I : \int_T^{+\infty} \phi(z) dz = P_I \quad (\text{Type I error})$$

$$II : \int_{-\infty}^T \phi_m(z) dz = P_{II} \quad (\text{Type II error})$$

Hypothesis testing is used in the detection to decide whether the watermark bit is embedded or not. The threshold T is used in the detection step. Detection procedure is as follows:

1. Map the secret key and the watermark to the seed of random-number generator to generate the subset C and D. $C = A$ and $D = B$ if a correct key is used.
2. Decide the probability of correct rejection $1 - P_I$ according to the application and calculate the threshold T from error type I equation.
3. Calculate the sample mean $E(z) = E(f(C,D))$ and choose between two mutually exclusive propositions:

H_0 : $E(z) \leq T$ the watermark bit is embedded.

H_1 : $E(z) > T$ the watermark is not embedded.

Hypothesis testing depends on appropriate test statistic. Two test statistics can be used in watermark detection:

1. The first test statistic uses the function to measure the difference between population means of A and B:

$$z = f(A, B) = \frac{\bar{a}' - \bar{b}'}{\sigma_{\bar{a}' - \bar{b}'}}$$

Therefore the two mutually exclusive propositions become:

$$H_0: \varnothing(z) = N(0,1)$$

$$H_1: \varnothing_m(z) = N(z_m, 1),$$

$$z_m = \frac{k(\bar{a} + \bar{b})}{\hat{\sigma}_{\bar{a}' - \bar{b}'}}$$

Where $N(\mu, \sigma^2)$ is the normal distribution with the mean μ and standard deviation σ , and

$$k = \sqrt{\frac{1}{1 - (z_1 - P_I + z_1 - P_{II})^2 \varepsilon^2} - 1}$$

2. The second test statistic uses another function:

$$z = f(A, B) = \frac{\frac{\bar{a}' - \bar{b}'}{\sigma_{\bar{a}' - \bar{b}'}}}{\frac{1}{2} \frac{\bar{a}' + \bar{b}'}{\sigma_{\bar{a}' + \bar{b}'}}} = 2 \frac{\bar{a}' - \bar{b}'}{\bar{a}' + \bar{b}'}$$

The threshold T must be computed and compared with the mean value calculated by one of the above statistics functions.

It is clear that the detection process doesn't require the original audio signal while it works to detect the statistics changes in the media to determine whether it is watermarked or not.

Further research has been achieved to improve the performance of above watermarking system, for more information see [Hong et al, 2002; Yeo and Kim, 2001]

5.2 Time Domain Audio Watermarking

In time domain watermarking techniques, watermark is directly embedded into audio signal. No domain transform is required in this process. Watermark signal is shaped before embedding operation to ensure its inaudibility (Figure 5.3). The available time domain watermarking techniques insert the watermark into audio signal by simply adding the watermark to the signal.

Embedding a watermark into time domain involves challenges related to fidelity and robustness. Shaping the watermark before embedding enables the system to maintain the original audio signal fidelity and renders the watermark inaudible. As for robustness, time domain watermarking systems use different techniques to improve the robustness of the watermark.

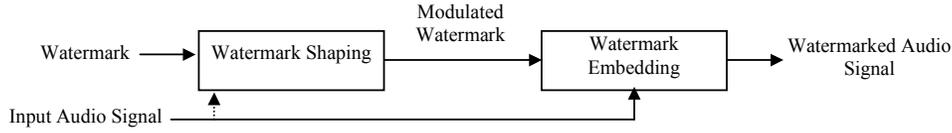


Figure 5.3
Time Domain Watermarking

Working in frequency domain enables watermarking system to embed a robust watermark, while it is possible to identify the most significant components of the cover signal. Also, masking characteristics of audio signal can be exploited, in order to reduce the distortion of embedded watermark.

In this section, two methods for audio watermarking in time domain are shown. The first one presented in [Bassia and Pitas, 1998; Bassia et al, 2001] and in which the watermark signal is modulated using the original audio signal and filtered by lowpass filter to reduce the distortion that might be result from embedding the watermark. The original audio signal is divided into segments and then each segment is watermarked separately by embedding the same watermark. Watermark signal, $w_i \in \{1, -1\}$, $i=0,1,\dots,n-1$ is generated by threshold a chaotic map in a way similar to the one described in [Bassia et al, 2001]. The seed (start point) of the chaotic sequence generator is the watermark key. Using the chaotic sequence generator is to ensure the security of the watermarking system i.e. the sequence generation mechanism cannot be reversed engineered.

Suppose that we have a segment of audio signal $S = s_1, s_2, \dots, s_n$ then the watermarking process begin by modulating the watermark signal w_i by using audio signal S ,

$$w_i' = \alpha |s_i| \oplus w_i \quad i = 0, 1, \dots, n-1$$

Where \oplus denotes a superposition law which can be multiplication, power law, etc, and α is a constant controls the amplitude of the watermark signal. The maximum allowable watermark

amplitude is the limited by the maximum perceived signal distortion.

In next stage, w_i' is shaped using a lowpass Hamming filter of length (order) L :

$$w_i'' = \sum_{l=0}^{L-1} b_l w_{i-l}'$$

where b_l is the filter coefficients. This process results in inaudible watermark signal. Figure 5.4 [Bassia et al, 2001] shows the power spectral density (*PSD*) of two watermark signals, one is shaped and the other is not. It is clear that the unshaped watermark signal is audible while it has a PSD exceeds the power of the original signal in certain frequencies. The PSD of the shaped watermark signal lies underneath the original audio signal in the entire frequency range.

Finally the shaped watermark signal is embedded into audio signal:

$$y_i = s_i + w_i'' \quad i = 0, 1, \dots, n-1$$

It is obvious that the calculation of watermarked sample y_i is based on the neighbors of the sample s_i and the chaotic signal (watermark) w_i .

In detection stage, the received signal, Y , broken in the same way that original signal is broken. Consider the following sum:

$$C_k = \frac{1}{n} \sum_{i=0}^{n-1} y_{(i+k) \bmod n} w_i$$

C_k is the correlation of W with Y , evaluated for all possible circular shift of Y . By substitution and

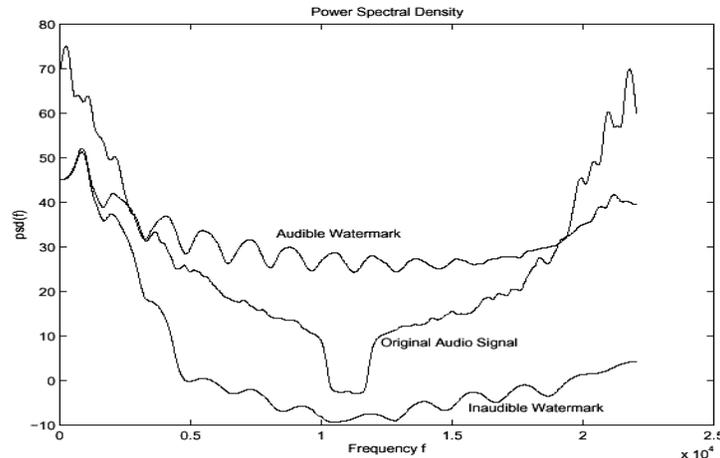


Figure 5.4
Power spectral density of two watermark and original signals

rearranging the above equation we get:

$$C_k = \frac{1}{n} \left(\sum_{i=0}^{n-1} s_{(i+k) \bmod n} w_i + \sum_{i=0}^{n-1} w''_{(i+k) \bmod n} w_i \right)$$

The expected value of the first sum is zero if either the watermark mean value m_w or the signal mean value m_s is equal to zero. In case m_w is not zero (the number of 1 and -1 is not the same), the quantity $\Delta w = \sum_{i=0}^{n-1} w_i$, must be taken into account. Let us denote by B a set of $N_B = |\Delta w|$ index values for which the corresponding w_i values are equal the -1 or 1 with the most occurrences. It is easy to show that:

$$\sum_{i \in B} w_i = \Delta w$$

Let us denote by A the set of all index values that do not belong to B. obviously, the cardinality of A is $N_A = n - |\Delta w|$ and the following equation holds

$$\sum_{i \in A} w_i = 0$$

So, C_k can be expressed as follows:

$$C_k = \frac{1}{n} \left(\sum_{i \in A} s_{(i+k) \bmod n} w_i + \sum_{i \in B} s_{(i+k) \bmod n} w_i + \sum_{i=0}^{n-1} w''_{(i+k) \bmod n} w_i \right)$$

Let us define the following terms:

$$T_{1,k} = \frac{1}{n} \left(\sum_{i \in A} s_{(i+k) \bmod n} w_i \right)$$

$$T_{2,k} = \frac{1}{n} \left(\sum_{i \in B} s_{(i+k) \bmod n} w_i \right)$$

$$T_{3,k} = \frac{1}{n} \left(\sum_{i=0}^{n-1} w''_{(i+k) \bmod n} w_i \right)$$

It can be easily shown that $E(T_{1,k}) = 0$, where $E()$ denotes the expected value operator. For the term T_2 , it is easy to show that:

$$T_{2,k} = \text{sign}(\Delta w) \frac{1}{n} \left(\sum_{i \in B} s_{(i+k) \bmod n} w_i \right) = \frac{\Delta w}{n} \frac{1}{N_B} \sum_{i \in B} s_{(i+k) \bmod n}$$

Therefore

$$E(T_{2,k}) = \frac{\Delta w}{n} m_s$$

If no watermark has been embedded in the signal, $T_3 = 0$ and thus:

$$C_k \approx T_{2,k} = \frac{\Delta w}{n} m_s$$

On the other hand, if the signal is watermarked

$$C_k \approx T_{2,k} \approx \frac{\Delta w}{n} m_s + \frac{1}{n} \left(\sum_{i=0}^{n-1} w''_{(i+k) \bmod n} w_i \right)$$

For watermark detection we construct the ratio r_k :

$$r_k = \frac{C_k - T_{2,k}}{T_{3,k}}$$

The original signal S is required for evaluation of $T_{2,k}$ and $T_{3,k}$, but it can be replaced by Y without significant error.

The value of r_k is computed for every $k = 0, 1, \dots, n-1$, for all segments. We compute the detection value of the audio segment j as $R_j = \sum_{i=0}^{n-1} r_i$, the final detection value is $R = \sum_{j=0}^{N_s-1} R_j$, where N_s is the number of segments in signal.

The decision about the existence of the watermark is made depending on a threshold value compared with R.

It is clear that this watermarking system is immune against time-shifting and cropping. The fact that C_k is computed for all possible circular shift of Y, ensures synchronization between Y and W will occur for certain value of $k=0, 1, \dots, n-1$.

Another watermarking system uses the HAS masking effects to shape the watermark signal [Boney et al, 1996; Swanson et al, 1998]. Shaping operation is performed in frequency domain, but the shaped watermark is embedded into audio signal in time domain. Watermark is a noise-like sequence generated by using two keys x_1 and x_2 . The first key x_1 is author dependent. The second key x_2 is computed from audio signal that the author wants to watermark. It is computed from the signal using a one-way hash function. The two keys are mapped to pseudorandom number generator to generate a noise-like sequence, watermark. Original audio signal is required in detection process to compute the second key x_2 , and to extract the embedded watermark.

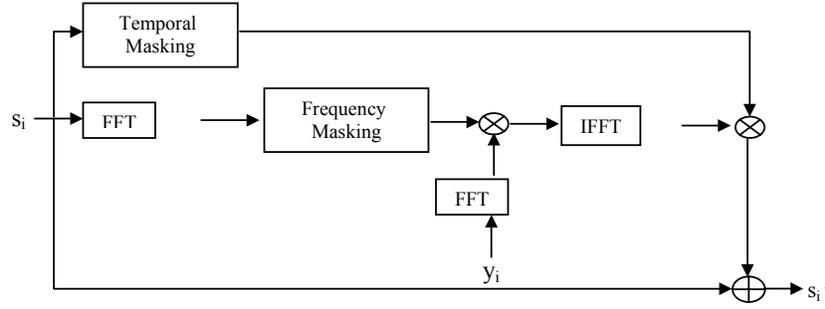


Figure 5.5
Audio Segment Watermarking Procedure

The watermarking process begins with dividing the audio signal into segments, and then each segment is watermarked separately. Suppose that you have a generated watermark y_i , and then the algorithm of watermarking an individual segment, s_i , works as follows:

1. Compute the power spectrum S_i of audio signal segment s_i as follows:

$$S_i = 10 \log_{10} \left[\frac{1}{N} \left\| \sum_{n=0}^{N-1} s_n h(n) \exp(-j2\pi \frac{2i}{N}) \right\|^2 \right]$$

Where $h(n)$ is a Hann window:

$$h(n) = \frac{\sqrt{8/3}}{2} \left[1 - \cos\left(2\pi \frac{n}{N}\right) \right]$$

N is the number of samples in one segment and j is $\sqrt{-1}$

2. Compute the frequency masking threshold M_i of the power spectrum S_i .
3. Use the mask M_i to weight the noise-like watermark, $P_i = M_i * Y_i$, where P_i is the weighted watermark and Y_i is the power spectrum of the watermark signal y_i .
4. Compute the inverse of FFT of the shaped watermark $p_i = \text{IFFT}(P_i)$.
5. Compute the temporal masking t_i of s_i .
6. Use the temporal masking t_i to further shape the frequency shaped watermark to create the final watermark $w_i = t_i * p_i$ of the audio segment.
7. Create the watermarked segment $s_i' = s_i + w_i$.

Figure 5.5 shows a diagram of watermark shaping and embedding.

In detection process, the original audio signal is known. Thus, second key can be computed and then

the watermark signal can be reconstructed. Also the embedded possible distorted watermark can be extracted. Assume that $r_i, i = 0, 1, \dots, N$ is a recovered piece of audio signal, then we can compute $x_i = r_i - s_i$. If r_i has a watermark then $x_i = w_i' + n_i$, where n_i is noise (intentionally or unintentionally added to the watermarked signal). Otherwise, $x_i = n_i$. Similarity between extracted watermark, x_i , and the reconstructed one can be measured by correlation as follows:

$$\text{sim}(x, w) = \frac{\sum_{i=0}^{N-1} x_i w_i}{\sum_{i=0}^{N-1} w_i w_i}$$

Then the value can be compared with a threshold T .

The recovered signal r_i is possible shifted. This leads to lose the synchronization between the extracted watermark and the reconstructed one. In such case we can assume that $r_i = s_{i+\tau} + x_i$, where x_i as mentioned before. τ is a n unknown delay, thus, a generalized likelihood ratio test must be performed to determine whether the audio signal is watermarked or not.

$$\frac{\max_{\tau} \exp(-\sum_{n=0}^{N-1} (r_n - (s_{n+\tau} + w_{n+\tau}))^2)}{\max_{\tau} \exp(-\sum_{n=0}^{N-1} (r_n - s_{n+\tau})^2)}$$

Then, this ratio is compared to a threshold.

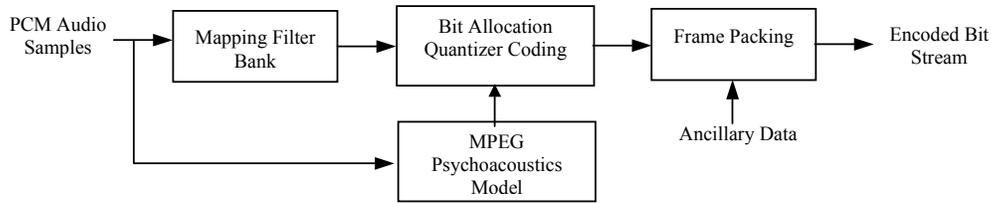


Figure 5.6
Structure of MPEG Audio Encoder

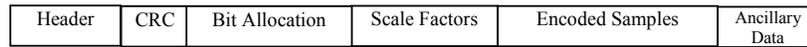


Figure 5.7
Frame Format of MPEG Audio

5.3 Compressed Domain Audio Watermarking

A number of techniques are proposed to embed a watermark signal into MPEG audio bit stream, rather than going through decoding/encoding process in order to apply watermarking scheme in uncompressed domain [Qiao and Nahrstedt, 1999; Neubauer and Herre, 2000a; Neubauer and Herre, 2000b; Neubauer and Herre, 1998]. Such systems are suitable for “pay audio” scenario, where the provider stores audio contents in compressed format. During download of music, the customer identifies himself/herself with his/her unique customer ID, which therefore is known to the provider during delivery. In order to embed the customer ID into the audio data using a watermarking technique, a scheme is needed that is capable of watermarking compressed audio on the fly during download.

MPEG audio compression is a lossy algorithm and uses the special nature of the HAS. It removes the perceptually irrelevant parts of the audio and makes the audio signal distortion inaudible to human ear. For more information about MPEG audio Compression see [Pan, 1995].

MPEG encoding process has the following steps:

1. Input audio samples pass through a mapping filter bank to divide the audio data into subbands (subsamples) of frequency.
2. At the same time, the input audio samples pass through MPEG psychoacoustics model, which creates a masking threshold of audio signal. Masking threshold is used by quantization and coding step to determine how to allocate bits to minimize the quantization noise audibility.
3. Finally, the quantized subband samples are packed into frames (coded stream).

Figure 5.6 shows the basic structure of an MPEG audio encoder.

Filter bank divides the input audio signal into 32 equal-width subbands, then the number of bits used

in quantization is determined upon masking threshold to minimize the audibility of possible distortion maybe introduced by quantization.

The MPEG audio stream consists of frames. Frame is the smallest unit which can be decoded individually. Each frame contains audio data, header, CRC (Cyclic Redundancy Code), and ancillary data. In frame, each subband has three groups of samples with 12 samples per group. The encoder can use a different scale factor for each group. Scale factor is determined upon masking threshold and used in reconstruction of audio signal. The decoder multiplies the quantizer output to reconstruct the quantized subband sample. Figure 5.7 depicts the general format of MPEG frame.

MPEG audio decoding process is simple a reverse of the encoding process. The decoding takes the encoded bit stream as an input, unpacks the frames, reconstructs the frequency samples (subbands samples) using scale factors, and then inverses the mapping to re-create the audio signal samples. This process is depicted in Figure 5.8.

One audio watermarking technique [Qiao and Nahrstedt, 1999] embeds the watermark into scale factors of MPEG audio frames. In this technique, DES encryption algorithm is used in generating non-invertible watermark. Original data is applied into encryption algorithm to get the watermark as follows:

First, a key KEY is selected and for each MPEG audio frame a_j , $j=1, \dots, N$ (number of audio frames), we apply DES with KEY to it to get a random byte sequence RBS :

$$RBS = DES_{KEY}(\text{one audio frame } a_j)$$

Second, let RBS_i be i -th byte of random byte sequence and w_i be the i -th bit of the watermark bit stream, then the watermark can be created by:

$$w_i = \begin{cases} -1 & \text{if } RBS_i = \text{even number} \\ 1 & \text{otherwise} \end{cases}$$

Each scale factor takes 6 bits; therefore, we have as many as 63 levels of scale factors (indexed

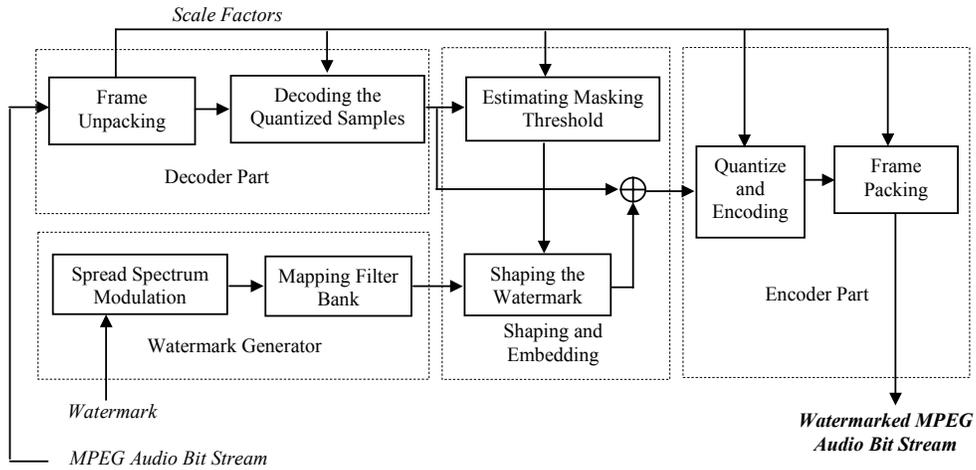


Figure 5.9
MPEG Audio Bit Stream Watermarking

from 0 to 62, 63 is not used by the standard). The level change of scale factor has an auditory effect that the sound becomes stronger when the scale factor level increases, and becomes weaker when the scale factor decreases. Increasing or decreasing scale factor by one level normally cannot be detected by listeners.

Let $ScaleFactor_i(index)$ be the i -th scale factor with the level indicated by $index$ and SW_i be the i -th watermarked one. The watermarking procedure works as follows:

$$SW_i = \begin{cases} ScaleFactor_i(index) & \text{if } index + w_i = -1 \text{ or } 63 \\ ScaleFactor_i(index + w_i) & \text{otherwise} \end{cases}$$

This scheme has drawbacks. The first one is that the scheme doesn't have much data to watermark due to the few number of scale factors in audio frame. Also, the watermark scheme is not robust enough against attacker who is trying lower scale factors by 2 or 3 levels. On the other side, multiple watermarks cannot be applied. The reason is that when multiple watermarks are applied, certain scale factors would be increased by multiple levels and perceptible noise would be introduced.

Another watermarking scheme embeds the watermark into the encoded data. However, changing the all encoded samples shows a perceptible distortion. *Spacing Parameter* sp is introduced to solve this problem. sp is used in way like that every sp samples, we randomly select 1 or 2 samples to be watermarked. The watermark generation procedure will be modified to incorporate spacing parameter:

$$w_i = \begin{cases} -1 & \text{if } RBS_i = 0 \pmod{sp} \\ 1 & \text{if } RBS_i = 1 \pmod{sp} \\ 0 & \text{otherwise} \end{cases}$$

Let $Sample_i$ be the i -th sample in audio frame and SW_i be the i -th watermarked sample. The watermarking will be:

$$SW_i = \begin{cases} Sample_i & \text{if every bit of } (Sample_i + w_i) \text{ is } 1 \\ Sample_i + w_i & \text{otherwise} \end{cases}$$

Both watermarking schemes described above use the concept of spread spectrum watermarking, but through compressed domain.

The original MPEG audio is required in detection process and the watermark can simply be extracted and verified.

Another technique [Neubauer and Herre, 2000a; Neubauer and Herre, 2000b; Neubauer and Herre, 1998] in MPEG audio stream watermarking is to partly decode the input bit stream, embed a perceptually hidden watermark in the frequency domain and finally quantize and code the signal again. Figure 5.9 illustrates a general structure of bit stream watermarking system.

This watermarking system consists of four parts. Each part has a specific function. We can see that this watermarking system has assembled parts of MPEG encoder and decoder, in addition to parts of frequency domain audio watermarking systems (watermark generation and watermark embedding). These parts have been modified in order to enable the system to embed the watermark in subbands samples.

The first part, decoder part, takes MPEG audio bit stream as an input and gives frequency subbands samples as output. This part supplies the other parts with scale factors that are necessary in masking threshold estimation and encoder process.

The second part, watermark generator, is used to convert the watermark to subband representation in order to be ready for embedding. The watermark can be any data provided by copyright owner. The

generated watermark is fed into watermark shaping and embedding part, which in turn, takes the decoded subbands samples and scale factors to estimate the masking threshold of the audio signal and use it in shaping the watermark. The last two parts have much similarity to the technique proposed in [Swanson et al, 1998].

The last part, encoder part, takes the watermarked subbands samples and scale factors. It decodes the samples using the original scale factors and then packs the resulting decoded samples. In order to avoid the possible distortion of requantization, the original scale factor is used and no need to recomputed new scale factors.

The embedded watermark can be detected in uncompressed domain as well as compressed domain. Original audio data is required to extract the watermark and then measure the similarity between the extracted watermark and the original one. The watermark detection in uncompressed domain can be achieved, exactly like the way presented in [Swanson et al, 1998], by using correlation measurement.

5.4 Wavelet Domain Audio Watermarking

Wavelet transform can be used to decompose a signal into two parts, high frequencies and low frequencies. The low frequencies part is decomposed again into two parts of high and low frequencies. The number of decompositions in this process is usually determined by application and length of original signal. The data obtained from the above decomposition are called the DWT coefficients. Moreover, the original signal can be reconstructed from these coefficients. This reconstruction is called the inverse DWT. The process of decomposition is depicted in Figure 5.10. For more information on Wavelet transform, see [Daubechies, 1992; Daubechies, 1988].

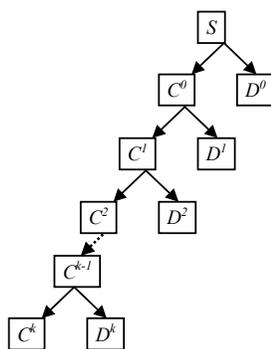


Figure 5.10
Wavelet Decomposition

A method of audio signal watermarking in wavelet domain uses patchwork algorithm [Kim et al, 2002]. In this method, a binary watermark w_i is embedded one bit in one data block. Watermark

bits are locally repeated for the purpose of robustness. Also a number of bits are added in front of watermarks bits to locate the point where the watermark bit is embedded in watermarked signal. These bits are called synchronization bits. For example, with local redundancy rate 3 and synchronization bits 10101011, we change the original watermark as:

$$w_0 w_1 w_2 \dots \rightarrow 10101011 w_0 w_0 w_0 w_1 w_1 w_1 w_2 w_2 \dots$$

Suppose that B is a block of audio signal being watermarked, we use DWT to have $D^0, D^1, D^2, \dots, D^k, C^k$, for some integer k. then after patchwork algorithm is used to embed the watermark by

$$P_N = \sum_{i \in I} D_i^k - \sum_{j \in J} D_j^k$$

artificially modifying a patch value P_N as

Where I and J are two subset of indexes randomly generated. Proposed algorithm modifies P_N in a way that the modified P_N is deviation away

$$D_i^k \rightarrow D_i^k + \delta, \quad D_j^k \rightarrow D_j^k - \delta \quad \text{if } w_n = 1$$

$$D_i^k \rightarrow D_i^k - \delta, \quad D_j^k \rightarrow D_j^k + \delta \quad \text{if } w_n = 0$$

from expected. To be specific, we modify some wavelet coefficients in D^k as

For $i \in I$ and $j \in J$, w_n is a watermark bit being embedded and δ is a real number.

Different two subsets of indexes I and J are

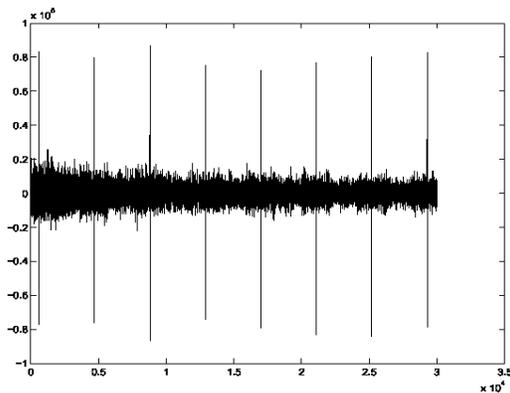


Figure 5.11
 A_N^t for watermarked Audio Signal

used to embed the synchronization bits for security purpose.

In detection process, $\Delta_N^t = P_N^{t+1} - P_N^t$, are computed, where P_N^{t+1} and P_N^t are two patch values of block B_{t+1} and B_t respectively. Figure 5.10 [Kim et al, 2002] shows Δ_N^t for watermarked audio signal [Kim et al, 2002].

The peaks shown in this figure refers to the watermark bits locations in audio signal. Then

detection is made according the following criteria, for $\beta > 0$:

- If $\Delta_N^t > \beta N\delta$ and $\Delta_N^{t+1} < 0$ then 1 is detected in block B_t .
- If $\Delta_N^t < -\beta N\delta$ and $\Delta_N^{t+1} > 0$ then 0 is detected in block B_t .
- If previous two conditions are not satisfied, then no watermark bit is detected in this block.

Synchronization bits must be found first to determine the location of watermark bits.

This watermarking system shows a high performance in synchronization and resisting time shifting attack [Kim et al, 2002].

7. CONCLUSIONS

All watermarking systems are designed to achieve one goal that is embedding a hidden robust watermark into digital media. These systems have to satisfy two conflicting requirements. First, watermark must be immune against intentional and unintentional removal. Second, watermarked signal should maintain a good fidelity, i.e. watermark must be perceptually undetectable. To accomplish this task, variety of techniques has been exploited, and different domains are involved to enhance a certain application of watermarking and/or improve fidelity and robustness of watermarked signal.

However, watermarking systems have a number of differences. These differences can be considered in evaluating performance of watermarking systems and suitability of these systems for a specific application. These differences can be explained as follows:

1. Some audio watermarking systems require the original audio signal, or any information derived from it, to be presented in detection process. This will leads to a large number of original works have to be stored and searched during detection.

Systems that require the original audio signal are not suitable for some type of applications, in case that detection process has no access to the original work or it is not acceptable to disclose it. On the other hand, presenting the original signal yields in efficient watermark extraction consequently efficient detection.

Audio watermarking systems that are based on patchwork algorithm use a statistical detection process (hypothesis testing) and don't need the original audio for detection purpose. The most techniques that are base on correlation measurement of similarity require that signal except method presented in [Bassia and Pitas, 1998; Bassia et al, 2001].

In spite of that a number of audio watermarking techniques require only the watermarked signal in detection watermark key is needed in both embedding and detection.

2. In order to maintain the watermark security, watermark would be embedded into selected regions of some domain transform of audio signal. These regions are selected randomly by generating a sequence of indexes. Sequence generation is parameterized by a key called watermarking key. This key is required in both embedding and detection.

In some watermarking systems, watermarking key is used to generate the watermark itself. In this case, the watermark would be a random sequence of bits or digits generated by some sort of algorithms ensure non-invertibility of watermark in order to maintain the security of watermarking key.

Watermarking key could be provided by the copyright owner or a combination of information provided by him/her and information derived from original signal. In such case, original signal will be required in detection process for key generation purpose. In all scenarios, the key is used as a seed for random number generator.

Sometimes, disclosing the watermarking key or having an access to it becomes impossible. Thus, using the same key in detection and embedding will not be acceptable. A solution to such problem could be found in using two keys, one for embedding and another for detection [Hong et al, 2002] (i.e. public-key or asymmetric watermarking system).

3. During embedding process, original audio signal is divided into frames. Then after, each frame is watermarked separately. Some watermarking systems embed the same watermark into a number of frames to enhance watermark robustness. But, in other systems each frame is watermarked with different watermark.
4. Because of sensitivity of HAS, watermark signal must be shaped to rent it inaudible. Masking characteristics of audio signal can be used for this purpose. Psychoacoustics MPEG model is commonly used to calculate masking threshold that is used in weighting the watermark. In some other audio watermarking systems, different techniques are used. These techniques use the original audio signal in modulating the watermark. Therefore; the amplitude of watermark signal is controlled by amplitude of audio signal. Watermark shaping process may effect the existence of the

watermark in cover work, consequently, false negative rate will be increased.

A general work frame for digital audio watermarking systems can be concluded as follows:

1. Watermarking system should be able to embed any set of data in to audio signal, and the detector should be able to retrieve the embedded data (i.e. not just report that watermark is presented or not)
2. Watermark embedded (detection) module should be independent of mode of operating. (e.g. the same watermark is embedded into multiple frames of audio signal or different watermark is embedded into each frame).
3. Watermarking key generation should be independent of watermark embedding and detection (e.g. embedding and detection will not be effected whether original signal is involved in key generation or not).

The above points enables audio watermarking system to be suitable for variety of application and make it possible to put standards (e.g. [SDMI, 2000]) and evaluation benchmarks.

7. REFERENCES

1. Arnold M. 2000, "Audio Watermarking: Features, Applications and Algorithms". *Multimedia and Expo. IEEE international Conf.*, Vol. 2, pp. 1013-1016.
2. Arnold M. 2001, "Audio Watermarking", *Dr. Dobb's Journal*, Vol. 26, Issue 11, pp. 21-26.
3. Bassia P. and Pitas I. 1998, "Robust Audio Watermarking in the Time Domain". *Signal Processing IX, theories and applications: proceeding of Eusipco-98, Ninth European Signal Processing Conf.*, Greece, pp. 8-11.
4. Bassia P., Pitas I., and Nikolaidis 2001, "Robust Audio Watermarking in Time Domain", *IEEE Trans. On Multimedia*, Vol. 3, pp. 232-241.
5. Bender W., Gruhl D., Morimoto N. and Lu A. 1996, "Techniques for Data Hiding", *IBM Systems Journal*, Vol. 35, No. 3&4, pp. 313-335.
6. Boney L. Tewfik A. H. and Hamdy K. N. 1996, "Digital Watermarking for Audio Signal". In *Proc. of EUSIPCO '96*, Sep., Vol. III, pp. 1697-1700.
7. Cox I. J., Kilian J. Leighton F. T. and Shamoon T. 1997, "Secure Spread Spectrum Watermarking for Multimedia". *IEEE Trans. On Image Processing*, Vol. 6, No. 12, pp. 1673-1687.
8. Cox I. J., Miller, M. L. and Bloom J. A. 2002, "Digital Watermarking". *Morgan Kaufmann Publishers*, USA.
9. Daubechies I. 1988, "Orthonormal Bases of Compactly Supported Wavelets". *Comm. Puse and Appk. Math.*, Vol. 41, pp. 909-996.
10. Daubechies I. 1992, "Ten Lectures on Wavelets", *SIAM*, Philadelphia.
11. Hong D. G., Park S. H. and Shin J. 2002, "A Public Key Audio Watermarking Using Patchwork Algorithm". *Proceedings of ITC-CSCC 2002*, pp.160-163.
12. Katzenbeisser S. and Petitcolas F. A. P. 2000, "Information Hiding Techniques for Steganography and Digital Watermarking". *Artech House*, UK.
13. Kim H. O., Lee B. K. and Lee N. -Y. 2002, "Wavelet-Based Audio Watermarking Techniques: Robustness and Fast Synchronization". In <http://amath.kaist.ac.kr/research/paper/01-11.pdf>.
14. Neubauer C. and Herre J. 1998, "Digital Watermarking and its Influence on Audio Quality". *105th AES Convention, Audio Engineering Society preprint 4823*, San Francisco.
15. Neubauer C. and Herre J. 2000a, "Audio Watermarking MPEG-2 AAC Bitstream", *108th AES Convention, Audio Engineering Society Preprint 5101*, Paris.
16. Neubauer C. and Herre J. 2000b, "Advanced Audio Watermarking and Applications". *109th AES Convention, Audio Engineering Society Preprint 5176*, Los Angeles.
17. Painter T. and Spanias A. 2000, "Perceptual Coding of Digital Audio". *Proc. of IEEE*, Vol. 88, No. 4, pp. 451-513.
18. Pan D. 1995, "A Tutorial on MPEG / Audio Compression". *IEEE Multimedia*, pp. 60-74.
19. Qiao L. and Nahrstedt K. 1999, "Non-invertible Watermarking Methods for MPEG Encoded Audio", *Conf. on Security and Watermarking of Multimedia Contents*, pp. 194-202.
20. SDMI, 2000, Call for proposal, <http://www.sdmi.org/cfp.html>.
21. Swanson M. D., Zhu B., Tewfik A. H. and L. Boney L. 1998, "Robust Audio Watermarking Using Perceptual Masking", *Elsevier Signal Processing, Sp. Issue on Copyrights Protection and Access Control*, Vol. 66, No. 3, pp. 337-355.
22. Voyatzis G. and Pitas I. 1998, "Chaotic Watermarks for Embedding in Spatial Digital

- Image Domain”. In *Proc. ICIP98*, Chicago, Vol. II, pp. 432-436.
23. Yeo I.-K. and Kim H. J. 2001, “ Modified Patchwork Algorithm: A Novel Audio Watermarking Scheme”. *Proc. of the International Conf. On Information Technology: Coding and Computing* , pp. 237 – 242.

AN AUDIO SEPARATION SYSTEM BASED ON THE NEURAL ICA METHOD

MICHAL BRÁT, MIROSLAV ŠNOREK

Czech Technical University in Prague

Faculty of Electrical Engineering

Department of Computer Science and Engineering

Karlovo náměstí 13, 121 35 Praha 2

Email: bratm@fel.cvut.cz, snorek@cslab.felk.cvut.cz

KEYWORDS

Data Mining, Signal Mining, Blind Signal Separation - BSS, Independent Component Analysis - ICA, Fast Fourier Transformation - FFT, Principal Component Analysis - PCA, Self-Organizing Map - SOM, Learning Vector Quantization – LVQ.

ABSTRACT

This contribution deals with the problems based on data mining, especially signal mining. The main representative of signal mining is Blind Signal Separation. This group of problems can be solved by traditional (mathematical) methods or also untraditional techniques that utilize artificial intelligence such as neural networks. They are not possible to use alone, therefore this contribution focuses on pre-processing of input signals too. In conclusion we show our developed system based on self-organizing neural network and several experiments with it.

1. INTRODUCTION

At this time the amount of data in electronic format in many academic and other disciplines is increased. Otherwise one from many problems of huge data is in their incomprehensiveness and blind information about them. The group of data problems comes under a discipline, which is called data mining. One part of data mining, that is concentrated only on the signal problems, is well known as data stream mining or also signal mining. We

may solve these problems by traditional methods such as mathematical algorithms, especially statistical algorithms or also other untraditional techniques based on artificial intelligence like neural networks.

2. PROBLEM DEFINITION: BLIND SIGNAL SEPARATION

Imagine a group of people who are sitting in a room and speaking simultaneously (see Figure 1). We are member of speaking group and we want to obtain speech from only a person who is speaking important information for us. We must quite concentrate on this person. Human ability of speech recognition can exactly focus on speech from one person and other noise is eliminated. We want to implement the same recognition abilities in a computer science.

This problem based on separation of a signal is well known as “cocktail party problem”. It is one problem of Blind Signal Separation (BSS). The separation is called **blind** because we do hardly know quite anything about an environment in which mixing of signals takes place. It is special section of signal mining, which focuses on signal separation with minimal information about input signals. They are just hard problems from data mining.

The BSS problem covers in as well other signal process. This is economic data stream mining, which wants to obtain knowledge about data stream. Other process is based on separation of damaged medical signals such as EEG or MEG. All these problems are almost solved by traditional techniques. The main representative of these techniques is Independent Component Analysis (ICA) [1]. It

could be used other techniques based on adaptive filters, decision rules and others.

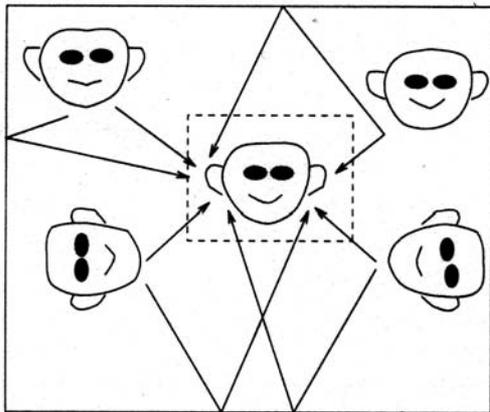


Figure 1: A typical situation in a cocktail party problem

3. STANDARD TECHNIQUES BASED ON MATHEMATICAL ALGORITHMS

The traditional methods for solving of problems, which come out the BSS problem, are almost based on complex mathematical algorithms. The main representative of these techniques is the ICA method. The basic idea of it comes out non-linear transformation of signals in co-ordinates system. The new one represents turning of co-ordinates to direction for better view of signals. Firstly, co-ordinates are turned to direction of maximal variance (this is second statistical moment, in fact it is only linear transformation). Then it is used non-linear transformation of signals. Co-ordinates are turned in direction of maximal kurtosis (it is third statistical moment). More details about it are in [3].

The ICA method is very useful but its computation by mathematical algorithms is quite complex. It can be implemented by easier techniques - using artificial intelligent especially neural networks. Neural networks can be usable for many applications and solutions of hard and non-algorithm problems. The basic idea of the neural ICA method comes out mathematical solution, but implementation is completely different.

4. IMPLEMENTATION OF THE ICA METHOD BASED ON NEURAL NETWORKS

First idea about neural solution of the ICA method has been inspired by article *Non-*

linear Blind Source Separation by Self-Organizing Maps [4]. This meaning was not quite perfect because author has entirely used SOM without using other methods for modification of input signals. Therefore we have prepared first version of a system, which is improvement of the idea came out promising article.

This system (ExNeurICA_PS) is based on neural networks with pre-processing of input signals. A structure of this system is shown in Figure 2. The basic idea of the system consists of pre-processing of input signals and a core of system using neural networks. Pre-processing of input signals is done by the PCA method. It is in fact the same pre-processing such as for mathematical solution of the ICA method. Co-ordinates are turned to direction of maximal variance.

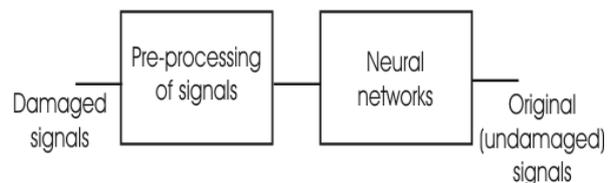


Figure 2: The structure of our systems

The second part of this system utilizes neural network, especially Kohonen's self-organizing map SOM. This neural network seems to be also used for non-linear transformation because of its architecture.

We have prepared several experiments with this system [3]. These results seemed to be not perfect therefore we have prepared new system. More details about previous system are in [2].

5. METHOD IMPROVEMENT: FREQUENCY DOMAIN APPROACH

The structure of new system is the same as previous system, but meaning is completely different. Audio signal in time domain is not quite applicable because it is dependent of quality and level of signal. Therefore almost all audio signals are processed in frequency domain because of easier elaboration. Generally, the signals in frequency domain keep better features.

The same idea about signals in frequency domain is usable for implementation of the ICA method. A developed system is just based on frequency pre-processing and clustering

according to self-organizing neural networks. Transformation from time to frequency domain has been performed by fast Fourier transformation (FFT).

Now we can define variables for computing of this system. The input signals are $x(t)$. They are in fact the damaged (or also mixed) signals, which are separated. The separated signals are marked as $s'(t)$. The original signals $s(t)$ mean etalon for test of quality results. In fact we have not these signals in real application. In addition to they are the basic variable and the inside (only in system) variable is Fourier's image $X(k)$.

5.1. Fast Fourier Transformation - FFT

This transformation has been known a long time but in era without computers it was disapproved and not much used. At this time this transformation is quite used, mainly in discipline, which deals with an audio process. The signal is transformed by the equation

$$X(k) = \sum_{i=0}^{N-1} x(i) e^{-\frac{j2\pi i k}{N}}, \quad k = 0, 1, 2, \dots, N-1$$

where $x(i)$ represents the mixed signal (in the time domain) and $X(k)$ is Fourier's image of the mixed signal (in the frequency domain). It is FFT, but we need also inversion of FFT (iFFT). It is defined by the equation

$$s'(i) = \sum_{k=0}^{N-1} S'(k) e^{\frac{j2\pi i k}{N}}, \quad i = 0, 1, 2, \dots, N-1$$

where $S'(k)$ represents Fourier's image of the estimated signal (in the frequency domain) and $s'(i)$ is the estimated signal (in the time domain).

5.2. Neural Networks SOM and LVQ

We use the same neural networks such as was used in first system. First used neural network SOM has been used as classifier [5], because of its non-linear ability of transformation. The basic idea of it is based on "change a position of neurons" (in fact it is only change the weight of neurons). These neurons are attracted to clustering. The basic idea of using SOM in develop system is shown in Figure 3. The spectral lines, which are very close among them, are clustered. Each cluster means an audio signal in frequency domain. After iFFT, these signals are separated to time

domain. SOM is unsupervised neural networks therefore we do not exactly set a number of clusters. This is very important, because a number of clusters must be the same as a number of signals. This condition cannot be followed.

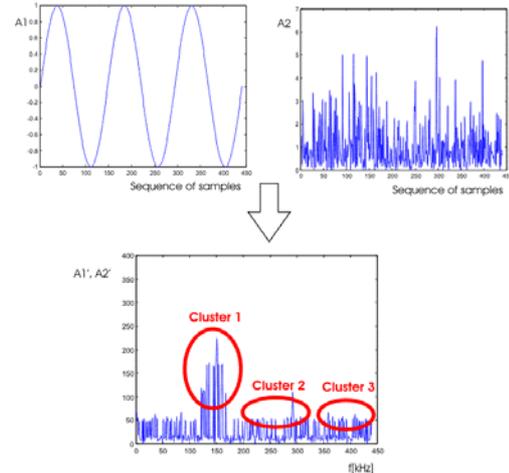


Figure 3: The basic idea based on SOM (This is not possible to set a number of clusters therefore there are different a number of clusters than signals.)

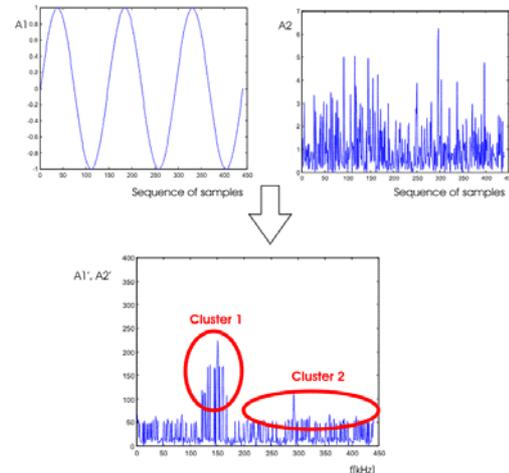


Figure 4: The basic idea based on LVQ (There is exactly to set a number of clusters. The number has to be the same as a number of signals.)

Accordingly we have used LVQ because of its similarity with SOM. This neural network is simply put "SOM with supervised learning" [5]. The idea of this system is the same as with SOM, but we can set exact number of clusters. The basic idea using LVQ is shown in Figure 4.

After both clustering (by SOM or LVQ) we transform signals in time domain from

frequency clusters. For example in Figure 4, the Cluster 1 is first separated signal and Cluster 2 is second separated signal. We describe only situation with two signals, but this idea is used for more signals. For easier explanation we show only this approach.

This system was programmed in Java programming language. It follows that it is independent of operation system. This system will be located on web page <http://cs.felk.cvut.cz/~bratm>.

6. EXPERIMENTS

We would like to describe several experiments with a developed system. Some experiments utilize simple audio signals (e.g. mixture of audio signals with exactly fixed frequency) and also songs or human speech. The experiments have been performed on a PC with the Intel 600 MHz processor, with 256 MB operation memory. The operation system has been Windows 2000. We have prepared more experiments, but now we show only experiments with audio signals.

The input (mixed) signals are shown in Figure 5 a) and b). There are two mixed signals, in fact damaged signals, which have to be repaired. This is simulation of cocktail party.

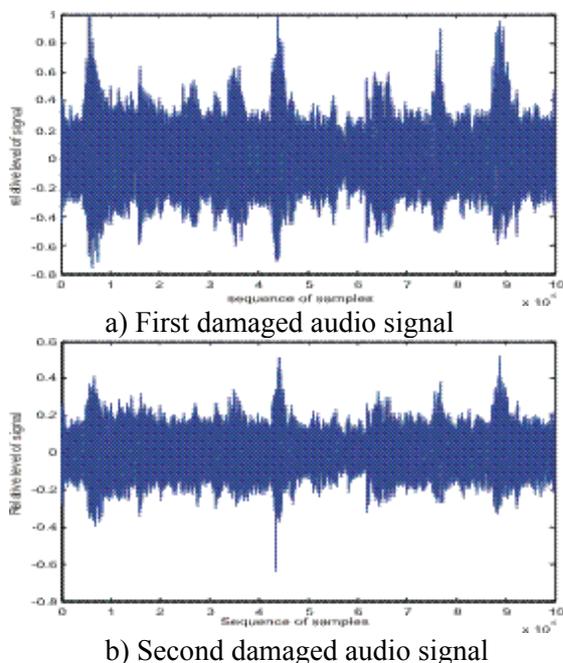


Figure 5: The input mixed signals (speech /one, two, .../ & song simultaneously)

The mixed signals have been pre-processing by FFT. After that we have only used SOM, because the results are quite good. The results can be shown in Figure 6 a) and b).

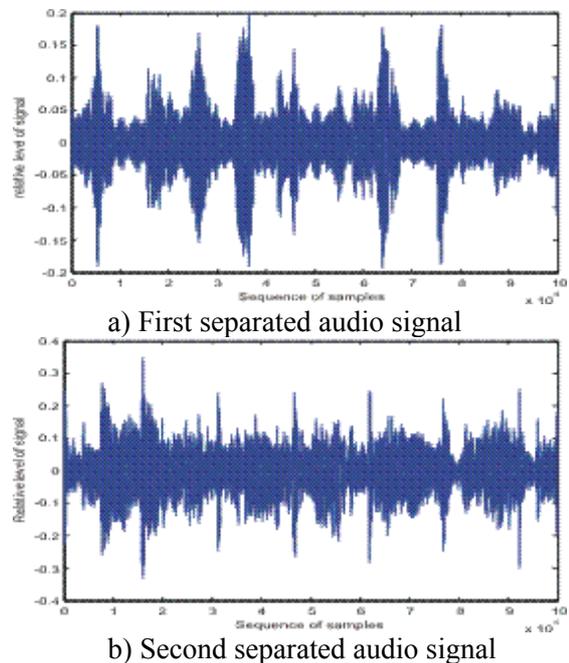


Figure 6: The separated signals

We would like to compare quality of separated signals. Quality can be obtained from joint density. This graph of joint density must be square, but it can be also turned. Figure 7 shows joint density of mixed signals. It is non-orthogonal (non-squared). Figure 8 shows joint density after separation. It is much better then mixed signals. If we look at audio signals or we are listening songs, it is quite good. In conclusion we resume our results.

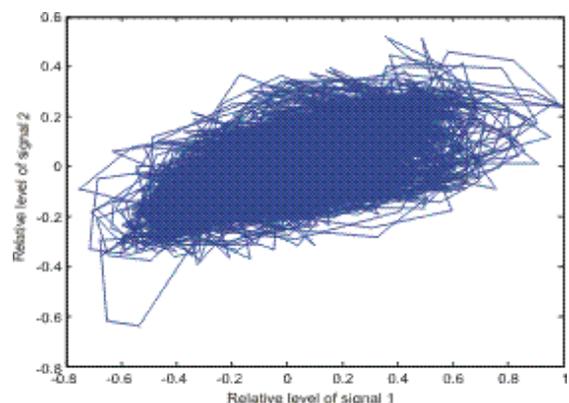


Figure 7: Joint density of mixed signals

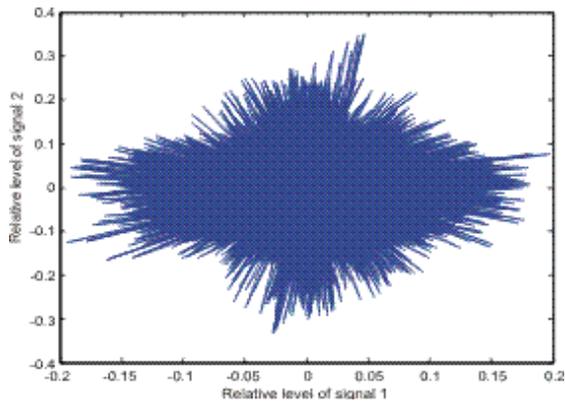


Figure 8: Joint density of separated signals

7. CONCLUSION

This system based on FFT and SOM seems to be very usable for audio separation problems. We can resume that this idea can be used for solving of the BSS problem especially cocktail party problem. We also know that this system is not completely perfect. Firstly we thought that the system based on clustering using SOM is inapplicable. But during developed experiments we ascertain that this system is quite good, maybe better than a system utilizes an idea of clustering by LVQ. We are increasing this system based on LVQ neural network and after that we compare the results. We would like to present our results on next conference.

REFERENCES

- [1] Hyvärinen, A., Karhunen, J., Oja, E. 2001. "Independent Component Analysis". Canada. ISBN 0-471-40540-X
- [2] Brát, M., Šnorek, M. 2002. "Extended Neural ICA for Blind Signal Separation". pages 125-132. MOSIS. ISBN 80-85988-71-2.
- [3] Brát, M. 2003. "Blind Signal Separation – Data Streams Mining Using Neural Network". Postgraduate Study Report DC-PSR-2002-11. CTU.
- [4] Pajunen, P., Hyvärinen, A., Karhunen, J. 2000. "Non-linear Blind Source Separation by Self-Organizing Maps". Helsinki University of Technology. Espoo.
- [5] Šíma, J., Neruda, R. 1996. "Teoretické otázky neuronových sítí". MATFYZPRESS. ISBN 80-85863-18-9.

AUTHORS BIBLIOGRAPHY

MICHAL BRÁT was born in south-bohemia in Počátky, Czech Republic, in 1977. He studied Computer Science and Engineering at Czech Technical University. At this time he is Ph.D. student at the Department of Computer Science and Engineering of Faculty of the same University (CTU).



He is interested on a processing of signals, audio and video process and artificial intelligent, especially neural networks.

MIROSLAV ŠNOREK was born in south bohemian town Písek, CZ, in 1947. He studied Technical Cybernetics at Czech technical University Prague and he graduated in 1970. He is currently



Associated Professor at the Department of Computer Science and Engineering of Electrical Faculty of the same university (CTU).

He is the head of Neural Network Group. His research interests include unsupervised clustering, GMDH algorithm and neural network applications in modelling and interfacing computers to the real world.

SIMULATION MODELING OF UML SOFTWARE ARCHITECTURES*

SIMONETTA BALSAMO MORENO MARZOLLA

*Dipartimento di Informatica
Università Ca' Foscari di Venezia
via Torino 155, 30172 Mestre (VE), Italy
e-mail: {balsamo/marzolla}@dsi.unive.it*

Abstract Quantitative analysis of software systems is being recognized as an important issue in the software development process. Performance analysis can help to address quantitative system analysis from the early stages of the software development life cycle, e.g., to compare design alternatives or to identify system bottlenecks. Modeling software systems by simulation allows the analyst to represent detailed characteristics of the system. We consider simulation for performance evaluation of software architectures specified by UML. We derive a simulation model for annotated UML software architectures. First we propose the annotation for some UML diagrams to describe performance parameters. Then we derive the simulation model by automatically extracting information about Use Case and Activity Diagrams from the XMI descriptions of UML diagrams. This information is used to build a discrete-event simulation model, which is finally executed. Simulation results are inserted back into the original UML diagrams as tagged values to provide feedback at the software architectural design level.

Keywords: Process-Oriented Simulation, Software Systems, Unified Modeling Language, Performance Evaluation.

1 INTRODUCTION

In recent years it has been recognized that the software development processes should be supported by a suitable mechanism for early assessment of software performance. Early identification of unsatisfactory performance of Software Architecture (SA) can greatly reduce the cost of design change [Smith, 1990; Smith and Williams, 2002]. The reason is that correcting a design flaw is more expensive the later the change is applied during the software development process. This is particularly true if the waterfall software development model is employed, as any change during the development process requires to start back from the beginning. However, this is still a relevant issue whenever a different development process is used.

Both quantitative and qualitative analysis can be performed at the software architectural design level. Qualitative analysis deals with functional properties of the software system such as for example deadlock-freedom or security. Qualitative analysis is carried out by measurement or by modeling the software system to derive quantitative figures of merit, such as, for example, the execution profile of the software, memory or network utilization. We focus on performance models of software systems at the SA level.

In this paper we consider quantitative evaluation of the performance of SA at the design level by means of simulation models. We consider SA expressed in terms of Unified Modeling Language (UML) [Object Man-

agement Group, 2001] diagrams. We propose to annotate the UML diagrams using a subset of annotations defined in the UML Profile for Schedulability, Performance and Time Specification [Object Management Group, 2002a] (referred as *UML performance profile*).

Simulation is a powerful modeling technique that allows general system models; simulation models can represent arbitrarily complex real-world situations, which can be too complex or even impossible to represent by analytical models. We define a simulation model of an UML software specification introducing an almost one-to-one correspondence between behaviors expressed in the UML model and entities or processes in the simulation model. This correspondence between system and the model helps the feedback process to report simulation results back into the original SA.

There are a few previous works dealing with simulation of UML specifications. Arief and Speirs [Arief and Speirs, 1999a,b, 2000] developed an automatic tool for deriving simulation models from UML Class and Sequence diagrams. Their approach consists in transforming the UML diagrams into a simulation model described as an XML document. This model can then be translated into different kinds of simulation programs, even written in different languages. In this way the performance model is decoupled from its actual implementation. De Miguel et al. [De Miguel et al., 2000] introduced UML extensions for the representation and automatic evaluation of temporal requirements and resource usage, particularly targeted at real-time systems. The extensions are expressed in term of stereotypes, tagged values and stereotyped constraints. These were intro-

*This work has been partially supported by MURST Research Project "Sahara" and by MIUR Research Project "Performance Evaluation of Complex Systems: Techniques, Methodologies and Tools".

duced in a commercial UML CASE tool, which has been made able to generate OPNET simulation models starting from annotated UML diagrams.

Previous simulation-based performance modeling approaches for evaluation of UML SA were developed before the UML performance profile was defined. Thus, they introduced their special extensions of UML or introduced non standard annotations to express quantitative information useful for deriving the model. In this paper we describe UML Performance Simulator (UML-PSI), a performance evaluation tool which translates UML Use Case and Activity diagrams into a discrete-event process oriented simulation model. The UML diagrams are annotated according to a subset of the UML Performance Profile. This additional information is used to define the simulation model, which is finally executed. Simulation results are inserted back into the original UML diagrams as tagged values to provide feedback at the software architectural design level.

This paper is organized as follows. In Section 2 we illustrate the proposed methodology for generating simulation models from UML SA. In Section 3 we describe UML-PSI, a tool we built to implement that methodology. In Section 4 we illustrate a simple case study, and conclusions and future works are discussed in Section 5.

2 METHODOLOGY

In order to assist the software developer during the design process, we illustrate in Fig. 1 a general framework for quantitative analysis of UML SA [Balsamo et al., 2002]. The starting point is a description of the SA. We consider a description as a set of UML diagrams annotated with quantitative information in order to derive a simulation-based performance model. The model is obtained using a suitable Modeling Algorithm. The model is then implemented in a simulation program, which is eventually executed. Simulation results are a set of performance measures that can be used to provide a feedback at the original SA design level. The feedback should pinpoint performance problems on the SA, and possibly provide suggestions to the software designer about how the problem can be solved. The modeling cycle can be iterated until a SA with satisfactory performances is developed.

Note that the modeling and performance evaluation framework of Fig. 1 is independent from the particular performance model that we apply. In this paper we consider simulation-based performance models of UML SA. We describe UML-PSI, a prototype performance evaluation tool which processes an XMI [Object Management Group, 2002b] description of UML Use Case and Activity diagrams. The UML SA has to be annotated using a simplified subset of the UML Profile for Schedulability, Performance and Time Specification [Object Management Group, 2002a]. The simulation model is process oriented and its objects are

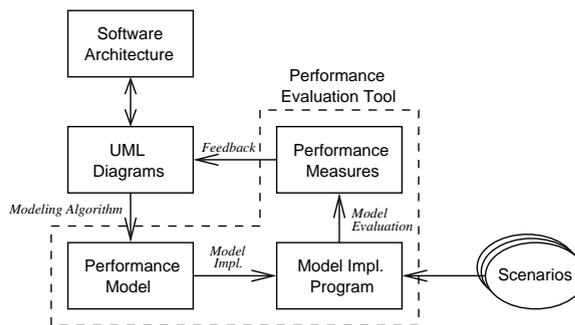


Figure 1: Framework for quantitative evaluation of UML diagrams

derived by the analysis of the UML diagrams annotated with performance specification of the software system components. The simulation model is implemented as a discrete-event simulation program written in C++, whose execution provides results for a set of performance indices. We evaluate through simulation the mean response time associated with the execution of each scenario (Use Case) and each scenario step (Activity). Simulation results, i.e., the performance measures of the software components are inserted back into the original UML SA as tagged values to provide feedback to the system designer.

Figure 2 illustrates the structure of the performance simulation model derived from the UML diagrams. The basic object of the simulation model is a `PerformanceContext`. This object contains the other elements of the model, namely `Workloads` and `Scenarios`. `Workloads` can be open or closed, depending on whether the number of users accessing the system is unbounded or fixed. Open workloads are characterized by the following attribute (the exact notation used to describe the attributes is given in the next section):

occurrencePattern (of type `RTarrivalPattern`, defined in Section 3) the pattern of interarrival times of consecutive requests

Closed workloads are characterized by two attributes:

population the total number of users in the workload

externalDelay (of type `PAPerfValue`, defined in Section 3) the delay between the end of a scenario execution and the beginning of the next request

Each workload actually drives one or more scenarios. Each time a new workload user requests service to the system, one of the scenarios associated with that workload is selected. Selection is done randomly, according to the probability associated to each scenario.

A scenario is a set of abstract scenario steps, represented by the `AbsStep` class. All kinds of scenario steps are characterized by the following attributes:

probability the probability to execute this step, in the case the predecessor step has multiple successors

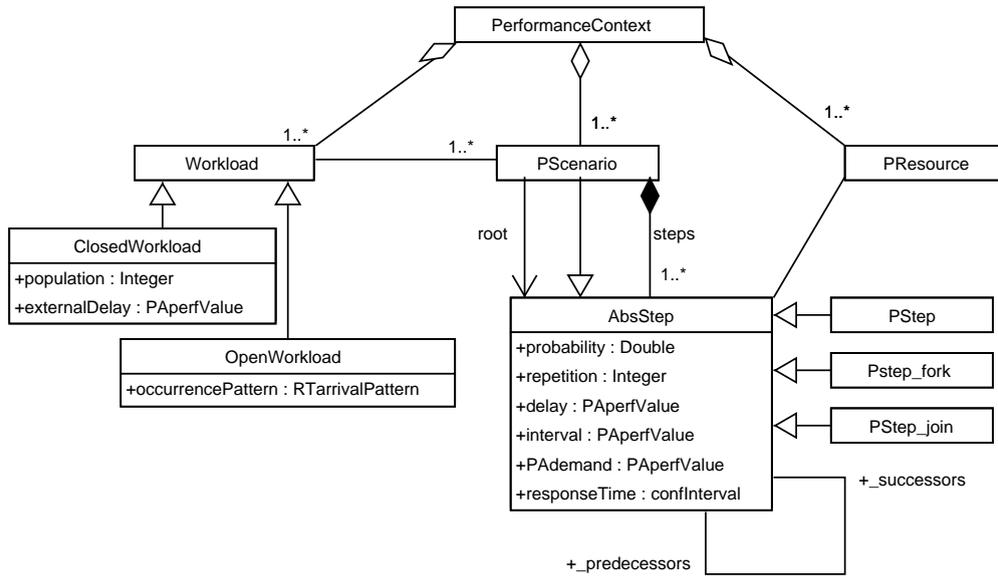


Figure 2: Structure of the simulation performance model

repetition the number of times this step has to be repeated

delay (of type PAPERfValue) an additional delay in the execution of this step, for example to model a user interaction

interval (of type PAPERfValue) the time between repetitions of this step, if it has to be repeated multiple times

PAdemand (of type PAPERfValue) the processing demand of this step

responseTime the computed delay between the starting and finishing time of this step. The estimation of this quantity is the result of the simulation model execution. This value has type confInterval, which we define as the pair of the confidence interval bounds.

Abstract scenario steps can either be composite steps (described by the PScenario class), or atomic steps of different kinds. Scenarios are collections of steps; exactly one of these steps is marked as the root step (starting step) of the scenario. Atomic steps can be of type PStep_fork for nodes representing the creation of multiple execution threads, PStep_join for nodes representing synchronization points between different threads, and PStep for normal atomic steps.

There are two main differences between the performance model depicted in Figure 2 and the one described in the UML Performance Profile. First, we assume a very simple model for resources: each Use Case has an associated computational resource; the resource is acquired when the Use Case starts, and is released at its completion. Thus, two instances of the same Use Case cannot be executed in parallel. This simplification

is motivated by the fact that we evaluate SA at the architectural level, without assuming any implementation on a specific platform. Indeed we assume that the software architect ignores the specific hardware platform on which the software will be executed. Detailed resource modeling is usually done in later stages of the software development process.

Second, the structure we propose for the class hierarchy describing the processing steps is slightly different from that in [Object Management Group, 2002a]. The UML Profile defines a PScenario class from which a PStep class is derived, thus every step is a scenario. This is because each step, at a deeper level of detail, could be modeled as a whole scenario, that is, a sequence of multiple sub-steps. We choose a different structure to model the PStep and PScenario hierarchy to keep atomic steps and scenarios as separate entities. We apply the Composite Pattern [Gamma et al., 1995] to reflect the hierarchical nature of the processing steps. This choice makes the construction of the simulation model easier, because there are different kinds of step (e.g., PStep, PStep_fork, PStep_join) which are modeled as different simulation object types. The behavior of a fork step consists of activating all the successor steps concurrently. A join step waits for the completion of all predecessor steps before activating the successor. A normal step simulates execution according to the specified delays, and activates one of the successor steps. Finally, the behavior of a scenario is to activate its root step.

To summarize, the proposed approach to derive the simulation model to evaluate SA performance from UML Use Case and Activity diagrams is defined as follows:

1. Consider an UML representation of a software

system in terms of Use Case and Activity diagrams. Both diagrams are respectively annotated as follows:

- UML Use Case diagrams describe the interaction between the software system and one or more Actors requiring service. As proposed in [Cortellessa and Mirandola, 2002; Pooley and King, 1999] we identify Actors to represent workloads applied to the system and Use Cases to represent scenarios. Actors can be stereotyped as \ll PAopenLoad \gg or \ll PAClosedLoad \gg to represent respectively open and closed population of users accessing the system. Use Cases are tagged with PAprob tags, whose value indicates the probability of executing that scenario.
 - Each Activity of an Activity diagram can be tagged with the following informations: the number of times the step has to be repeated (PArep); the delay between repetitions (PAinterval) of the same step; an additional delay for each step representing user “think time” (PADelay); the service demand of the step (PADemand).
2. The simulation model is automatically derived from the XMI description of the UML diagrams. Currently UML-PSI uses the XMI dialect of the open-source ArgoUML CASE tool [ArgoUML, 2003], The simulation model is an instance of the general class structure depicted in Figure 2, and includes a PerformanceContext object and a set of workloads and scenarios.
 3. The simulation model is executed, optionally asking the user to specify some parameters for the simulation, such as the desired confidence level for the estimation of the performance indices, the confidence interval width and the simulation length.
 4. Simulation results are inserted back into the UML model as tagged values associated with Activities and Use Cases. We consider as simulation results the average delays (PArespTime) of Activities and Use Cases execution.

3 UML-PSI TOOL DESCRIPTION

The steps of the proposed methodology are implemented into a simulation tool called UML-PSI. As concerns the first step of the annotation of UML diagrams, the UML Performance Profile suggests the use of TVL (Tag Value Language) to describe values of tags applied to model elements, which is a subset of the Perl language [Wall et al., 2000]. We use a freely available Perl interpreter library [CPAN, 2003] to evaluate tag values, so taking advantage of the full Perl language. Note that

both the UML Performance Profile and UML-PSI do not strictly depend on the specific language used to express annotations.

The PAperfValue and RTarrivalPattern data types are expressed according to the following BNF notation, which is a simplified version of the annotations defined in the UML Performance Profile:

```

< PAperfValue > := '[' assm|pred|msrd,
                    dist, < PDFstring > ']'
< PDFstring > := '[' < constantPDF > |
                    < uniformPDF > |
                    < exponentialPDF > |
                    < normalPDF > ]
< constantPDF > := < real >
< uniformPDF > := uniform, < real >, < real >
< exponentialPDF > := exponential, < real >, < real >
< normalPDF > := normal, < real >, < real >
< RTarrivalPattern > := '[' < bounded > |
                        < unbounded > |
                        < bursty > ']'
< bounded > := bounded, < int >, < int >
< bursty > := bursty, < PDFstring >, < int >
< unbounded > := unbounded, < PDFstring >

```

UML-PSI parses an XMI description of UML diagrams, annotated as described above. The UML model is translated into a C++ discrete-event simulation program according to step 2 of the proposed methodology. UML-PSI includes a general purpose discrete-event simulation library providing roughly the same functionality of the Simulation class of the SIMULA language [Dahl and Nygaard, 1966], namely pseudo-parallel process execution using coroutines and Sequencing Set scheduling facilities. We developed the simulation library for several reasons, mainly code portability, availability of compilers and the necessity to use a freely available C library for parsing XML documents [libxml, 2003].

The simulation library includes random number generators and some statistical functions. Random variates of various distributions are generated using the uniform random number generator described in [L'Ecuyer, 1999].

The library provides basic estimation functions, e.g. mean, variance and confidence interval. Since we consider steady-state simulation, we discard an appropriate initial portion of the observations to remove the initialization bias. The mean is computed using the method of independent replications [Banks, 1998]. The simulation stops when the relative width of the computed confidence intervals is smaller than a given threshold. Both the confidence level and the threshold can be defined by the user. If they are not provided, we assume default values of 90% confidence level and 5% threshold.

Each UML Actor is mapped into an appropriate `Workload` object, that can be an open or closed workload, depending on the stereotype which is applied to the UML element. Workloads are active objects in the simulation, which simply perform an endless loop in which they select and activate a Use Case, whose behavior is activating the root step of its associated Activity diagram. Each Activity in the Activity diagram is translated into the corresponding kind of `AbsStep` object. These objects simulate execution of the corresponding step or scenario according to the structure of the diagram and the value of the associated tags. At the end of the simulation, the computed average response time of each `AbsStep` object (that is, a step or scenario) is inserted into the original UML diagram the `PArespTime` tag.

We propose to apply the UML-PSI tool as described in the modeling cycle illustrated in Figure 1. Namely, the software designer defines an UML SA with ArgoUML and specifies model parameters as tagged values associated with UML elements. Then, the user optionally selects parameters such as simulation length or confidence interval width, and runs the simulation. When the simulation finishes, the results are automatically inserted into the UML diagrams. At this point the software designer accesses the simulation performance results by opening the ArgoUML project file to access the results, and then possibly iterates the performance evaluation analysis by providing a new set of parameters in the UML model tags.

4 CASE STUDY

In this section we illustrate with a simple example how UML-PSI works. The example involves an e-commerce application in which users can browse a web catalog of products or submit purchase orders. We assume an unlimited stream of users requiring service. Users inter-arrival time is exponentially distributed with mean A . Users can browse the online catalog with probability p , and make an order with probability $1 - p$. Browsing the catalog involves two sequential activities, which are: issuing a request to the product database and composing the web page. Making an order involves the following activities: selecting a product, filling the order form, processing the order and verifying the payment informations. Orders must be paid by credit card, whose number must be validated. Order validation and credit card checking are performed in parallel.

The UML Use Case and Activity diagrams are depicted in Figure 3. Model elements are tagged according to the notation described in Section 2.

Note that an analytical model of the system of Figure 3 can not be easily evaluated due to the fork/join component of the Make Order Activity. The simulation model which is derived from the SA is made of several active components (processes) arranged according to the structure of Figure 2. The actor is represented by

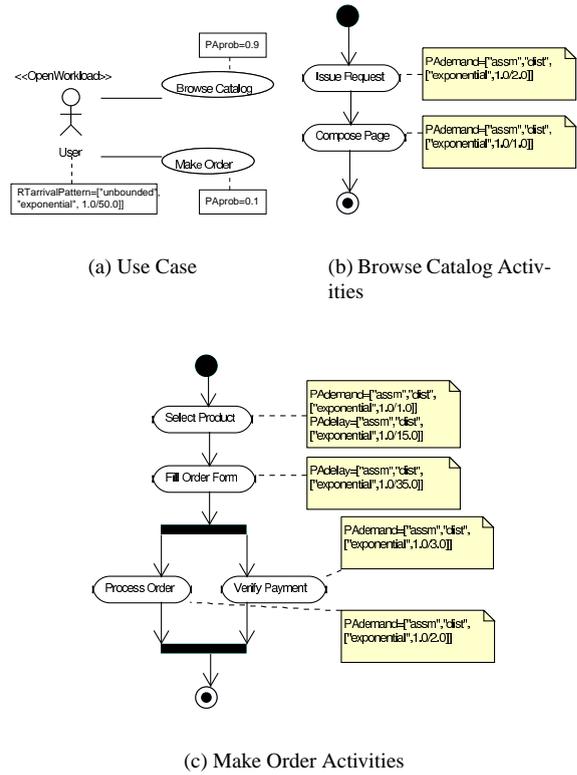


Figure 3: UML representation of an E-commerce application.

an object of type `OpenWorkload`, which generates a stream of users. Use Cases are objects of type `PScenario` and have a queue which collects users according to their arrival order. Users select the Use Case to join according to the associated probability. Use Cases simulate one user request at a time activating the root step of the corresponding `PScenario`. When the scenario execution completes, a new user requests in the Use Case queue is started. Finally, there is one simulation process for each node in the Activity diagram. Nodes can be Activity nodes, which are modeled by processes of type `PStep`, or fork and join nodes which are modeled by processes of type `PStep_fork` and `PStep_join`, respectively.

Scenario	Average delay (seconds)
Make Order	[2.98394, 3.00172]
Browse Catalog	[54.7787, 55.7254]

Table 1: Simulation results for the e-commerce application of Figure 3.

A numerical example of the performance results, computed as steady-state average delays are reported in Table 1. The results are obtained by setting $A = 50sec$ and $p = 0.9$; other model parameters are set as shown in Figure 3. The computed intervals are at 90% confidence level. The results are inserted into the original UML diagrams as values of the `PArespTime` tag. These

tagged values are attached to each relevant model element. The software designer can now explore different situations by repeating the modeling and performance evaluation process with different tag values.

5 CONCLUSIONS

We have proposed a simulation-based performance modeling approach for UML software architectures. The notation we have used to describe performance parameters is a subset of the one defined in the UML Performance Profile. We have derived a simulation model from Use Case and Activity diagrams. The simulation model is executed and the results are inserted into the original UML diagrams as tagged values. We have presented UML-PSI, a prototype tool to implement the methodology.

The proposed approach has been defined to evaluate software performances at the SA design level. We plan to extend this approach to further steps of the software development process, by considering Deployment diagrams and resource allocation. Further research will be devoted to a more complete set of performance measures. UML-PSI will be extended accordingly.

REFERENCES

- ArgoUML (2003). ArgoUML – Object-oriented design tool with cognitive support. <http://www.argouml.org/>.
- Arief, L. B. and N. A. Speirs (1999a, June). Automatic generation of distributed system simulations from UML. In *Proceedings of ESM '99, 13th European Simulation Multiconference*, Warsaw, Poland, pp. 85–91.
- Arief, L. B. and N. A. Speirs (1999b, November). Using SimML to bridge the transformation from UML to simulation. In *Proc. of One Day Workshop on Software Performance and Prediction extracted from Design*, Heriot-Watt University, Edinburgh, Scotland.
- Arief, L. B. and N. A. Speirs (2000, September). A UML tool for an automatic generation of simulation programs. See [[Proceedings of WOSP 2000, 2000](#)], pp. 71–76.
- Balsamo, S., A. D. Marco, P. Inverardi, and M. Simone (2002, December). Software performance: state of the art and perspectives. Technical Report MIUR SAHARA Project TR SAH/04.
- Banks, J. (Ed.) (1998). *Handbook of Simulation*. Wiley–Interscience.
- Cortellessa, V. and R. Mirandola (2002, July). PRIMA–UML: a performance validation incremental methodology on early UML diagrams. In *Proceedings of the Third International Workshop on Software and Performance (WOSP 2002)*, Rome, Italy. ACM Press.
- CPAN (2003). Comprehensive Perl Archive Network (CPAN). <http://www.cpan.org/>.
- Dahl, O.-J. and K. Nygaard (1966, September). SIMULA—an ALGOL-based simulation language. *Comm. of the ACM* 9(9), 671–678.
- De Miguel, M., T. Lambolais, M. Hannouz, S. Betgé-Brezetz, and S. Piekarec (2000, September). UML extensions for the specifications and evaluation of latency constraints in architectural models. See [[Proceedings of WOSP 2000, 2000](#)], pp. 83–88.
- Gamma, E., R. Helm, R. Johnson, and J. Vlissides (1995). *Design Patterns: Elements of reusable Object-Oriented programming*. Addison–Wesley.
- L’Ecuyer, P. (1999). Good parameters and implementations for combined multiple recursive random number generators. *Operations Research* 47, 159–164.
- libxml (2003). libxml: the XML C library for Gnome. <http://xmlsoft.org/>.
- Object Management Group (2001, September). Unified modeling language (UML), version 1.4.
- Object Management Group (2002a, March). UML profile for schedulability, performance and time specification. Final Adopted Specification ptc/02-03-02, OMG.
- Object Management Group (2002b, January). XML Metadata Interchange (XMI) specification, version 1.2.
- Pooley, R. J. and P. J. B. King (1999, February). The Unified Modeling Language and performance engineering. In *IEE Proceedings – Software*, Volume 146, pp. 2–10.
- Proceedings of WOSP 2000 (2000, September). *ACM Proceedings of WOSP 2000, 2nd International Workshop on Software and Performance*, Ottawa, Canada.
- Smith, C. U. (1990). *Performance Engineering of Software Systems*. Addison–Wesley.
- Smith, C. U. and L. Williams (2002). *Performance Solutions: A Practical Guide to Creating Responsive, Scalable Software*. Addison–Wesley.
- Wall, L., T. Christiansen, and J. Orwan (2000, July). *Programming Perl* (third ed.). O’Reilly & Associates.

Metamodels for real-time control - an automotive design study.

Paul Stewart (corresponding author), Peter J. Fleming

Abstract— This paper examines the use of *metamodels* in the context of rapid prototyping for real-time control systems. It is desired that a drive by wire throttle controller be designed to minimise the acceleration oscillations which are a result of the first torsional mode of automotive drivelines. A response surface metamodel is fitted to the output of a complex experimentally verified model of the vehicle and driveline. Subsequently, the metamodel is used as a rapid prototyping tool, and as a basis for a final closed-loop design by evolutionary methods.

Keywords— Response surface methodology, metamodeling, evolutionary optimisation, rapid prototyping.

I. INTRODUCTION

Implementation of drive by wire strategies for the replacement of the conventional cable link between the throttle pedal and the throttle body has been the focus of development for many major automotive manufacturers. By fitting a stepper or permanent magnet servo motor [12] to the throttle body, and a throttle pedal with proportional voltage to position output, a “drive-by-wire” system can be implemented with a simple linear amplifier. If a micro-processor is added to the system, then control algorithms can be added to the operation of the throttle [13]. Controllers have been designed [11], which allow fast and accurate tracking of pedal demand, and have been shown to possess robust operating characteristics. An engine torque controller is designed and implemented in this paper to shape the vehicle response to the first torsional mode of the driveline. The initial requirement is to damp the acceleration oscillations generated by throttle step-demands. This dynamic mapping is constrained by the requirement to maintain where possible the vehicle acceleration response available to the throttle. Control analysis and design for this automotive system is complicated by a number of factors. There are a number of nonlinearities present, such as backlash in the gearbox, a tyre force which varies nonlinearly with road speed, and nonlinear clutch elements. Also, there is a process lag between throttle actuation and torque production and a nonlinear engine torque speed mapping [5]. Experimental data is available from a test car which was fitted with a data acquisition system including three axis accelerometers. A vehicle was loaned for the purpose of analysis, design and testing. This facilitated the development and verification of an accurate Matlab/Simulink dynamic model of the vehicle, driveline and engine. Al-

though accelerometers were available for experimental verification, in the first instance it was desired that only signals and measurements available on a standard unmodified car be used in the control system. A systematic excitation of the driveline was made experimentally on the vehicle model by performing step demands in all gears at discrete points throughout the effective engine speed range of the vehicle. The generated data (road speed, acceleration, engine speed etc.) was then modelled using the “Response Surface Methodology” (RSM) [7]. The method allows the exploration and optimisation of response surfaces, where the response variable of interest (vehicle acceleration) is related to a set of predictor variables (road speed, selected gear). In the development of a model and control system constrained by computational considerations, and the requirement of rapid prototyping, the RSM allows a low order approximation to be derived [14] by the method of least squares. The reduced order representation can then be employed in the controller design. Application of the RSM analysis to the vehicle response data allows a system model to be developed which lends itself to the design of a scheduled controller structure which is shown to control the first torsional mode of the driveline.

II. METAMODELLING OF THE SIMULINK MODEL

In the analysis of the acceleration response of the vehicle, there are three variables of interest, namely vehicle loading, road speed and selected gear ratio. Vehicle loading was assumed to be a fixed standard two person loading of $160kg$. In order to reduce the overall number of data sets required to construct the response surface of the system, a factorial approach to designed experiments was adopted [4]. The combinations of factorial experiments at $5ms^{-1}$ increments in each gear requires 25 experiments to be performed. Each proposed factorial combination was assigned a serial number and performed in random order via output from a random number generator. Each experiment consists of coasting the vehicle model in the appropriate gear at the appropriate roadspeed, and performing a 100% tip-in with the accelerator pedal. The simulated asphalt test road was assumed to be dry, with overcast sky and ambient temperature of $60^{\circ}F$. The effect of the energy storage components in the driveline can be clearly seen at $10ms^{-1}$ in second gear in Figure 1. Examination of the vehicle response to tip-in reveals a system which can be approximated as a delay and second order dynamic response with an overshoot and settling time which varies with road speed and selected gear. This approximation to describe the entire vehicle response can be formulated by application of

Paul Stewart is with the Electrical Machines and Drives Group in the Department of Electrical Engineering, Mappin Building, Mappin Street, Sheffield S1 3JD, United Kingdom. Tel: +44 (0)114 2225841, e-mail: p.stewart@sheffield.ac.uk. Peter J. Fleming is with the Department of Automatic Control and System Engineering, University of Sheffield, Sheffield U.K.

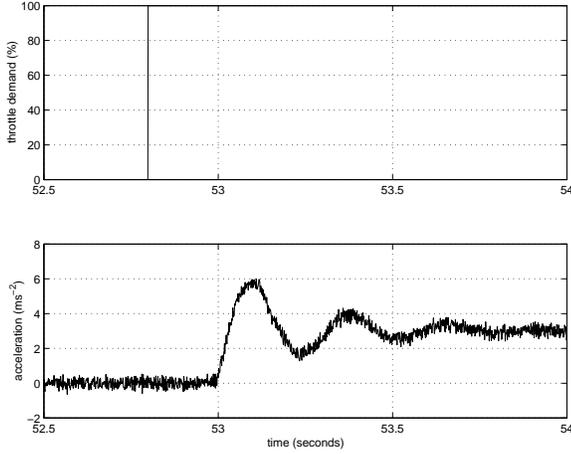


Fig. 1. Vehicle filtered experimental acceleration step response in second gear at 10ms^{-1} .

RSM to the experimental data. The data which has been gathered can be synthesised into a response map for the vehicle in order to design an oscillation control system. For the definition of driveability under consideration here, the vehicle response can be characterised as the damping ratio of the second order approximation map with variables road speed and selected gear. The individual responses may be expressed in terms of overshoot and settling time. The transfer function describing the open loop system may be described as

$$\frac{C(s)}{R(s)} = k \frac{1}{as^2 + bs + c} \quad (1)$$

from which the damping ratio of the system can be calculated as the ratio of the actual damping b to the critical damping $b_c = 2\sqrt{ac}$ [8]. Thus the damping ratio ζ can be calculated from $\zeta = \frac{b}{b_c}$. The roots of the characteristic equation 1 are $s_1, s_2 = -b_c \pm jc\sqrt{1 - b_c^2}$. This forms a complex conjugate pair from which the damping ratio and natural frequency can be computed. The natural units ξ_1 and ξ_2 of the experimental data (road speed and selected gear) are first transformed into the corresponding normalised coded variables x_1 and x_2 , such that

$$x_{i1} = \frac{\xi_{i1} - [\max(\xi_{i1}) + \min(\xi_{i1})]/2}{[\max(\xi_{i1}) - \min(\xi_{i1})]/2} \quad (2)$$

and

$$x_{i2} = \frac{\xi_{i2} - [\max(\xi_{i2}) + \min(\xi_{i2})]/2}{[\max(\xi_{i2}) - \min(\xi_{i2})]/2} \quad (3)$$

The model can be expressed in matrix form as [7]

$$\mathbf{y} = \mathbf{X}\beta + \epsilon \quad (4)$$

where

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1k} \\ 1 & x_{21} & x_{22} & \dots & x_{2k} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} & \dots & x_{nk} \end{bmatrix},$$

$$\beta = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix}, \quad \epsilon = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}. \quad (5)$$

It is now necessary to find a vector of least squares estimators \mathbf{b} which minimises the expression

$$L = \sum_{i=1}^n \epsilon_i^2 = \epsilon' \epsilon = (\mathbf{y} - \mathbf{X}\beta)' (\mathbf{y} - \mathbf{X}\beta) \quad (6)$$

and yields the least squares estimator of β which is

$$\mathbf{b} = (\mathbf{X}'\mathbf{X})^{-1} \mathbf{X}'\mathbf{y} \quad (7)$$

and finally, the fitted regression model is

$$\hat{\mathbf{y}} = \mathbf{X}\mathbf{b}, \quad \mathbf{e} = \mathbf{y} - \hat{\mathbf{y}} \quad (8)$$

where \mathbf{e} is the vector of residual errors of the model.

The second order model to be fitted to the data is

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{12} x_1 x_2 + \epsilon \quad (9)$$

Utilising equations (5-8), we obtain the coefficient matrix

$$\mathbf{b} = \begin{bmatrix} 0.4079 \\ -0.804 \\ 0.3809 \\ 0.0519 \\ -0.0429 \\ -0.0121 \end{bmatrix} \quad (10)$$

therefore the response surface in terms of the coded variables is obtained.

$$\hat{y} = 0.4079 - 0.804x_1 + 0.3809x_2 + 0.0519x_1^2 - 0.0429x_2^2 - 0.0121x_1x_2 \quad (11)$$

Comparing the computed response surface against a second experimental testdata set gave an average residual error of 1.65%. The design of the prototype driveability compensator will be considered in the next section.

III. RAPID PROTOTYPE DESIGN

A response surface has been obtained which describes accurately the vehicle's damping ratio map. As an initial design target, a damping ratio of 0.7 across the entire operating map would be a desirable response. The RSM analysis allows a simple open loop feedforward controller to be immediately designed and implemented to allow a fast appraisal of the actuator potential. The system response surface extracted from the experimental data is a representation of the complex conjugate pole pairs of the approximation (Figure 2) in terms of the system's varying damping ratio. The approach will be to effect a pole-zero cancellation of these complex conjugate poles to give

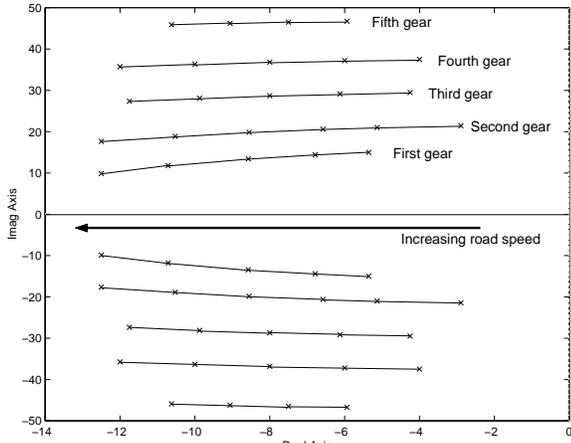


Fig. 2. Vehicle complex conjugate pole map.

a satisfactory response. This method does rely on accurate knowledge of the position of the uncompensated poles which has been ascertained experimentally for the purpose of this development. The parameters of the feedforward controller are derived from the response surfaces and are a function of selected gear and road speed. The feedforward compensator takes the form $\frac{as^2+bs+c}{as^2+ds+c}$ where the coefficient b is calculated from the damping ratio response surface, and performs pole-zero cancellation. Coefficient d produces the desired pole placement and forms the required damping ratio.

The control scheme was implemented on the microcontroller in assembly language, to ensure the fastest execution time. The demand from the throttle pedal and road speed were read in via A/D ports, and the selected gear read in via the digital I/O. Output from the controller was sent to a power amplifier via the PWM port. With the controller in place, the experimental set was on the vehicle to confirm the designed performance

IV. EXPERIMENTAL RESULTS - RAPID PROTOTYPING

The goal of rapid prototyping was achieved in a matter of days between initial model approximation and final implementation testing. The original set of experiments were repeated on the vehicle with the electronic throttle both compensated and uncompensated. A comparison at the $10ms^{-1}$ in second gear step response is shown in Figure 3. The time axis in both traces was zeroed at the initiation of the step demand for the purpose of clarity. The marked improvement in vehicle oscillation obvious in figure ?? was repeated throughout the operating map of the vehicle. The smooth acceleration increase is in marked contrast to the results achieved with (for example) polynomial pole placement techniques [10] in which oscillation is present in the rising acceleration trajectory. The compensated vehicle responses over the operating region of the vehicle were found to have a mean damping ratio of 0.68, with a maximum residual of 0.07. The tip-in driveability of the vehicle was found to be subjectively very improved, in addition to the experimental evidence of the vehicle compensated step re-

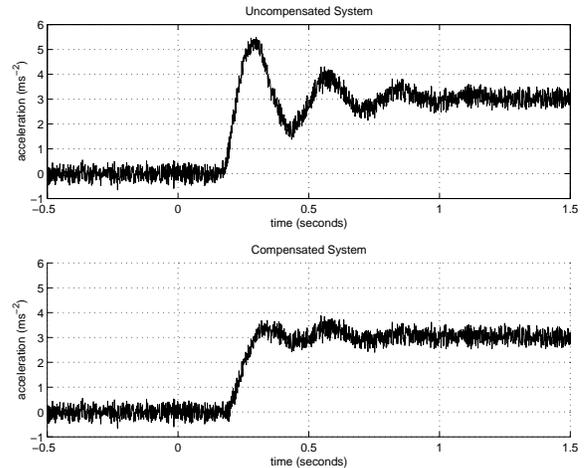


Fig. 3. Vehicle compensated and uncompensated filtered experimental step response.

sponse. Although a small amount of acceleration rate is sacrificed to achieve the suppression of oscillations subjectively the vehicles performance was not felt to be affected. The controller did endow the vehicle with a “turbine like” (a smooth positive rise in acceleration without any associated oscillations) feel.

V. METAMODELLING FOR CLOSED LOOP POLE-PLACEMENT DESIGN

In order to design a closed loop controller to achieve robustness against system parameter variations, a pole-placement approach was adopted. The objective of the pole-placement design method is to design a closed loop system with specified poles and thus the required dynamic response. The measured variable for feedback considered here is provided by a longitudinal accelerometer, should the derivation of acceleration from the velocity signal prove inaccurate or too noisy. The resulting characteristic equation will determine the features of the system, such as rise time, overshoot and settling time. The system model and its linear controller can be expressed respectively as

$$A(s)y(s) = B(s)u(s) \quad (12)$$

$$S(s)u(s) = T(s)u_c(s) - R(s)y(s) \quad (13)$$

where $A(s)$ and $B(s)$ are polynomials in the Laplace domain and $u(s)$ is the control variable. $S(s)$, $R(s)$ and $T(s)$ are the error, feedback and feedforward controller polynomials in the complex domain. The controller has input $u_c(s)$, which is the command signal and $y(s)$, is the measured output of the plant. Three constraints are associated with the model: the degree of $B(s)$ is less than the degree of $A(s)$, there are no common factors between polynomials $A(s)$ and $B(s)$, and $A(s)$ is a monic polynomial. From equations 12 and 13, the characteristic equation of the closed loop system will be

$$F(s) = A(s)S(s) + B(s)R(s) \quad (14)$$

The objective of the pole placement design is to find polynomials $S(s)$ and $R(s)$ that satisfy equation 14 for specified $A(s)$, $B(s)$ and $F(s)$. Equation 14 is known as the *Diophantine equation* and can be solved if the polynomials do not have common factors and the system is proper. The Diophantine equation can be solved using a linear matrix. Two major questions arise with the use of the pole placement design, firstly what is the optimum location of the poles for the characteristic equation of the controller, and secondly how many poles must be placed? Resolving the issue may be a matter of trial and error if the system is complex and there is noise in the feedback signals or (as it is the case of the driveline) there are nonlinearities, such as delays and saturation curves. In this case, a multiobjective genetic algorithm was adopted to find the optimal pole placements. The design problem in this case is described by a five component objective function:

- minimise rise time
- minimise overshoot
- minimise settling time
- minimise steady-state error
- minimise delay

The experimentally elicited metamodel for the vehicle, in terms of damping factor for a second order fit to acceleration response is shown in figure 4. The polynomials $B(s)$

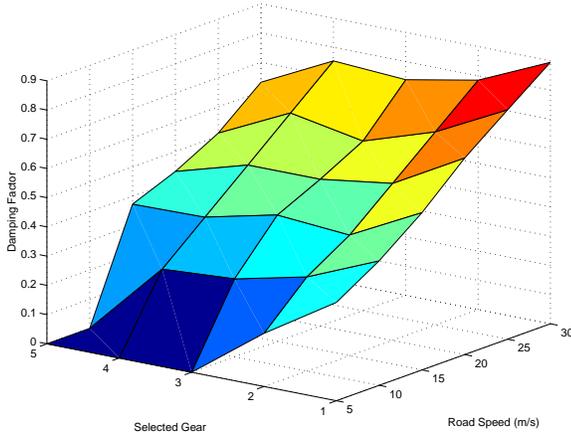


Fig. 4. Experimental vehicle damping ratio response surface.

and $A(s)$ can be obtained. For example, at 15mph in 2nd gear;

$$\frac{v(s)}{u(s)} = \frac{B(s)}{A(s)} = \frac{1615.7}{s^3 + 4.3s^2 + 521.8s} \quad (15)$$

As part of the model, a second order approximation will be added to equation 15 to model the delay relating to the manifold fill delay. The resulting polynomials were

$$\frac{B_1(s)}{A_1(s)} = \frac{B(s)P_{num}(s)}{A(s)P_{den}(s)} \quad (16)$$

where $A_1(s)$ and $B_1(s)$ are the new polynomials representing the system and $P_{num}(s)$ and $P_{den}(s)$ are the numerator and denominator respectively of the Pade approximation. The order difference in the process polynomials is $\alpha = n_a - n_b = 5 - 2 = 3$, and defining $\beta = 0$, the order of

the controllers $S(s)$ and $R(s)$ and the closed loop characteristic equation $F(s)$ can be found as in equation 14. Then, $n_s = 4$, $n_r = 4$ and $n_c = 9$. This leads to the matter of determining the value of nine roots for the characteristic equation. The lower order approximations have been used to determine a tractable number of poles to place in the controller design. The original Matlab/Simulink model is computationally too slow for the iterative procedure inherent in evolutionary optimisation. Consequently the metamodel is adopted for its speed of execution. The pole locations will be the decision variables in the multi objective genetic algorithm (MOGA), since those are the unknowns in equation 14 and was used to calculate directly the values of the coefficients for the polynomials $R(s)$, $T(s)$ and $S(s)$. The gain of $T(s)$ was also included as a decision variable. The coding of the variables was real, since that allows more natural data handling and is more efficient. The objectives set the goals to reach and ensure that every selected individual satisfies the specifications. The performance objectives have already been described, and relate to overshoot, settling time etc. The initial conditions (selected gear, road speed) in addition to reasonable real-life variations in the mechanical parameters (lash etc.) for the driveline nonlinear model were changed randomly for each individual in each generation, in such a way that the best controllers are the ones that could perform adequately under wide system parameter and condition variations. The variation in lash in particular replicates one of the fundamental characteristics of the ageing of the system. A further addition to this approach was the variation in road conditions and vehicle loading from one to four occupants. This approach was intended to achieve as far as possible, a robust controller. Finally, the minimisation of the control energy was included amongst the objectives in order to achieve a feasible, efficient controller. The bounds of the random variations were as follows:

- lash 0° to 30° at wheels
- road conditions from μ 0.4 to 1
- vehicle loading 1 to 4 occupants (standard occupant = 80kg)
- road gradient -10 to $+10\%$

This bounding set was later utilised to assess the robustness of the robust controller. The GA Toolbox for Matlab with the MOGA extension tools developed at the University of Sheffield [3] was utilised to perform the search procedure. The decision variables are in this case assigned to the controller pole placement positions.

VI. METAMODELLING FOR EVOLUTIONARY OPTIMISATION - RESULTS

A candidate controller of the form;

$$R(s) = s(s - 20.1 + 12.1i)(s - 20.1 - 12.1i)(s - 17.3)$$

$$S(s) = (s - 167.8)(s - 41.3 + 103.2i)(s - 41.3 - 103.2i)(s - 18.6)$$

$$T(s) = (s - 40.6 + 3.3i)(s - 40.6 - 3.3i)$$

$$(s - 37 + 1.71i)(s - 37 - 1.71i) \quad (17)$$

was selected from the family of potential solutions on the basis of its minimisation of all the objectives stated in the objective function, and its overall driving “feel”. The controller was simulated under varying initial conditions and mechanical parameters to verify its performance, and also assess its robustness. The predicted effects of ageing (for example on lash) were found in simulation of the closed loop system to produce acceleration responses which were within the bounds of desired “driveability”. Although experimental assessment of the controller in terms of varying lash was not possible, a number of step responses were taken under varying vehicle loading. A factorial study

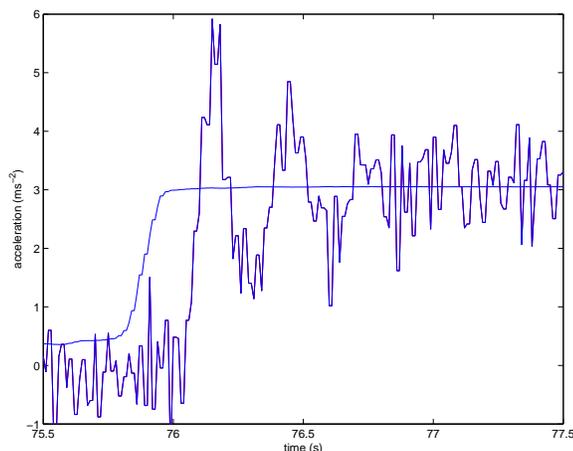


Fig. 5. Open loop unfiltered experimental step response, 1 passenger, 15mph, 2nd gear.

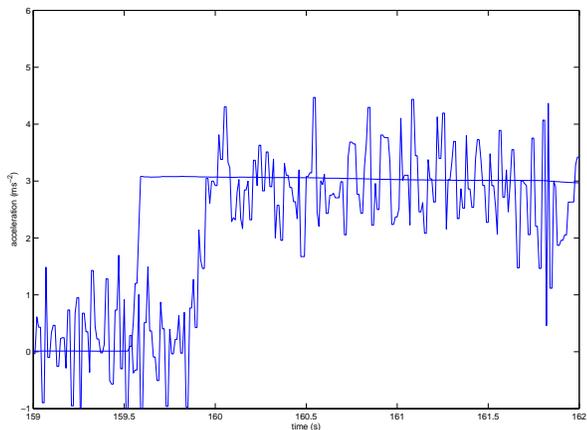


Fig. 6. Closed loop unfiltered experimental step response, 1 passenger, 15mph, 2nd gear.

was undertaken at a range of road speeds and selected gear ratios, with vehicle loading being chosen randomly. A typical open loop response is shown in figure 5 with the corresponding closed loop response being shown in figure 6. The response is satisfactory in terms of driveability both in the step response and the delay time of the vehicle acceleration. The controller was found to be robust to changes

in vehicle loading, and showed an excellent response across the vehicles entire operating range. Although it was not possible to vary the vehicle lash, simulation results predict that the controller is again robust for levels of lash up to 30° at the wheels.

VII. CONCLUSIONS

This project had two initial objectives, firstly to examine the application of the Response Surface Methodology to rapid control design prototyping, and having successfully achieved this goal, the second objective was to use the RSM to examine the potential of electronic throttle control to shape the acceleration response of an experimental vehicle. The timescale from experimental data capture through RSM analysis to experimental verification and assessment of the electronic throttle’s potential for vehicle acceleration shaping was under four days. At this point, a judgement upon the viability and potential of the project could easily be made with confidence. In this context, the use of metamodeling becomes extremely attractive. The technique has allowed an examination of the system control potential to be made at the start of the project, benefitting both the confidence of the industrial partners, and giving a realistic benchmark of potential performance. The controller derived by the RSM is immediately useable on the experimental vehicle, providing a demonstration facility at project inception. Some other benefits of this development tool are also significant. A controller is quickly available for verifying the mechanical and electrical components of the control system, giving a stable platform for subsequent controllers as and when they become available. The controller as designed via the RSM is simple and low order by nature, and thus can be installed on a very simple microcontroller. Finally, a quick and cheap assessment of a system’s potential can be rapidly made in order to support project development proposals. The design of a useable implementation has been achieved on a time varying process with significant time delays and nonlinearities. A closed-loop controller was subsequently derived using the pole placement method. Multi objective evolutionary algorithms were applied to find the optimal location of the poles for the characteristic equation. Random initial conditions were applied to each generation to achieve robust solutions. The response of the selected controller shows a dramatic improvement over the open loop response, and does not require tuning depending on variations in the system parameters. The controller response also proves to have a better performance than the results obtained in the literature. The combination of the pole placement method with MOGA as a technique for driveline controller optimisation results in an efficient design procedure, where the lack of knowledge of the possible solutions does not necessarily affect the result of design process. Although an accelerometer was fitted to the vehicle for verification purposes, the implemented controller worked with the vehicle speed feedback signal which was readily available.

The effect of the closed loop controller upon the driver and passengers was perceived to be extremely benefi-

cial. It was possible to repeat driven routes with the controller both engaged and disengaged. Although this method is extremely subjective when compared to rise-time/overshoot/settling time analysis, for the end user (a variety of drivers), the effect of the controller was found to give a distinct improvement to the driving experience.

REFERENCES

- [1] Best M.C., "Nonlinear optimal control of vehicle driveline vibrations", *UKACC International Conference on Control '98*, pp 658-663, Swansea, UK. September 1-4, 1998.
- [2] Camacho E.F., "Model Predictive Control", *Springer Verlag London Ltd.*, ISBN 3-540-76241-8, 1999.
- [3] Chipperfield A.J., Fleming P.J. and Polheim H.P., "The Genetic Algorithm Toolbox for MATLAB", *Department of Automatic Control and Systems Engineering, University of Sheffield, U.K.* Version 1.2.
- [4] Hicks C.R. and Turner K.V.Jr., "Fundamental concepts in the design of experiments", *Oxford University Press*, New York. ISBN-0-19-512273, 1999.
- [5] Kiencke U., Nielsen L., "Automotive Control Systems", *Springer-Verlag*, Berlin-Heidelberg-New York, ISBN 3- 540-66922-1, 2000.
- [6] Michalewicz Z., "Genetic algorithms + data structures = evolution programs", *Springer-Verlag*, 3rd ed., 1999.
- [7] Myers R.H., and Montgomery D.C., (1995) "Response surface methodology: process and product optimization using designed experiments", *John Wiley and Sons Inc. USA*, ISBN 0-471-58100-3.
- [8] K. Ogata, "Modern control engineering", *Prentice-Hall International Inc.*, New Jersey. ISBN 0-13-598731-8, 1990.
- [9] Richard S., Chevrel P. and Maillard B., "Active control of future vehicles drivelines", *38th IEEE Conference on Decision and Control*, IEEE, Piscataway, NJ, USA; vol.4, pp. 3752-3757, 1999.
- [10] Richard S., Chevrel P., de Larminat P. and Marguerie B., "Polynomial pole placement revisited: application to active control of car longitudinal oscillations", *1999 European Control Conference*, Karlsruhe, Germany, 1999.
- [11] Rossi C., Tilli A., and Tonielli A., "Robust control of a throttle body for drive by wire operation of automotive engines" *IEEE Transactions on Control Systems Technology*, vol.8, no.6, November 2000.
- [12] Stewart P and Kadiramanathan V., (1999) "Dynamic control of permanent magnet synchronous motors in automotive drive applications", *1999 IEEE American Control Conference*, San Diego, USA. pp.1677-1681, June 2-4, 1999.
- [13] Stewart P and Kadiramanathan V., (2001) "Dynamic model reference PI control of flux weakened permanent magnet AC motor drives" *IFAC Journal of Control Engineering Practice*. In Print
- [14] Stewart P and Fleming P., (2001) "The Response Surface Methodology for real-time distributed simulation.", *IFAC Conference on New Technologies for Computer Control.*, Hong Kong, China. pp.128-133, November 19-22, 2001.
- [15] Zavala J.C., Stewart P., and Fleming P.J., (2002) "Multiobjective automotive drive by wire controller design." *IEE / IEEE International Conference on Computer aided Control System Design*, Glasgow, Scotland, pp69-73, 18-20 September, 2002.

SENSOR INFORMATION FUSION FOR THE NEEDS OF FAULT DIAGNOSIS IN MARINE DIESEL ENGINE PROPULSION PLANT

RADOVAN ANTONIĆ¹, ZORAN VUKIĆ², ANTE MUNITIĆ¹

¹ Split College of Maritime Studies, Univ. Split
Zrinsko-Frankopanska 38, 21000 Split, Croatia
e-mail: antonio@pfst.hr

² Faculty of Electrical Engineering and Computing, Univ. Zagreb
Unska 8, 10000 Zagreb, Croatia

Abstract

Complex industrial processes like Marine Diesel Propulsion Plant (MDPP) have complex interrelations and interdependencies between variables and parameters. This characteristic could be used in estimating unknown or unmeasured variables from the information gathered by other measurements and sources using information fusion by means of a soft computing methods. In the paper, a structural analysis approach to identifying most relevant variable interrelations, components or subsystems of MDPP with inherent redundant information has been proposed. Sensor information fusion method was chosen to be using artificial neural networks (ANN). The paper presents proposed ANN with structure and learning algorithms. Simulation have been carried out in Matlab-Simulink environment with engine speed estimation example.

Keywords

Diesel engine propulsion plant, structural description, redundant information, sensor fusion, neural network

1. INTRODUCTION

Diagnostic and control systems of marine diesel propulsion plant require a large number of different sensors with different measuring types and locations at various critical points on the propulsion engine and its subsystems (temperatures, pressures, flow rates, levels, metal content of the lubricating oil, water content in the fuel oil and more). The data from sensors are collected and transmitted to the processing units.

The main purpose of most signal processing is to yield knowledge of a situation so that proper decisions can be done. Many of these signals should be combined in some way to enable decisions of such conditions as emergency states, when to change oil, time to repair or replace parts, engine efficiency etc.

In some specific situations human intuition, heuristic knowledge and experience have to be fused together with sensor data for good plant estimation (overall engine efficiency, degradation of oil condition, fault conditions,..). One

effective approach in such case is information fusion that will be discussed in the paper.

In the cases when a sensor fails to operate or operates with faults, sensor information fusion methods are needed to reconstruct the lost signals - information. Aiming to use sensor information fusion with existing sensors the need for exploring possible redundancies inherent to the system structure is evident. One suitable method is structural description or analysis of the system decomposing it into functional dependent end related components or subsystems. This approach for MDPP will be presented in the paper. Sensor information fusion method was chosen to be using artificial neural networks which are very suitable in the case of on-line gathered data.

Simulation example will be given for marine diesel engine speed estimation using redundant relations and data.

2. STRUCTURAL DESCRIPTION OF A SYSTEM: GENERAL APPROACH

One can consider a system S like union of its functional components $\bigcup_{i=1}^n C_i$, each of them establishing some relations or constraints f_i between a set of variables and parameters (known or unknown) z_j of the system, i.e. :

$$f_i(z_1, z_2, \dots, z_p), \quad 1 < p \leq m$$

where f_i can represent dynamic, static, linear or non linear relation, crisp or fuzzy rules, empirical or any other relation-constraint.

Structural model of the system can then be represented with a set of constraints: $F = \{f_1, f_2, \dots, f_n\}$ and a set of variables and parameters $Z = \{z_1, z_2, \dots, z_m\} = Z^K \cup Z^X$ to which constraints are valid. $Z^K = U \cup Y \cup C$ is a set of known variables and parameters, where U represents a set of control variables, Y is a set of measured outputs and C is a set of known constant parameters. Z^X is a set of unknown variables and parameters of the system.

Now, the structural model of the system can be represented by directed graph with nodes and connecting arcs $G(F, Z, A)$. The elements of a set of arcs in such graph $A \subset (F \times Z)$ are defined with the following mapping scheme - binary relations:

$$\begin{aligned} A &: F \times Z & \{0,1\} \\ A^K &: F \times Z^K & \{0,1\} \\ A^X &: Z^X \times F & \{0,1\} \end{aligned} \quad (1)$$

For more details see (Izadi-Zamanabadi 1999).

3. STRUCTURAL ANALYSIS OF MARINE DIESEL PROPULSION PLANT – REDUNDANT DATA AND RELATIONS

A structural analysis model to identify most relevant variable interrelations, components or subsystems of MDPP with inherent redundant information which could be used in fusion process will be explored.

3.1 Structural description of MDPP

The main purpose of the structural description of MDPP here is to explore some inherent redundant relations which can be used in calculating unknown or unmeasured variables using sensor information fusion method.

Figure 1 shows the structure of MDPP with its main structural components: C_1 - diesel engine dynamics, C_3 - engine shaft dynamics, C_5 and C_6 - propeller shaft dynamics, C_8 - ship speed dynamics, C_{10} - hull dynamics, and corresponding sensors: fuel index

sensor C_2 , engine speed sensor C_4 , pitch propeller sensor C_7 and ship speed sensor C_9 .

Relations and constraints between variables and parameters can be obtained in various ways: by mathematical modelling, by simulation, using experimental data, eliciting expert's and operator's knowledge, etc. For details see (Antonić and Radica 1991; AntoniĆ et al. 2000; AntoniĆ and Vukić 2002; Vukić et al. 1998; Izadi-Zamanabadi 1999).

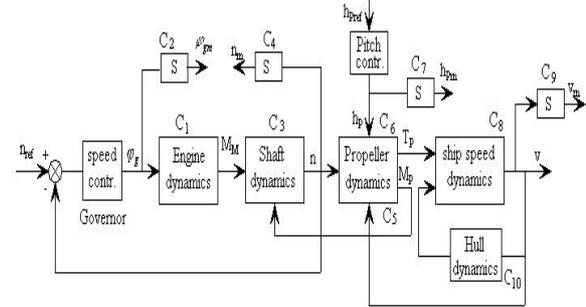


Figure 1: The structural diagram of diesel propulsion plant

where: n_{ref} - engine reference speed (set value); n - engine speed; φ_g - fuel link position; h_{pre} - propeller pitch set value; h_p - propeller pitch; v - ship speed; K_M , T_M - engine gain and time constant; M_p , T_p - propeller torque and thrust;

v_a - advance propeller speed; R_u - total hull resistance
The structure of MDPP in Figure 1 can be represented as union of its components : $\bigcup_{i=1}^{10} C_i$.

A set of constraints / structural relations is:

$$F = \{f_1, f_2, \dots, f_{10}\} \quad (2)$$

A set of known measurable variables and parameters is: $Z^K = \{\varphi_{gm}, n_m, h_{pm}, v_m, K_M\}$ (3)

A set of unknown variables and parameters is: $Z^X = \{\varphi_g, n, h_p, v, M_M, M_P, T_P, R_u\}$ (4)

The measuring noise is here neglected so:

$$n_m = n; \varphi_{gm} = \varphi_g; h_{pm} = h_p; v_m = v. \quad (5)$$

Adequate structure graph of the MDPP with variable and parameter relations is shown in figure 2.

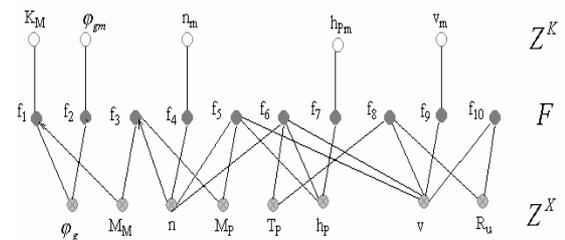


Figure 2: Structure graph of MDPP system with variable and parameter relations

3.2 Redundant relations and information fusion

From the structural graph of MDPP system one can get analytical redundant relations between variables and parameters: direct relations and indirect or derived ones (with sensor information fusion and some reasoning method). For direct relations structural constraints are applied only to known - measured variables i.e. to subset Z^K , while derived relations are those to which structural constraints of unknown - unmeasured variables are applied, i.e. to subset Z^X .

Indeed, derived redundant relations are more interesting, because they result with analytical redundancy what is a key point for information fusion. These are frequently based on the human expert knowledge and operator experience.

It is evident from the structure graph model of MDPP, that there are redundant relations and information which can be used in case of faulty sensors.

For instance, in the case of engine speed sensor fault (component C_2 in structural diagram) the value of the engine speed could be estimated i.e. calculated using information fusion from other sources (C_5 and C_6).

Unknown variable can be estimated by integrating several other measurements into a single robust estimator (software sensor). The fusion of data from different sensors will add new valuable information that would be otherwise unavailable. The need of data fusion arises also from the fact the information gathered is often incomplete, uncertain, imprecise or may be from a faulty sensor. There are several possible methods for data fusion and the very effective one is artificial neural network approach.

4. INFORMATION FUSION IN MDPP USING ANN APPROACH – SIMULATION EXAMPLE

The ability of ANN to learn from experience i.e. from history of data during on-line operation is making them the preferred choice for process modelling with intrinsic variable and parameter interrelations. In the above structural description of MDPP the redundant relations between variables and parameters were illustrated. Some of them will be used in the information fusion example.

4.1 Engine speed estimation using information fusion: Speed sensor faulty - simulation example

Engine diagnosis and control system needs speed information during normal operation and gets it continually from speed sensor.

In the case of speed sensor failure it would be desirable to have a system that could estimate engine speed (most critical variable in closed loop speed control) from various sets of inputs i.e. information sources giving redundancy in speed information and thus leading to more robust control system. That is

especially important if all speed sensors (usually two) are in faulty conditions.

The required engine speed value could be estimated on-line from other variables which are related to it (see Figure 2) : propeller torque M_p or propeller thrust T_p , ship speed v , propeller pitch h_p if the propeller is controllable (CPP).

Figure 3 a and b illustrate engine speed estimation from other known variables - signals measured on-line (M_p , T_p , v).

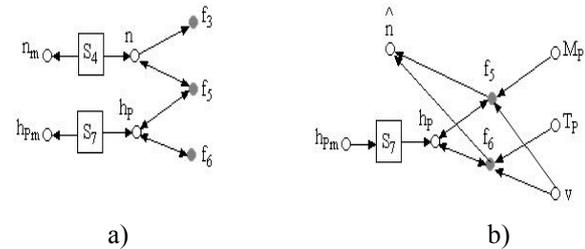


Figure 3: Engine speed estimation using information fusion

4.2 Neural network structure and learning algorithms

In the engine speed estimation example three independent input signals to the ANN and one output signal which should be the best estimate of engine speed in case of faulty sensor were used.

The data from different sources are usually pre-processed (data normalization, filtering, principal component analysis, etc.) before applied to the ANN for fusion purpose.

The ANN, in this experiment, was organized in two processing stages i.e. two ANN were designed and used (Figure 4).

The first stage consists of estimation ANN and is for feature extraction from input signals. The second stage consists of ANN for information fusing i.e. decision making and selecting the best estimate from the first ANN.

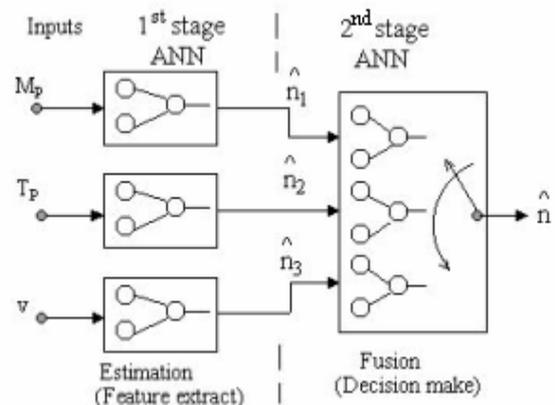


Figure 4: Concept of ANN for engine speed fusion

The first stage consists of three identical feed forward NN (in Figure 5a shown only one for input

variable M_p) each with one hidden layer with log-sigmoid transfer function and one output layer with linear transfer function. The second stage consists of self-organising NN with one competitive layer with three inputs (these are outputs from the first stage) and one output ADALINE stage (in figure 5 b). There are three neurones in competitive layer and only one is a winner in a time. Euclidean distance measure (see Antonić and Vukić 2002) in decision making i.e. choosing the best estimate in each time step was used. In the estimation stage of NN, 3 inputs are fed (propeller torque M_p , propeller thrust T_p and ship speed v) to estimate engine speed n .

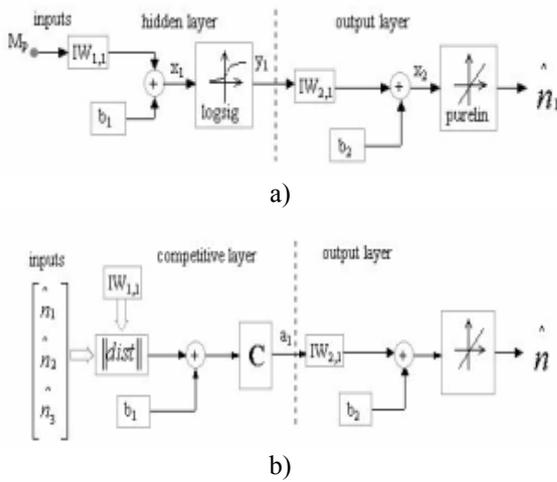


Figure 5: Structure of ANN for engine speed estimation

The mse (the mean squared error between the target i.e. expected values and the network outputs – estimated values) performance function is chosen as a criterion.

$$mse = \frac{1}{N} \sum_{k=1}^N (n(k) - \hat{n}(k))^2 \quad (6)$$

Performance goal was set to $mse = 0.01 \text{ rad}^{-1}$.

In minimising performance function the gradient descent back-propagation learning algorithm for updating network weights and biases with adaptive learning rate was used.

$$x(k+1) = x(k) - \alpha(k)g(k) \quad (7)$$

where $x(k)$ is a vector of current weights and biases, $g(k)$ is the current gradient and $\alpha(k)$ is the learning rate.

For comparison purpose we used two learning algorithms:

- Quasi-Newton (BFGS) learning algorithm,

$$x(k+1) = x(k) - H^{-1}(k)g(k) \quad (8)$$

$H(k)$ is the Hessian matrix (second derivatives) of performance function at the current values of the weights and biases.

- Levenberg-Marquardt learning algorithm

$$x(k+1) = x(k) - [J^T J + \mu I]^{-1} J^T e \quad (9)$$

where J is the Jacobian matrix which contains first derivatives of the network errors with respect to the

weights and biases, e is a vector of network errors, μ is a scalar.

4.3 Simulation results in engine speed estimation

The training set used for the proposed ANN is obtained from the real diesel engine propulsion plant simulator PPS2000 (Norcontrol) with propulsion diesel engine MAN B&W type 5L90MC with maximum power of 18.000 kW installed on the very large crude carrier, (fully loaded). We've got training set values with diesel engine working in four basic operating regimes – modes (table 1): Full ahead (with engine power of 100 %), Half (engine power of 75 %), Slow (engine power of 50 %) and Dead slow (engine power of 25 %).

Table 1: Simulated engine data for training ANN

Engine regime	Engine power (%)	Engine speed - n (rad/s)	M_p (Nm) $\times 10^6$	T_p (N) $\times 10^6$	Ship speed v (m/s)
Full Ahead	100	7.74	2.20	1.46	7.71
Half	75	7.02	1.90	1.21	7.06
Slow	50	5.14	1.05	0.66	5.11
Dead slow	25	3.10	0.41	0.26	3.10

The second part of the simulation was carried out by using Matlab/Simulink environment.

After training the ANN given in Figure 5 using training data set from table 1, we've got very good results for engine speed estimates in four operating points (Full Ahead, Half, Slow, Dead slow). These are presented in table 2 and Figure 6. The differences between speed estimates are very small (with $mse: 3.3 \times 10^{-3}$ with M_p data, 9.98×10^{-4} with T_p and 6.16×10^{-4} with v data set).

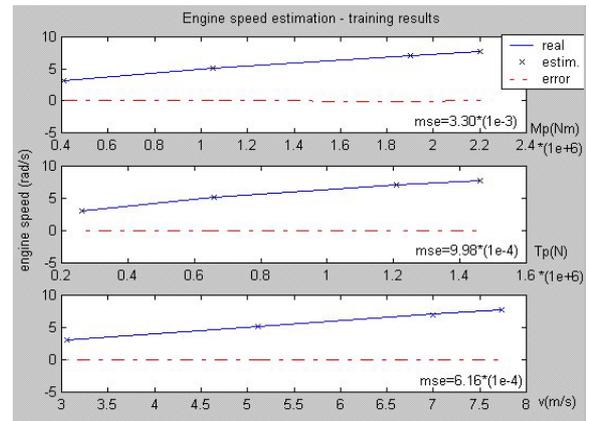


Figure 6: Engine speed estimation with training data set from M_p , T_p , v

Table 2: Estimated engine speed n from M_p , T_p , v

Engine speed (target) n (rad/s)	Estimated speed from other signals (with training data)		
	M_p	T_p	v
7.740	7.682	7.712	7.692
7.020	7.114	7.065	7.073
5.140	5.097	5.114	5.124
3.100	3.114	3.113	3.118

Comparing results obtained during training session of NN with two different learning algorithms: Levenberg-Marquardt (LM) and Quasi-Newton we've noticed very little difference (table 3).

Nevertheless, we prefer LM learning algorithm because the estimation error (mse) and training period (epochs) were a bit lesser.

Table 3: Comparison results of two learning algorithms in training NN

	Levenberg-Marquardt			Quasi-Newton		
	M_p	T_p	v	M_p	T_p	v
mse	3.30 $\cdot 10^{-3}$	9.98 $\cdot 10^{-4}$	6.16 $\cdot 10^{-4}$	3.30 $\cdot 10^{-3}$	9.79 $\cdot 10^{-4}$	9.61 $\cdot 10^{-4}$
epoch	>500	115	26	>500	118	35

Performance goal (mse = 0.01) for the best engine speed estimate (with T_p data set) was reached in very short time (4.17 s) i.e. after only 115 epochs of training.

Applying testing data set to ANN concurrently for three inputs: M_p , T_p and v , less accurate results were obtained (Figure 7) but nevertheless useful for practical use, except those estimated from ship speed data where the mean squared error was 11.55 %. The best results were obtained from propeller thrust measurement T_p (mse = 1.73 %).

The largest discrepancy between training and testing results were obtained for ship speed signal, maybe because of small training set.

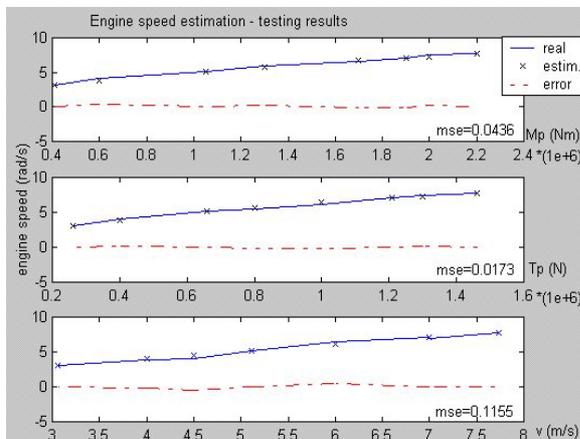


Figure 7: Engine speed estimation with testing data set from M_p , T_p , v .

In each time step, the designed ANN chooses the best estimate on its output so the final results were acceptable. Testing example with engine power of 100 % and expected real value of $n = 7.74$ rad/s: the best speed estimate, was with T_p data: $n_e = 7.712$ rad/s (see Figure 8a). We also tested ANN output in the case of lost one or even two of three input signals and have got good engine speed estimate. Figure 8b illustrates situation with two input signals missed (sensor faults). The ANN output was $n_e = 7.785$ rad/s (Figure 8b).

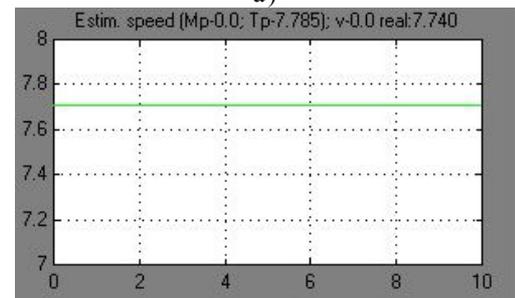
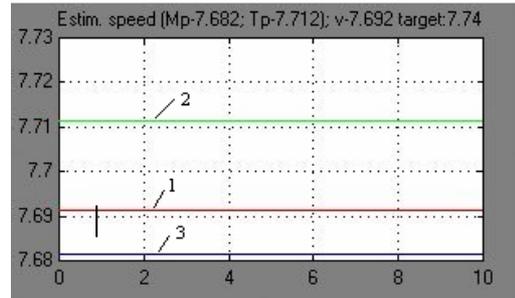


Figure 8: Engine speed estimate (best ANN output)

Applying testing data within all operating regions is illustrated in Figure 9. Test results for engine speed estimate are fairly good for M_p and T_p .

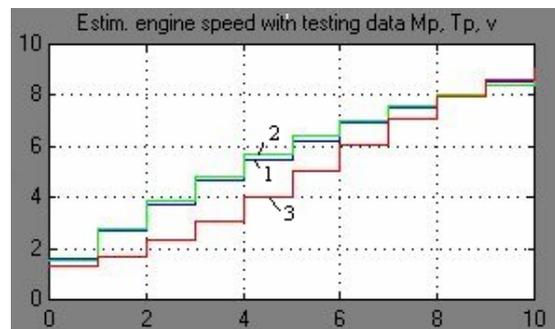


Figure 9: Estimating engine speed with testing data M_p , T_p , v within the operating region

Data fusion of three signals with expert modification of contribution coefficients on engine speed with $K_{M_p} = 0.34$, $K_{T_p} = 0.36$, $K_v = 0.30$ had given quite good estimate (see Figure 10).

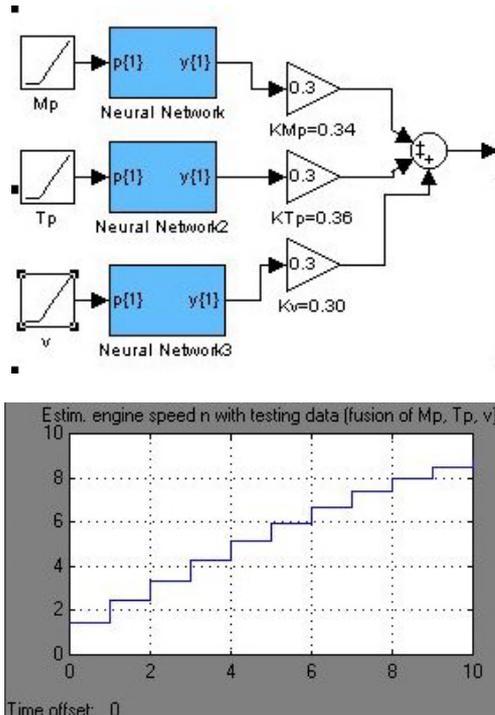


Figure 10: Estimating engine speed with data fusion of M_p , T_p , v

Finally, three testing cases with M_p as input signal to the ANN has been studied in parallel and the output (speed estimate) was recorded in the diagram (see Figure 11):

The first case was with M_p as only input signal. Input signal in the second case was M_p with the added noise (zero mean Gaussian with variance of 0.02). In the third case, the disturbance signal (sine wave of amplitude of 0.1 and frequency of 1 rad/s) was added to M_p . We could conclude that proposed fusion scheme is rather robust to noise and disturbance in input signals.

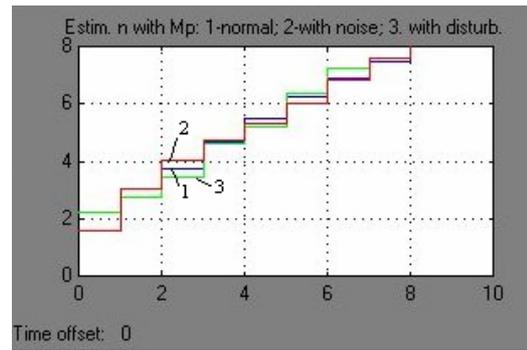
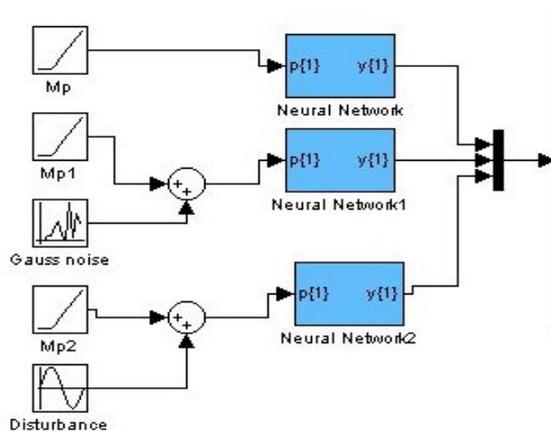


Figure 11: Estimating engine speed with M_p : 1 – normal ; 2-with Gauss noise; 3-with sine disturbance

CONCLUSION

Sensor information fusion concept is becoming more and more attractive especially in the area of diagnostics and control systems. Some important advantages of using information fusion in combination with soft computing technologies like artificial neural networks, fuzzy logic, genetic programming could give more robustness, reliability, fault tolerance and intelligence to control systems.

The structural approach is presented and applied to the marine diesel engine propulsion plant as an effective method to identify the subsystems with inherent redundant information. Based on that analysis we proposed ANN for information fusion process which consists of two stages: The first stage is an estimation ANN for feature extraction from input signals and the second stage is for information fusing i.e. decision making and selecting the best estimate from the first stage ANN. We tested it with the simulation example. Diesel engine speed was estimated on the basis of three other signals: propeller torque M_p , propeller thrust T_p and ship speed v . It was shown that good speed estimation could be obtained using other available information in the case of faulty speed sensor. Only a part of the obtained results was presented in the paper.

The proposed fusion scheme was also tested with noise and disturbance signals added to the M_p input signal and concluded fairly good scheme robustness.

Better results would probably be obtained if larger sets of training and testing data were used. The generalisation scheme in the sensor information fusion within MDPP will be of our interest in the near future.

REFERENCES

- Antonić, R. and G. Radica. 1991. "Diesel Engine Diagnostic Simulation Model based on Trend Curves", Proceedings of 33rd Symposium ETAN in Marine, Zadar, 29-31.
- Antonić, R., Z. Vukić, and I. Kuzmanić. 2000. "Basic Principles and Techniques of Expert Knowledge Elicitation for the Needs of Technical Systems Diagnostics", Brodogradnja 48,4, 330-337.
- Antonić, R. and Z. Vukić. 2002. "Knowledge Representation in Diagnosis and Control of Marine Diesel Engine", Brodogradnja 50,1, 57-66.
- Izadi-Zamanabadi, R. 1999. Fault Tolerant Supervisory Control - System Analysis and Logic Design, PhD Thesis, Department of Control Eng. , Aalborg Univ. , Denmark.
- Samarsooriya, V.N.S. and P.K. Varshney. 2000. "A fuzzy modelling approach to decision fusion under uncertainty", Fuzzy sets and systems 114, 59-69.
- Vukić, Z. H. Ožbolt and M. Blanke. 1998. "Analytical Model-based Fault Detection and Isolation in Control Systems", Brodogradnja 46, 3, 253-263
- Zilouchian, A. and Mo. Jamshidi. 2001. Intelligent Control Systems Using Soft Computing Methodologies. CRC Press, 2001



Radovan Antonić received his B.Sc. degree in electrical engineering from University of Split in 1980, and M.Sc. and Ph.D. degree in electrical engineering from University of Zagreb in 1986 and 2002. respectively. He is IEEE member and research interest include control systems, fault tolerant control, expert systems, specially in marine applications.

Author or co-author of about 40 bibliographical units (scientific and professional conference and journal papers, research projects, text books, etc.). Currently, he is a senior lecturer at Split College of Maritime Studies.



Zoran Vukić received his B.Sc, M.Sc. and Ph.D. in electrical engineering from University of Zagreb in 1972, 1977 and 1989 respectively. He is professor at the University of Zagreb, Department of Control and Computer Engineering in Automation. His research interest is in fault tolerant control, adaptive & robust control, nonlinear control, guidance and control of marine vehicles.

He is member of IEEE Control Sys. Soc., IEEE Oceanic Eng. Soc. and IFAC Technical Committees on Marine systems and Adaptive and learning control. Author or co-author of more than 100 bibliographical units (conference and journal papers, research projects, referee reports, textbooks etc.).



Ante Munitić received his first B.Sc. in Electric and Energetic Engineering in 1968, and his second in 1974, his M.Sc. degree Organisation System and Cybernetics Science (Operational Research) in 1978, and his Ph.D. of Organisation Science (System Dynamics), in 1983. He is currently a full Professor of Computer and Informatics Science at the University of Split.

He has published over 100 papers on system dynamics modelling and simulation, operational research, marine automatic control system and The Theory of Chaos. He has published two books: "Computer Simulation with help of System Dynamics" and "Basic Electric Energetic and Electronics Engineering".

DERIVING A HYBRID ALGORITHM TO SOLVE HEAT FLOW PROBLEMS

S.BERRY and V.LOWNDES

*School of Computing and Technology
University of Derby*

Abstract: The Heat and Fluid Flow problems commonly occurring within the design of heat exchangers are difficult and time consuming to solve using traditional techniques. This paper describes an investigation into the suitability of the use of heuristic optimisation techniques, Genetic Algorithms, Tabu Search and Simulated Annealing, in the solution of complex heat flow problems. The derived Hybrid Algorithm was constructed by combining elements from all three approaches and the final algorithm acted to reinforce their strengths and minimise their weaknesses to produce an efficient and effective algorithm.

Keywords: Heuristic Methods, Hybrid Techniques, Heat Flow.

1 PROBLEM DEFINITION

The heat flow across a slab, ABCD see figure 1

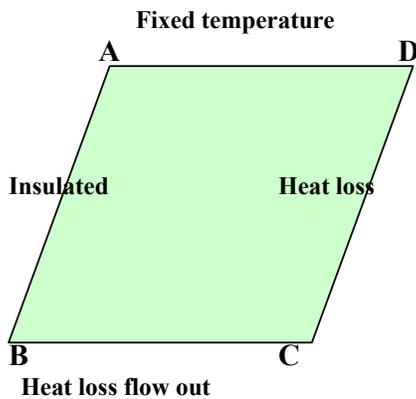


figure 1

is modelled by the partial differential equation:-

$$\frac{\partial^2 z}{\partial x^2} + \frac{\partial^2 z}{\partial y^2} = 0$$

which together with the respective boundary conditions:-

$$\text{AB} \quad \frac{\partial T}{\partial x} = 0$$

$$\text{BC} \quad k \frac{\partial T}{\partial y} = -\phi_0$$

$$\text{CD} \quad -k \frac{\partial T}{\partial x} = h(T - T_0)$$

$$\text{DA} \quad T = T_0$$

enables the determination of the heat profile, steady state temperatures, across the slab.

2 SOLUTION METHODOLOGY

The solution to this problem was derived using elements from:-

Genetic Algorithms
Simulated Annealing
Tabu Search

The final form of the algorithm was constructed to take advantage of the strengths of each of these methods and to minimise the effect of their weaknesses.

2.1 The initial GA approach

The initial stage was concerned with an investigation into the use of Genetic Algorithmic techniques, thus the key stages in a Genetic Algorithm approach to solving this problem had to be defined, these were: -

The form of the GA strings

Here each string consisted of 24 real numbers, each representing the temperature at one of the grid points on the slab.

Selection for crossover

The strings were selected on a proportional basis, to their fitness, to be included in the population used to generate the next set of strings, using crossover.

The crossover technique

In this problem a block cross over was employed, that is the information from a randomly chosen block was exchanged by the chosen pair of strings. This method was employed because it better represented the geometry within the problem.

Elitist approach

Following crossover one of the new strings was chosen randomly to be replaced by the existing best string.

2.1.1 Implementation

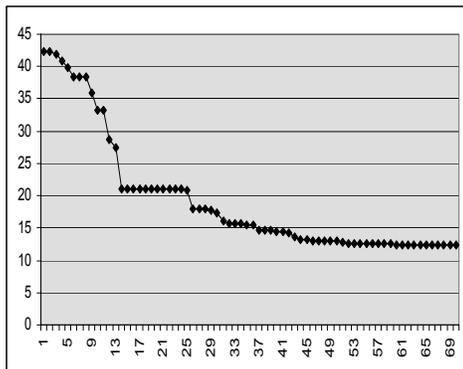
These procedures were combined to give the basic approach this was transformed into an iterative approach through the addition of the routine:-

when the (GA) process has converged, a new set of strings close to the existing best solution are generated and the process repeated, thus leading to an Iterative Genetic Algorithmic (IGA) approach to this problem, note that the search space is reduced at each successive iteration.

2.1.2 Results

These procedures were applied with varying sizes of initial population and to problems of varying difficulty, where difficulty is assumed to be related to the range of temperatures in the initial population.

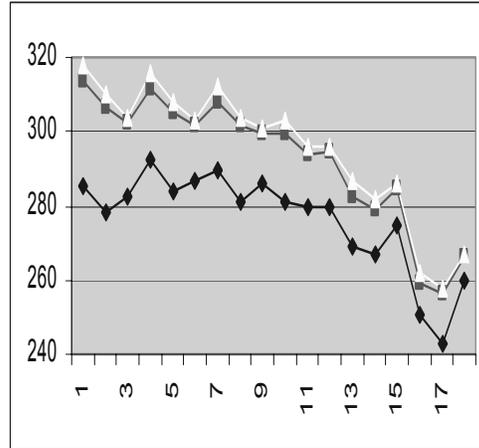
Regardless of the initial population the Genetic Algorithm adopted an asymptotic behaviour, but tending towards a residual of 10, the target was to obtain a residual of about 0.1. See graph 1.



Graph 1: Residual values

The Genetic Algorithm acted to produce solutions which were aligned with (parallel to) the temperatures at the optimal solution, but it was not able to move very close to the optimal solution, see Graph 2 where the “top set of

points” is at the solution, the lower set of points is the converged solution, and the third set of points has a residual of less than 1.

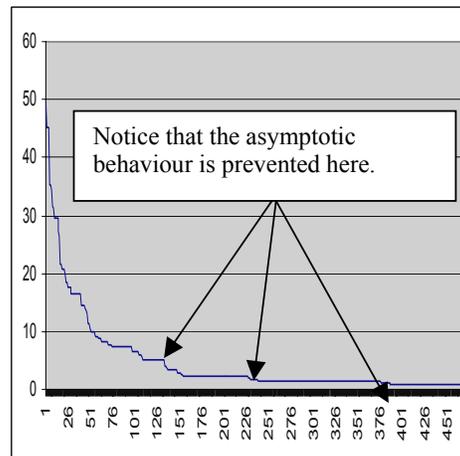


Graph 2: Temperatures across the slab

Thus it can be seen that although the Iterative Genetic Algorithm did not in itself lead to a satisfactory solution methodology it did act to “line up” the temperatures across the slab. Therefore this approach was adapted by incorporating elements from Simulated Annealing to attempt to overcome this convergence problem.

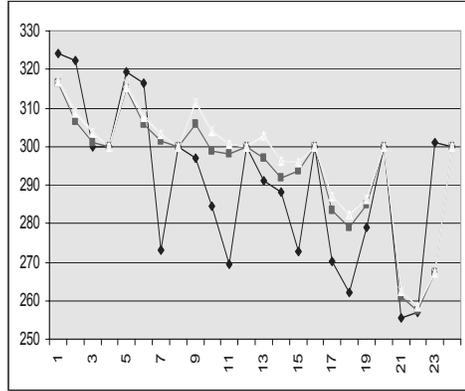
2.2 Refined Iterative Genetic Algorithm (RIGA)

Aspects of Simulated Annealing are introduced into the methodology at the stage when a new solution set is generated. In the IGA approach the search space is narrowed at each iteration, in the refined methodology, the search space is randomly widened. This has the effect of overcoming the asymptotic “early” convergence of the IGA approach, replacing Graph 1 with Graph 3.



Graph 3: Residual values using restart

This procedure has acted to overcome the original asymptotic “early convergence” enabling the process to produce a solution with a residual less than 0.1. See graph 4 where the optimal solution is indicated by the “triangular point markers”.



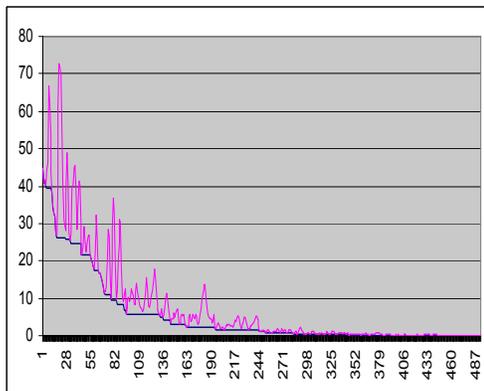
Graph 4: Temperatures across plate

2.3 The Hybrid Algorithm

The RIGA obtained a solution to the problem to the required accuracy however the implicit asymptotic behaviour of the GA acted to slow down the process. Therefore the next stage in the development of the hybrid algorithm investigated the incorporation of a Tabu Search.

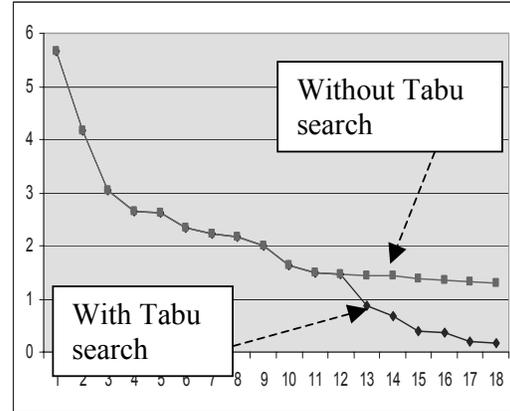
The RIGA algorithm was employed until the residual was reduced to a value between 0.5 and 1.5, (where the asymptotic behaviour has not yet started to dominate the process). At this point a varying step size Tabu Search is implemented.

Graph 5 shows the search space investigated by the Tabu Search procedure where the current point is shown together with the current best point.



Graph 5: Tabu search

The Tabu search acted to improve the search for the optimal solution, see graph 6. This shows how the Tabu Search prevents the asymptotic behaviour of the RIGA algorithm, leading to the Hybrid Algorithm (MA).



Graph 6: Tabu search effect on Residual

3 COMPARISONS BETWEEN THE APPROACHES:

To enable comparisons between the two approaches RIGA, and the Hybrid Algorithm (RIGA incorporating Tabu Search), the run times have been normalised so that the Tabu search requires 1 time unit.

Six problems were solved using both approaches. The final residual obtained and the time to solve the problems were averaged and these results are summarised in Table 1.

Results	RIGA	MA
Average final residual	0.06	0.06
Time MA = 0.57 * Time RIGA		

Table 1: Comparing Methods

Thus this algorithm works efficiently to obtain the solution to the heat flow problem. The algorithm acts to combine the strengths of each of the constituent techniques and minimises their weaknesses.

4 CONCLUSIONS

This work has demonstrated the fact that heat flow problems can be solved using modern optimisation techniques. However it has also shown that the solution algorithm has to be derived around the needs of the problem under consideration. Here a Genetic Algorithm, alone, could not produce a satisfactory solution (not enough accuracy) but it could indicate the answer, see graphs 2 and 4 where the GA

found solutions which were parallel with the optimal solution. Therefore once this state has occurred, which relates to the asymptotic behaviour of the GA, an alternative methodology, a search methodology can be employed to get to the solution, here a Tabu search was used. Thus the efficient solution methodology, a Hybrid Algorithm, was derived by combining elements from a set of methods. This methodology is now being extended into the study of more complex heat flow problems.

BIBLIOGRAPHY

- Goldberg D.E, 1989, Genetic Algorithms; Addison Wesley.
- Reeves C (Ed), 1995, Modern heuristic techniques for combinatorial problems: McGraw-Hill.
- Corne D, Dorigo M, Glover F, 1999, New Ideas in Optimisation, McGraw-Hill.

DESIGNING A CONSTANT WORK IN PROGRESS PRODUCTION CONTROL SYSTEM

S.BERRY and V.LOWNDES

*School of Computing and Technology,
University of Derby*

Abstract: A “Constant Work in Progress” production control system acts by restricting the number of jobs present in a “workshop” at any one time (maximum number n jobs) with the manager determining the “best” value for n . This shows how simulation methods and measures of complexity can be used to design an optimal constant work in progress system for a firm. Section 1 validating the measures of complexity by simulating their performance in small manufacturing firms. Section 2 shows the results from the simulations showing how simulation can be used to derive optimal production control systems.

Keywords: Production Control, Simulation, Measures of Complexity.

1 QUEUEING/DECISION MAKING ENTROPY

The derivation an entropy formula and justification for the use of Entropy to measure complexity is given in Frizelle [Frizelle and Woodcock, 1995] here a simpler version is considered. A simpler version is appropriate because a “Flow Shop” describes the predominant production system in small manufacturing firms.

Consider a manufacturing system consisting of m stages where queues could arise at any of these stages. The firm will make a decision, which job to process next, if at the time when a job is finished at a stage the queue of waiting jobs at the stage is greater than 1.

Therefore the simulation aims to determines and the entropy measure indicate the number of times that a decision is made at each stage, where n_i is the number of decisions at stage i , and $N = \sum n_i$ gives the total number of decisions made during the simulation.

Then $p_i = n_i/N$ gives the probability that a decision needed to be made at stage i given that the firm has to make a decision, where N decisions are made during the simulation and n_i at the i^{th} stage.

The “Decision Making Entropy” for the system is now given by

$$E = \sum -p_i \ln(p_i)$$

The obvious extreme values calculated by this measure occur when:-

(a) decisions are made at only one stage, for example stage 1 when $p_1 = 1$, and $p_i = 0$ all other i in this case $E = 0$, and the decision

maker considers only this first stage, the simplest situation.

(b) decisions are made at all stages with the same frequency when

$n_i = k$, for all i and $N = mk$ in which case

$$p_i = 1/m,$$

and

$$E = \sum -1/m \ln(1/m) = \sum 1/m \ln(m)$$

giving $E = \ln(m)$ the practical worst case where decisions have to be made equally often at all stages.

Therefore Decision Making Entropy can be represented on a $[0,1]$ scale using the measure

$$DME = E / \ln(m) .$$

where $DME = 0$ implies that all decisions are made at one particular stage, in practice the first stage, and $DME = 1$ implies that decisions have to be made at all stages in the process with equal probabilities.

But because a DME of 1 can be obtained when k is small and hence N is small, no real problem few decisions to make and when k is large when many decisions have to be made this measure cannot be satisfactory when considered in isolation.

1.1 Simulation Results

To validate and evaluate the measures for planning and control complexity in small and larger firms the following manufacturing configurations were simulated. The results assumed that the dominant machine was the first machine, assuming strict dominance $T_i > T_j$ all $i > j$ where T_j is the average processing time on the j^{th} machine.

From each simulation the following results were collected

- (a) Mean μ
mean process time per job
- (b) s.d. σ
standard deviation of process time
- (c) s.d./mean σ/μ
- (d) Entropy E
- (e) Entropy/stage E/m
- (f) Mean/average {mean process time}/ {sum of average machine times}

The aim being to determine whether or not measures c and e provide alternative approaches to the estimation of system complexity.

1.1.1 n machines in series

Here it is assumed that there is one machine at each stage. The system was simulated with 1, 3, 5, and 10 production stages. The results from these simulations are summarised in the table 1. From these results it can be seen that there is a relationship between parameters c and e, both parameters decreasing as the number of stages increases.

1.1.2 n machines in parallel

Similar results were obtained when configurations with only one stage in the production process, but many processors at the stage, were simulated. The system was simulated with 1, 3, 5, and 10 processors at each stage. The results from these simulations are given in table 2. From these results it can be seen that there is a relationship between parameters c and e.

Stages	1	3	5	10
mean	20.2	48.0	58.5	102.
s.d.	15.2	20.2	20.3	21.0
s.d./mean	0.75	0.42	0.34	0.20
Entropy	0.35	0.69	0.55	0.74
Entropy/stage	0.35	0.23	0.11	0.07
Mean/average	2.52	2.02	1.50	1.36

Table 1: n machines in series

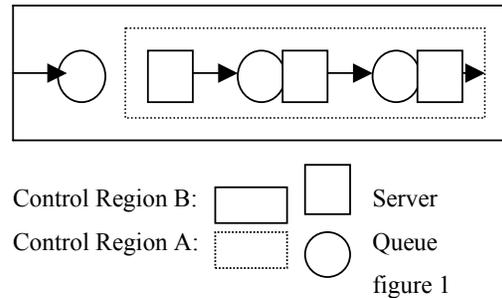
Processors	1	3	5	10
Mean	20.18	11.16	9.62	8.51
s.d.	15.28	5.10	3.18	1.77
s.d./mean	0.757	0.456	0.330	0.208
Entropy	0.353	0.367	0.362	0.341
E/m	0.353	0.122	0.072	0.034
Mean/ave	2.522	1.395	1.202	1.063

Table 2: n machines in Parallel

These results showing that the value given by σ/μ can be used to indicate the complexity of a manufacturing system, in place of the more complex Entropy value..

2 USING SIMULATION TO DESIGN CONSTANT WORK IN PROGRESS SYSTEMS

Consider the typical three stage small manufacturing firm (see figure 1) where jobs arrive at random and the job times at each processor are described by a rectangular distribution.



To be able to control the workload in region A, the manager needs to be able to determine the number of jobs, n, allowed into the system at any one time.

The number n being chosen to optimise the “cost” of the production system where cost can be expressed as a function of T and C,

$$cost = G(T, C)$$

where

- T is the total time for a job in the system, region B, and
- C measures the complexity in region A, which can be measured by the time a job spends in region A or from the entropy of region A, or more simply from the ratio σ/μ .

Three configurations were simulated, in each the final stage was the dominant stage, longest average job time at this stage,

- a) Three stages one processor at each stage,
- b) Three stages three processors at each stage,
- c) Three stages with five processors at the final stage.

Each of these can be considered to represent a small manufacturing firm. assuming that there is a single worker at each stage then in all cases the number of (production) workers is less than 10.

2.1 Three stages with one processor at each stage F(1,1,1)

This system was simulated with work in progress constraints of 1,2,3,4,5,and 6 jobs. The average time and variance of the average times in region A and B were calculated for each simulation, the results are given in table 3.

Region B				Region A		
wip	μ_B	σ_B^2	σ_B/μ_B	μ_A	σ_A^2	σ_A/μ_A
1	6489	M	0.59	29	4.36	0.07
2	889	M	0.57	29	4.63	0.07
3	52	459	0.41	33	10.79	0.10
4	51.9	449	0.41	39	38.52	0.16
5	51.9	449	0.41	44	102.8	0.22
6	51.9	449	0.41	47	189.1	0.29

Table 3: Comparing entropies

from these results it can be seen that there is no advantage to the firm from setting a limiting WIP_A value greater than 3.

Notice that for $WIP_A > 3$

- μ_B is constant, at 52
- σ_B is constant, at 449
- both the mean and the variance of the time in region A start to increase, the system start to become more complex

In this small firm the WIP limit is the same as the number of processors in the system (3) additional jobs remaining in the first queue.

2.2 Three stages F(3,3,3)

This system, representing a larger firm producing the same product type, was simulated with work in progress constraints of 6, 7, 8, 9, 10, 11, and 12 jobs. The average time and variance of the average times in region A and B were calculated for each simulation, the results are given in table 4.

For this firm the WIP it can be seen that the limiting value is “close to” the number of processors in the system, but not necessarily the same.

Region B				Region A		
wip	μ_B	σ_B^2	σ_B/μ_B	μ_A	σ_A^2	σ_A/μ_A
6	249	17967	0.54	28.4	4.22	0.07
7	51.39	309.2	0.34	28.7	4.49	0.07
8	34.46	53.41	0.21	29.2	5.91	0.08
9	32.5	31.5	0.17	29.9	8.04	0.09
10	32.29	29.02	0.17	30.8	12.4	0.11
12	32.26	28.5	0.17	31.9	21	0.14

Table 4: Comparing Entropies

2.3 Three stages F(1,1,5)

This system was simulated with work in progress constraints of 3, 4, 5, 6, 7, 8, 9, and 10 jobs. The average time and variance of the average times in region A and B were calculated for each simulation, the results are given in table 5.

In this firm the WIP limit is (again) not obvious and a means of combining the results from regions A and B, which might involve fuzzy logic for example, would be required to be able to select the optimal WIP value. However it can be stated that the maximum number of jobs allowed into region A at one time will be in excess of 6.

Region B				Region A		
wip	μ_B	σ_B^2	σ_B/μ_B	μ_A	σ_A^2	σ_A/μ_A
3	M	M	0.57	69.3	47.25	0.10
4	M	M	0.56	70.0	51.70	0.10
5	728	M	0.52	71.0	55.70	0.11
6	113	M	0.30	72.0	55.30	0.11
7	87.0	400	0.23	74.0	72.	0.11
8	83.0	282	0.20	77.0	93.	0.13
9	82.6	271	0.20	79.0	130.	0.14
10	82.6	271	0.20	80.9	170.	0.16

Table 5: Comparing Entropies

3 CONCLUSIONS

This investigation has shown that complexity measures can be employed to enable the design an optimal control system for a manufacturing firm and that simulation methods can be used to design this control system.

The results also show that as the firm grows and the production system becomes more complex, more processors at each stage the

parameters for the optimal control system become less obvious. This result emphasises the fact that as firms grow they will need to re evaluate their production control procedures if they wish to continue to operate optimally.

REFERENCES

Berry, S. and Murphy, W., 1998, Efficient planning and control in small manufacturing firms using white board systems, Proceedings 14th National Conference on Manufacturing Research XII. pp203-210.

Frizelle, G. and Woodcock, E., 1995, Measuring Complexity as an aid to Developing operational strategy, International Journal of Operations and Production Management, 15 (5), pp26-39.

Hopp, W.J. and Spearman, M.L., 1996, Factory Physics: Foundations of Manufacturing Management, Irwin.

Spearman, M.L., Woodruff, D.L. and Hopp, W.J, 1990, CONWIP: a pull alternative to Kanban, International Journal of Production Research, 28 (5), pp147-171.

FUZZY MODELLING APPLIED TO JOBSHOP SCHEDULING

V. LOWNDES*, J. M. CARTER⁺, M. H. WU* and S.BERRY*

*University of Derby

⁺Retired

Abstract: Fuzzy set theory allows the complexity of real-life issues to be included within the confines and rigours of the mathematical model. The authors have applied fuzzy methodology to the scheduling of jobs, the objective being the determination of an optimal sequence for dynamic job arrivals such that potentially conflicting priorities are satisfied. This paper concentrates on the theory on which the models are based, demonstrating an application by referring to a static problem.

Keywords: Scheduling, Fuzzy Logic, Jobshop.

1. INTRODUCTION

1.1 Scheduling

A scheduling problem can be considered to be an exercise in finding an appropriate timetable for the processing of jobs, by machines, such that some performance measure achieves its optimal value. Within this definition, it can be seen that there are two aspects to be considered concurrently, the satisfaction of constraints (e.g. availability of resources) and the optimisation of objectives (e.g. flow-times).

In general, such problems are known to be NP hard and probably as a consequence of this, scheduling has been an active area of research for many years. However, Pinedo [1995] notes, real-world scheduling problems are usually very different from the mathematical models studied by researchers in academia. Panwalker & Iskander [1977] also reported on the discrepancy between performance measures used by researchers and those preferred in industry. Actual firms place a higher priority on meeting due-dates than on typical research objectives such as minimising flow-time. (Gee & Smith [1993])

Woolsey [1982] also warns of the dangers inherent in failing to take a holistic view of production scheduling.

Pinedo lists a number of important requirements of real-manufacturing that are not normally met by OR models. An example of this is the existence of multiple objectives, i.e. there is not a single objective but multiple objectives to be optimised.

For example, given a jobshop with random job arrivals, which processes all jobs on a single machine, (the scheduler may need to consider the following goal:

Satisfy all due-dates, however certain jobs are for particularly important customers and it is a major priority to ensure that these jobs are completed on time.

1.2 The System

The authors have applied fuzzy methodology to the scheduling of jobs, the objective being the determination of an optimal sequence for dynamic job arrivals such that potentially conflicting priorities are satisfied. This paper concentrates on the theory on which the models are based, demonstrating an application by referring to a static problem. The main focus is on the study of a jobshop processing all jobs on a single machine. The difficulty in scheduling these jobs arises as a consequence of the existence of a multi-criteria objective, i.e. to meet all due-dates, whilst ensuring the satisfaction of the most significant customers.

The relevance of a fuzzy logic approach can be justified in the desire to optimise multiple objectives and so achieve a closer resemblance to the real-world. (Zadeh [1973], in his paper *Outline of a New Approach to the Analysis of Complex Systems & Decision Processes*, proposed that conventional quantitative techniques of system analysis are unsuited to dealing with humanistic systems.)

2. FUZZY SCHEDULING

2.1 Fuzzy modelling

Fuzzification

The first stage in producing a model, is to identify those linguistic variables to be included. It was decided that *due-date* and *customer priority* were the most significant factors, with *processing time* being of lesser importance.

Due-date

The actual allocation of due-dates was deemed to be outside the control of the scheduler. This is frequently the case in reality, due-dates frequently being given to customers by sales personnel without reference to the production

staff. Consequently in line with this practice, every job was allocated a due-date of 28 days from its arrival time.

The relevance of due-date to the scheduler was assumed to be in terms of ‘close’ and ‘distant’.

Hence given the universe $U = [-\infty, 28] \in \mathbf{Z}$, and the fuzzy sets $C = \text{CLOSE}$ and $D = \text{DISTANT}$, the membership functions can be defined as below.

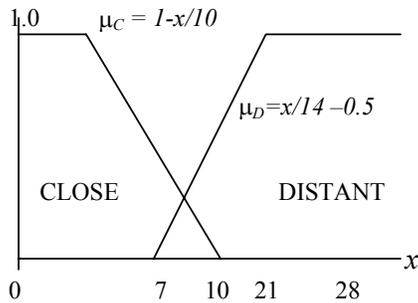


Fig. 1 Membership of CLOSE and DISTANT

Membership of CLOSE

$$\begin{aligned} \mu_C(x) &= 1.0 & x \leq 0 \\ \mu_C(x) &= 1.0 - x/10 & 0 < x < 10 \\ \mu_C(x) &= 0 & x \geq 10 \end{aligned}$$

Membership of DISTANT

$$\begin{aligned} \mu_D(x) &= 0 & x \leq 7 \\ \mu_D(x) &= x/14 - 0.5 & 7 < x < 21 \\ \mu_D(x) &= 1.0 & 21 \leq x \leq 28 \end{aligned}$$

The selection of a ‘trapezoidal’ form of membership function for ‘close’ is based on the assumption that the criticality of the closeness of an impending due date increases linearly with time up to the point at which the job becomes ‘late’. The ‘distant’ function represents a wish to avoid too early completion causing stock holding problems. The linear representation of increasing (and decreasing) closeness (and distance) has been selected, not only as a practical modelling assumption, but also as an appropriate one, in the absence of any established evidence of a need for a more complex (e.g. quadratic) form.

Customer Priority

The universe of discourse was deemed to be the set of ‘customer ratings’, {Bad, Low, Medium, High, Very Important}, with membership of the fuzzy set $CP = \text{CUSTOMER PRIORITY}$ taking the form:

μ_{CP}	0.0	0.2	0.5	0.75	1.0
Cp	bad	low	Med	High	Very Important

$$\begin{aligned} \mu_{CP}(\text{Bad}) &= 0.0 \\ \mu_{CP}(\text{Low}) &= 0.20 \\ \mu_{CP}(\text{Medium}) &= 0.50 \\ \mu_{CP}(\text{High}) &= 0.75 \\ \mu_{CP}(\text{Very Important}) &= 1.0 \end{aligned}$$

Processing Time

It is assumed that at the time of scheduling the exact processing times are unknown, (Hestermann & Wolber[1997]). However the scheduler can estimate a processing time as ‘short’, ‘medium’ or ‘long’. Thus the following membership functions are defined for the fuzzy sets SHORT, MEDIUM and LONG.

$$U = [0, 14] \in \mathbf{Z}$$

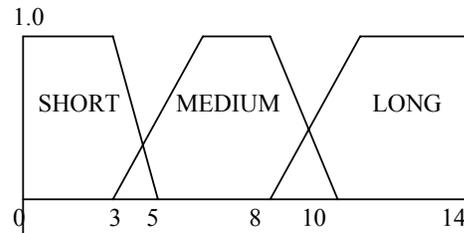


Fig. 2 Membership of SHORT, MEDIUM and LONG

Rule Evaluation

The fuzzy inputs of CLOSE, DISTANT and CUSTOMER PRIORITY are combined to produce an output which is a sequence priority. (Table 1)

Rule Matrix

Customer Priority \ Due-Date	Due-Date	
	Close	Distant
Bad (B)	Reject	Reject
Low (L)	Sequence quite high	Sequence very low
Medium (M)	Sequence high	Sequence low
High (H)	Sequence very high	Sequence quite low
Very important (VI)	Sequence extremely high	Sequence medium

Table 1 Summary of Sequencing Priorities

For example:

IF customer priority is Bad AND due-date is Close THEN Reject.

IF customer priority is Low AND due-date is Close THEN Sequence quite high.

IF customer priority is High AND due-date is Distant THEN Sequence quite low.

Sequencing Priority

Sequence -	
Extremely high	EH
Very high	VH
High	H
Quite high	QH
Medium	M
Quite low	QL
Low	L
Very low	VL
Reject	R

Table 2 Ordering of sequence priorities

If more than one job has the same priority at the head of the sequence, then a job with 'shortest' processing time will be selected for processing.

The general model

Composition or relational product:

Suppose $T = S \circ R$,

where $R \in F(X \times Z)$, $S \in F(Z \times Y)$.

$\forall (x, y) \in X \times Y$:

$$\mu_T(x, y) = \sup_{z \in \mu_2} \min \{ \mu_R(x, z), \mu_S(z, y) \} \quad [1]$$

Union

$$X = A \cup B \Leftrightarrow \forall x \in U$$

$$[\mu_X(x) = \mu_A(x) \vee \mu_B(x)]$$

$$= \forall x \in U [\mu_X(x) = \max \{ \mu_A(x), \mu_B(x) \}] \quad [2]$$

The fuzzy relation SP representing the sequencing priorities, is derived from an application of equation [1]

$$\mu_{SP}(c, d) = \sup_{cp \in CUSTPRI} \min \{ \mu_R(c, cp), \mu_S(cp, d) \} \quad [3]$$

where $R \in F(\text{CLOSE} \times \text{CUST-PRI})$

$S \in F(\text{CUST-PRI} \times \text{DISTANT})$

and $SP = S \circ R$.

2.2 Application

A hand-worked example will illustrate how the rule base enables a job to improve its sequencing priority as the due-date gets closer. Note, however that a job for a Bad customer will be rejected and not included in the sequencing schedule. The following example will illustrate the mechanics of the fuzzy algorithm. There are six jobs waiting to be processed, one of which is for a customer considered to be of 'medium' importance and two for 'very important' customers.

The example has been deliberately chosen to create problems for the scheduler in the light of conflicting priorities, i.e. of fulfilling all promised due-dates whilst ensuring the satisfaction of the most significant customers.

The due-dates range from 0 days for Job 1 (medium) to 28 days for Job 4 (very important).

The fuzzy values for 'customer priority', 'close' and 'distant' have been derived according to the definitions given in §2.1.

The sequencing priority is then determined by applying equation [3] in the form:

$$\mu_{SP}(c, d) = \min \{ \mu_c, \mu_{cp} \} \vee \min \{ \mu_{cp}, \mu_d \},$$

according to the rule matrix in Table 1.

Job	1	2	3	4	5	6
Due-date	0	10	6	28	26	14
Process time	5	1	8	6	2	4
Cust. Priority	M	L	V. I	V. I	H	L
Fuzzy cust-pri	0.5	0.2	1.0	1.0	0.75	0.2
Fuzzy dd-close	1.0	0.0	0.4	0.0	0.0	0.0
Fuzzy dd-dist	0.0	0.21	0.0	1.0	1.0	0.5
Max-min	else	dist	else	dist	dist	dist
Seqnce:	H	VL	EH	M	QL	VL

Table 3: Test example – Six jobs waiting to be processed.

Step 1

Consider Job 1:

Comparing the fuzzy value of 'customer priority' with 'close' and 'distant' –

$$\min \{ \mu_c, \mu_{cp} \} \vee \min \{ \mu_{cp}, \mu_d \}$$

$$\mu_{cp} = 0.5 \text{ (customer priority is Medium)}$$

$$\mu_c = 1.0 \text{ (membership of 'close')}$$

$$\mu_d = 0.0 \text{ (membership of 'distant')}$$

$$(\min \{ 1.0, 0.5 \} = 0.5) \vee (\min \{ 0.5, 0.0 \} = 0.0)$$

$$\max \{ 0.5, 0.0 \} = 0.5 \quad \text{'close'}$$

(Application of equation [2])

Thus: Medium and close =>

Sequence high; μ_{SP} (according to Table 1)

$$\text{Job 2: } \min \{ 0.0, 0.2 \} = 0.0$$

$$\min \{ 0.2, 0.21 \} = 0.2$$

$$\max \{ 0.0, 0.2 \} = 0.2 \quad \text{'distant'}$$

Thus: Low and Distant =>

Sequence very low

$$\text{Job 3: } \min \{ 0.4, 1.0 \} = 0.4$$

$$\min \{ 1.0, 0.0 \} = 0.0$$

max {0.4, 0.0} = 0.4 'close'
 Thus: Very Important and Close=>
 Sequence extremely high

Job 4: min {0.0, 1.0} = 0.0
 min {1.0, 1.0} = 1.0
 max {0.0, 1.0} = 1.0 'distant'
 Thus: Very Important and Distant =>
 Sequence medium

Job 5: min {0.0, 0.75} = 0.0
 min {0.75, 1.0} = 0.75
 max {0.0, 0.75} = 0.75 'distant'
 Thus: High and Distant =>
 Sequence quite low

Job 6: min {0.0, 0.2} = 0.0
 min {0.2, 0.5} = 0.2
 max {0.0, 0.2} = 0.2 'distant'
 Thus: Low and Distant =>
 Sequence very low

The sequencing priority is given by:
 < 3, 1, 4, 5, 2, 6 > Job 3 (the head of the
 sequence) is processed – duration 8 days..

Step 2

Step 2 will repeat all the tasks in Step 1, for the
 remaining five jobs.

All the due-dates are adjusted:
 due-date(new) = due-date(old) – process
 time(job 3)

Job	1	2	4	5	6
Due-date	-8	2	20	18	6
Process time	5	1	6	2	4
Cust. Priority	M	L	V. I	H	L
Fuzzy cust-pri	0.5	0.2	1.0	0.75	0.2
Fuzzy dd-close	1.0	0.8	0.0	0.0	0.4
Fuzzy dd-dist	0.0	0.0	0.93	0.79	0.0
Max-min	close	close	dist	dist	close
Sequence:	H	QH	M	QL	QL

Table 4: Test example – Five jobs in queue.

Job 1: min {1.0, 0.5} = 0.5
 min {0.5, 0.0} = 0.0
 max {0.5, 0.0} = 0.3 'close'
 Thus: Medium and Close =>
 Sequence high

Job 2: min {0.8, 0.2} = 0.2
 min {0.2, 0.0} = 0.0
 max {0.2, 0.0} = 0.2 'close'

Thus: Low and Close =>
 Sequence quite high

Job 4: min {0.0, 1.0} = 0.0
 min {1.0, 0.93} = 0.93
 max {0.0, 0.93} = 0.93 'distant'
 Thus: Very Important and Distant =>
 Sequence medium

Job 5: min {0.0, 0.75} = 0.0
 min {0.75, 0.79} = 0.75
 max {0.0, 0.75} = 0.75 'distant'
 Thus: High and Distant =>
 Sequence quite low

Job 6: min {0.4, 0.2} = 0.2
 min {0.2, 0.0} = 0.0
 max {0.2, 0.0} = 0.2 'close'
 Thus: Low and Close =>
 Sequence quite high

The current sequencing priority is given by:
 < 1, 2, 6, 4, 5 > thus Job 1 is
 processed – duration 5 days.

Step 3:

Job	2	4	5	6
Due-date	-3	15	13	1
Process time	1	6	2	4
Cust. Priority	Low	V. Imp	High	Low
Fuzzy cust-pri	0.2	1.0	0.75	0.2
Fuzzy dd-close	1.0	0.0	0.0	0.9
Fuzzy dd-dist	0.0	0.57	0.43	0.0
Max-min	close	Distant	distant	Close
Sequence:	Quite high	Medium	quite low	quite high

Table 5: Test example – Four jobs in queue.

Job 2: min {1.0, 0.2} = 0.2
 min {0.2, 0.0} = 0.0
 max {0.2, 0.0} = 0.2 'close'
 Thus: Low and Close =>
 Sequence quite high

Job 4: min {0.0, 1.0} = 0.0
 min {1.0, 0.57} = 0.57
 max {0.0, 0.57} = 0.57 'distant'
 Thus: Very Important and Distant =>
 Sequence medium

Job 5: min {0.0, 0.75} = 0.0
 min {0.75, 0.43} = 0.43
 max {0.0, 0.43} = 0.43 'distant'
 Thus: High and Distant =>
 Sequence quite low

Job 6: min {0.9, 0.2} = 0.2
 min {0.2, 0.0} = 0.0
 max {0.2, 0.0} = 0.2 'close'
 Thus: Low and Close =>
 Sequence quite high

The sequencing priority for the current jobs is now: $\langle 2, 6, 4, 5 \rangle$

Jobs 2 and 6 have the same sequencing priority, so the algorithm considers the *estimated* process time.

Job 2 would be classified as 'short'

$$(\mu_{\text{SHORT}}(j_2) = 1.0),$$

Job 4 has a probability of 0.5 of being estimated as 'short',

$$(\mu_{\text{SHORT}}(j_4) = 0.5, \mu_{\text{MED}}(j_4) = 0.5).$$

Job 2 would be chosen for processing – duration 1 day.

Step 4:

Job	4	5	6
Due-date	14	12	0
Process time	6	2	4
Cust. Priority	V. Imp	High	Low
Fuzzy cust-pri	1.0	0.75	0.2
Fuzzy dd-close	0.0	0.0	1.0
Fuzzy dd-dist	0.5	0.36	0.0
max-min	Distant	distant	close
Sequence:	Medium	quite low	quite high

Table 6: Test example – Three jobs in queue.

Job 4: $\min \{0.0, 1.0\} = 0.0$
 $\min \{1.0, 0.5\} = 0.5$
 $\max \{0.0, 0.5\} = 0.5$ 'distant'

Thus: Very Important and Distant =>
 Sequence medium

Job 5: $\min \{0.0, 0.75\} = 0.0$
 $\min \{0.75, 0.36\} = 0.36$
 $\max \{0.0, 0.36\} = 0.36$ 'distant'

Thus: High and Distant =>
 Sequence quite low

Job 6: $\min \{1.0, 0.2\} = 0.2$
 $\min \{0.2, 0.0\} = 0.0$
 $\max \{0.2, 0.0\} = 0.2$ 'close'

Thus: Low and Close =>
 Sequence quite high

The sequencing priority is given by:

$\langle 6, 4, 5 \rangle$

so Job 6 is processed, - duration 4 days.

Step 5: (see table 7)

Job 4: $\min \{0.0, 1.0\} = 0.0$
 $\min \{1.0, 0.21\} = 0.21$
 $\max \{0.0, 0.21\} = 0.21$ 'distant'

Thus: Very Important and Distant =>
 Sequence medium

Job 5: $\min \{0.2, 0.75\} = 0.2$
 $\min \{0.75, 0.07\} = 0.07$
 $\max \{0.2, 0.07\} = 0.2$ 'close'

Thus: High and Close =>
 Sequence very high

Job	4	5
Due-date	10	8
Process time	6	2
Cust. Priority	V. Imp	High
Fuzzy cust-pri	1.0	0.75
Fuzzy dd-close	0.0	0.2
Fuzzy dd-dist	0.21	0.07
max-min	Distant	close
Sequence:	Medium	very high

Table 7: Test example – Two jobs in queue.

Job 4: $\min \{0.0, 1.0\} = 0.0$
 $\min \{1.0, 0.21\} = 0.21$
 $\max \{0.0, 0.21\} = 0.21$ 'distant'

Thus: Very Important and Distant =>
 Sequence medium

Job 5: $\min \{0.2, 0.75\} = 0.2$
 $\min \{0.75, 0.07\} = 0.07$
 $\max \{0.2, 0.07\} = 0.2$ 'close'

Thus: High and Close =>
 Sequence very high

This gives the final sequencing priority:

$\langle 5, 4 \rangle$

Job 5 is processed, - duration 2 days.

Thus the complete schedule is defined as:

$\langle 3, 1, 2, 6, 5, 4 \rangle$

and is summarised in the following table:

Job	3	1	2	6	5	4
Cust. Priority	V. I	M	L	L	H	V. I
Due-date	6	0	10	14	26	28
Process time	8	5	1	4	6	2
Start time	0	8	13	14	18	20
Completion time	8	13	14	18	20	26
Lateness	2	13	4	4	-6	-2

Table 8: Test example – Final schedule.

The job completion times for the most important customers (V. Imp and High) are satisfactory.

The main cause for concern, at first glance, is the 13 day lateness attributed to the 'medium'

rated customer. However closer scrutiny reveals that this was unavoidable, the algorithm correctly gave priority to the 'very important' job.

The dynamic process can be summarised by considering the sequence priority each time the machine becomes available:

- <3,1,4,5,2,6> Job 3 processed
- <1,2,6,4,5> Job 1 processed
- <2,6,4,5> Job 2 processed
- <6,4,5> Job 6 processed
- <5,4> Job 5 processed

2.3 Further Development

A model for fuzzy decision making which considers conflicting scheduling priorities, has been described. Further enhancement/refinement could be incorporated. For example, additional fuzzy variables associated with 'earliness' and 'lateness' could be considered for inclusion in the algorithm, in order to allow consideration of stock-holding costs to be included in the model.

An increase in demand naturally leads on to consideration of the use of two or more machines. A second model for a multi-machine problem considered the availability of two machines with the following properties:

Machine A:

Cheap to run, but incurs longer process times.

Machine B:

Expensive to run but incurs shorter process times.

At times of light or normal demand, jobs would be processed on Machine A, the alternative action, *process job on Machine B*, could be triggered as heavier demand causes queue build-up.

A typical inference rule would be:

- IF Number of jobs in queue is *heavy*
- OR Number of jobs with sequence priority \geq medium is *normal*
- THEN Action

The antecedent of the rule is represented by an application of *union*, equation [2].

The consequent of the rule would be:

Action

Process head of sequence on Machine B

The universe of discourse is N.

Membership of the fuzzy subsets *light*, *normal* and *heavy* is defined according to Figure 3:

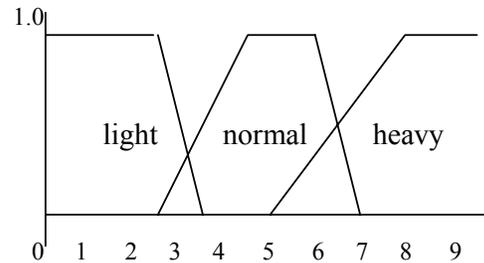


Fig 3 Fuzzy sets associated with queue state

Examples

1. Suppose there are 3 jobs currently in the queue, all have customer priority rating of 'medium' or above, then no action.
2. Suppose there are 5 jobs in the queue, 4 of which have customer importance rated as 'medium' or higher, then action.
3. Suppose there are 7 jobs in the queue, only 3 jobs with customer importance rating of 'medium' or higher, then action.

3. CONCLUSIONS

Fuzzy set theory allows the complexity of real-life issues to be included within the confines and rigours of the mathematical model. In this paper, a theoretical model has been presented which demonstrates how fuzzy decision making can support the dynamic scheduling process, enabling the conflicting priorities of multi-objectives to be managed effectively in polynomial time.

Bibliography

- Bandemer H., Gottwald S., (1996) *Fuzzy Sets, Fuzzy Logic, Fuzzy Methods*. Wiley.
- Gee E. S., Smith C. H., (1993) *Selecting Allowances for Jobshop Performance*. Int. J. Prod. Research, Vol. 31, No. 8, p1839-1852.
- Panwalkar S.S, Iskander W., (1977) *A Survey of Scheduling Rules*. Operations Research, 25 p45-61.
- Pinedo M., (1995) *Scheduling: Theory, Algorithms and Systems*. Prentice Hall.
- Woolsey R. E. D.,(1982) *The Fifth Column: Production Scheduling as it really is*. Interfaces 12(6),p115-118. Inst. of Man. Science.
- Zadeh L. A., (1973) *Outline of a New Approach to the Analysis of Complex Systems & Decision Processes*. IEEE Trans. Syst., Man, Cybern. , Vol. SMC-3 No. 1.

DIME-II: A COMPUTING FRAMEWORK FOR TRAFFIC SYSTEMS

MOHAMED KHALIL and EVTIM PEYTCHEV

*School of Computing and Mathematics, The Nottingham Trent University,
Burton Street, Nottingham, NG1 4BU, UK*

mohamed.khalil@ntu.ac.uk, evtim.peytchev@ntu.ac.uk

Abstract: Building a successful distributed shared memory system depends enormously on the degree of consideration of certain design issues in the designing stage. This degree varies according to the nature of the distributed application itself. DIME-II, an extension of DIME-I system, is designed with features specific for traffic control distributed systems taken into account. The paper presents implementation of this system considering number of common design issues and inheriting some features from the implementation of DIME-I.

Key words: DIME, Granularity, Scalability, Heterogeneity, Distributed Computing.

1. INTRODUCTION

Over the last years, distributed shared memory (DSM) paradigm has attracted researchers who have investigated different approaches that hide remote communication mechanism from the programmers on a cluster of workstations, where each workstation has computing power comparable to the mini-mainframe in the past.

Many of Distributed shared memory (DSM) algorithms have been successfully implemented in a wide range of experimental and commercial applications. Building an efficient, successful software distributed shared memory system depends mostly on the application that implements the DSM algorithm. However, there are number of requirements or designing issues which influence the performance and the efficiency of the system, as presented in [Nitzberg B. et al, 1991]. The level of satisfying these issues varies from one application to another. Therefore, considering the nature of an application in the designing stage can effectively increase the performance of that application. These design issues are: structure and granularity; scalability; heterogeneity; and memory consistency.

The **D**istributed **M**emory **E**nvironment (DIME) [Argile A. et al, 1999] is a software DSM system that provides an interface between distributed software modules that execute on networked workstations. DIME has been designed specifically to support vast range of transport telematics applications and it offers a convenient interface to the applications programmer. The first implementation of DIME system is called DIME-I. As it was built as a user-level software DSM system, DIME-I provides an easy to use communication interface that simply and reliably delivers data and messages to all nodes in the system. In [Khalil M. et al, 2003] a revised

framework of DIME-I was introduced in order to improve the performance of DIME system mainly by avoiding its limitations and minimizing the time of data retrieval from the viewpoint of user application. This new framework is called DIME-II. This paper presents an implementation for DIME-II.

The presented implementation chooses to continue using user-level implementation for DIME-II software DSM system, as it does not require changes in the lower levels of the system (compiler and operating system). Besides, such implementation provides good portability in distributed systems. There are some pre-existing requirements that have been taken into account in the designing stage of DIME-II. For example, the implementation of DIME-II, as in DIME-I, supports two types of data structures that naturally exist in urban traffic information and control system. Other inherited properties from the DIME-I system are the granularity and the non-locking approach.

2. TYPES OF DATA IN TRAFFIC CONTROL SYSTEM

Building a successful DSM system requires a detailed knowledge of the data transactions in the system; therefore a special consideration of data flows in a traffic control system was taken in the design stage of DIME-I [Peytchev E., 1999]. In a traffic control system there are two kinds of data can be recognized:

-Dynamic data: It is collected by the real-time traffic control system. It contains all information about traffic counts and local controls as they occur in the traffic network. It is characterized by its high volume - in excess of 120 Mbytes per day per one specific type of message. Besides, this

kind of data is updated in a high frequency rate (per second basis).

- Static data: This kind of data is updated in a much longer period of time and its purpose (in general) is to make the results from traffic modules available for reading by the other functional modules in the system.

3. DESIGNING ISSUES

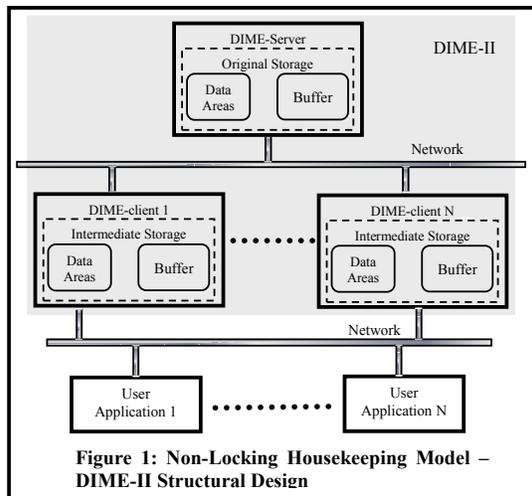
In the designing stage of building DIME-II numbers of design issues have been considered. In the traffic system there are two types of data structures: dynamic and static data. The data representing the dynamic data type is in fact a constant flow of uniform messages issued from the traffic system. To accommodate this type of data, the system needs a formation capable of accepting a number of uniform structures at a time, and at the same time keeping the most recent data only. The implementation of DIME-I utilizes circular buffer for this type of data, where each element of the buffer is a user defined message structure and its size depends on the size of the urban traffic network. On the other hand, the relatively static data in the traffic system usually reflects the value of some internal variables in the traffic modules. The volume and format of this data is usually module dependent since each module has its own internal representation of the traffic. Therefore, DIME-I utilizes an array of bytes of user's defined size for static data, as it's the most suitable choice [Peytchev E., 1999]. The implementation presented in this paper continues using this granularity, since it is suitable and convenient for representing the two types of data of traffic control systems.

Moreover, DIME-II is designed to run on different platforms, and therefore, it can run in a heterogeneous environment. However, DIME-II employs communication algorithm that internally exchanges data and messages as bytes; therefore, software modules have to make their own internal simple conversions. DIME-II is designed with capability of extension to contain further addition of software modules. As described in [Nitzberg B. et al, 1991] there are two factors that can greatly limit the scalability of distributed shared memory systems. These factors are: general common knowledge and central bottleneck. In DIME-II these factors have been overcome. The framework of DIME-II supports the presence of data replicas at intermediate memories each in a location near to certain application. Since each application in DIME-II performs its operation on an intermediate memory with no competition with other applications [Khalil M. et al, 2003], therefore; this housekeeping algorithm can reduce the contention on the central shared memory. Consequently, it can

decrease the likelihood of central bottleneck.

4. DIME-II STRUCTURAL DESIGN

As introduced in [Khalil M. et al, 2003], the architecture of DIME-II system employs non-locking approach and consists of three layers as depicted in Figure 1. In the first layer DIME-server takes control over the original shared memory system, and has the role of monitoring any modification in the central memory in order to keep all intermediate memories throughout the system informed with the updates. In the second layer DIME-client controls accesses to certain intermediate memory that is associated with only one user application. An intermediate memory holds copy of part of the original shared memory which is required by the associated traffic module.



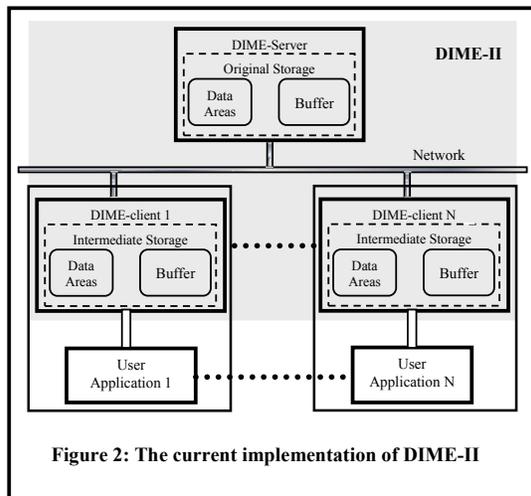
DIME-client's task is to communicate with DIME-server on behalf of its user application to perform write operations, and at the same time it looks up in the intermediate memory to retrieve certain data for the user application. In the third layer there are user applications, which are traffic control system modules. A user application performs its operations on the local memory, leaving the time delay burden of contacting the server to DIME-client for making the intermediate memory up to date, and reflecting the update in the original memory. This saves valuable time for user applications - usually wasted in network communications- to perform its native tasks.

Since this model supports the presence of data replicas, special care has been taken to avoid data inconsistency in this architecture. The consistency model presented in [Khalil M. et al, 2003] intends to maintain systemwide consistent view of the memory in terms of data area and buffer structures. This model is designed specifically to support the

two types of data structures that comprise the shared memory, and it has a flavour of sequential consistency model [Lamport L., 1979] as it is the most intuitive definition for programmers. In other words, it's a relaxed definition of sequential consistency model. Unlike sequential consistency, the consistency definition is advantageous in such a way that it supports multi-reading/multi-writing.

5. AN IMPLEMENTATION FOR DIME-II

The first decision has to be taken in this stage is where to locate the process of DIME-client in the framework. This implementation chooses to place DIME-client process at the same machine as its associated user application in order to achieve the sought goals. Therefore, slight modification to the framework in figure 1 has to be made and is illustrated in figure 2.



For implementing this framework, two separate executables have been coded for DIME-server and DIME-client. Both make use of the Java programming language.

Java's multithreaded support is essential for the successful programming of the DIME-II software. The presented implementation exploits the potential of multithreading as it has shown improved performance in DSM systems by hiding the long communication latencies typically associated with software DSM systems [Speight E. et. al, 1997] [Mueller F., 1997].

The produced software is described comprehensively in the following subsections in terms of DIME-server and DIME-client.

5.1. DIME-server

DIME-server executes command packets

(`cmmnd_pkt`) in the order they are received (not the order they were sent). In accordance with the atomicity of DIME-II system, each command is performed as an indivisible operation. In other words, there is no interleaving when DIME-server is performing a command.

DIME-server keeps a list of every created shared item and a list for each item in the list, which contains names of user applications that currently have replica of it. When it receives a request for creating shared item, DIME-server allocates space for the item in the DSM only if it has not been created before. Otherwise, the name of the requesting application is just added to the list of the applications that have the replica of that shared item. In the case of write operations; it sends updates only to the applications that have replica of the updated value using the relative list of applications. In other words, unlike BDSM [Auld P. et al, 2000], DIME-II system employs a multicast-based algorithm to disseminate updates to the application that are involved in the write operation. On the other hand, when it receives a request for the deletion of a shared item, DIME-server deletes the name of the application from the list of that item. The item is removed permanently only if the requesting application is the last one in the list.

In order to improve the performance of DIME-server, numbers of threads are used. Each application is serviced by separate thread that listens to its requests and inserts them in a queue of command packets. This thread is called *client-service* and is created when DIME-server receives request from a user application to use the shared memory. The thread of client-service consists of two other threads: 1. *ListenToPacket* that continuously listens to command packets sent from its user application and inserts them in a queue to be processed later, 2. *SendPacket* thread that keeps checking another queue of command packets ready to be sent to the user application. All client-service threads -particularly ListenToPacket threads- insert every received command packet in one single queue. This queue is processed by another thread called *sequencer*. The sequencer is the only thread that can perform operations on the shared memory in DIME-server. After processing an operation, the sequencer passes an appropriate command packet to certain client-services, which in turn send the command to certain applications – particularly done by SendPacket thread.

Employing several threads allows dividing the task of the DIME-server into a number of sub-tasks to be executed at the same time, enhancing the functionality of the DIME-server.

The speed of the sequencer performing commands in the shared memory (read & write) is much higher than the speed of the underlying network and therefore this is not a cause for bottleneck problems. Figure 3 illustrates a general view of DIME-server.

In addition to the main task of controlling the shared memory, DIME-server holds number of permission tables. These tables prescribe levels of

so, otherwise an error message is sent to it. For write operation, DIME-client can apply the update locally in the case of area writing, and then the update is sent to the DIME-server.

At DIME-client side, there is a thread that continually listens to messages from DIME-server and acknowledges them. The main task of this thread is to receive new updates from the DIME-server and then update the intermediate memory

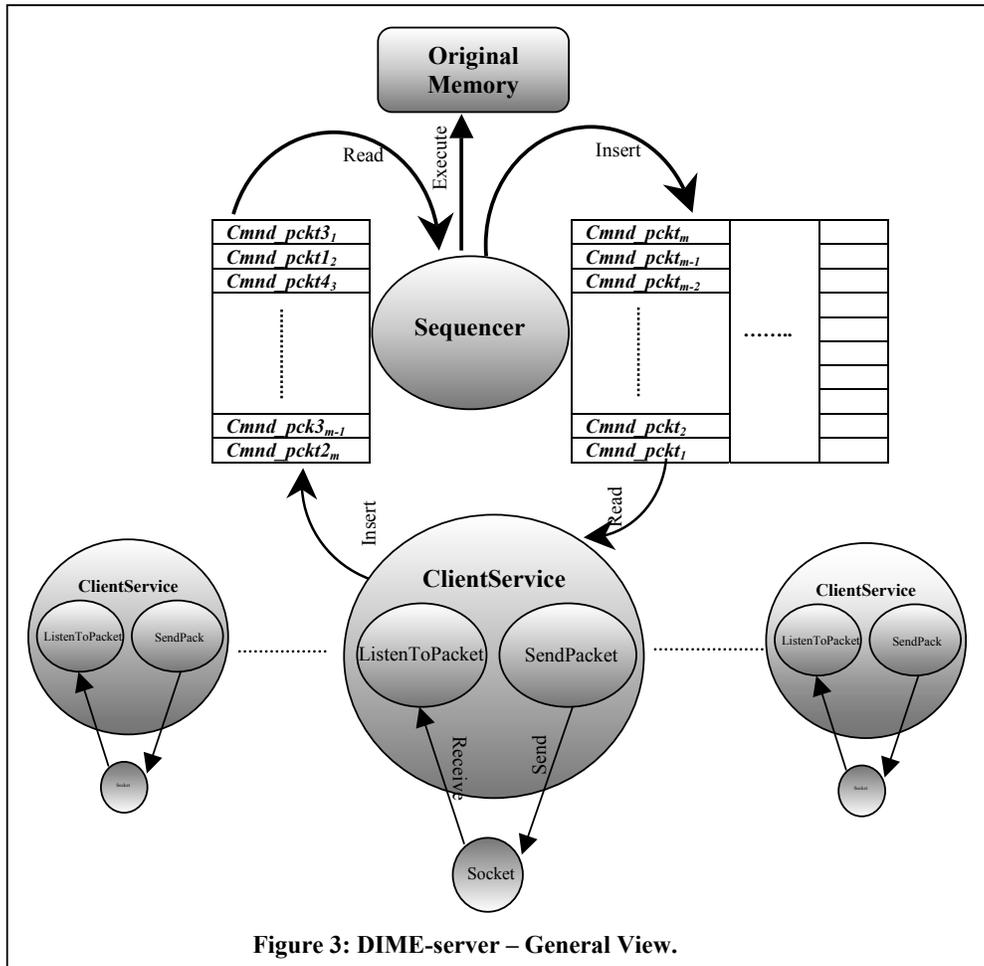


Figure 3: DIME-server – General View.

access that each user application has on the shared memory. The level of access is either no access, read only or read/write. DIME-server disseminates certain permission table to a DIME-client upon initiating a process of traffic module. This permission table is used by DIME-client upon performing any operation on the shared memory.

5.2. DIME-client

DIME-client keeps a copy of the permission table of its user application. It checks the privilege of its user application upon performing any operation in the intermediate memory. An operation is processed only if the application is permitted to do

accordingly. Thereby, DIME-client can guarantee the consistency of the local replicas of the shared memory. On the other hand, any read operation can be performed directly on the available local memory without need to contact the main shared memory controlled by the DIME-server over the network. This locality of reference is advantageous in saving network bandwidth, and reducing time of data retrieval for user application.

5.3. User's Interface of DIME-II Software

So far, the produced system provides number of

functions for performing different operations on the distributed computers shared memory. These functions are:

- Initializing intermediate shared memory. User application calls this function to get permission for initializing the shared memory and start using the DSM. If the application has permission to use the system, a permit will be sent along with permission table. This permission table contains names of shared items the application is permitted to use, and the access privileges for each item.
- Creating data area/buffer. This function is called to create new shared memory item. A shared memory item is created only if the creating user application is permitted to use it.
- Writing in area/buffer. A user application invokes this function to update an existed shared item. This operation is performed only if the user application has enough permission to write in that item.
- Reading from area/buffer. The required data are sent to the requesting application along with the number of the values. Zero is sent if the requested item is empty. This operation is performed locally.
- Destroying area/buffer. Performing such operation results in removing certain shared item from the intermediate memory of the requesting application.

6. RELATED WORKS

TreadMarks [Amza C. et al, 1996] is a software DSM system where messages and data traffic is reduced by relaxing consistency semantics of the shared memory. TreadMarks is a user-level implementation of DSM relies on UNIX standard libraries in order to accomplish remote process communication, and memory management, therefore no need to make modifications on the operating system kernel. In [Lu H. et al, 1995] experimental results have shown that the separation of synchronization and data transfer and the request-response nature of data communication are responsible for lower performance comparing with PVM message-passing model. In DIME-II, there is no need for synchronization mechanism for a user application to have an exclusive access to the shared memory, since each user application is associated with an intermediate memory where all its operations are performed.

BDSM [Auld P. et al, 2000] is a broadcast-based, fully replicated software distributed shared memory system. Similar to our framework, each user process has an associated DSM subsystem that manages the shared memory, however, each user process has a complete copy of the shared memory where it processes all reads and writes locally.

Also, unlike our system, all writes to memory modify the local copy and arrange to broadcast the updated values to all the other processes. Another major difference with the presented framework is that BDSM allows one user process to be executed on a workstation.

Brazos system [Speight E. et al, 1997] utilizes multithreading at both the user level and system level. Multiple user-level threads allow applications to take advantage of symmetric multiprocessor servers by using all available processors for computation. In the runtime system there are two main threads. One thread is responsible for quickly responding to asynchronous requests data from other processes and runs at the highest possible priority. The other thread handles replies to requests previously sent by the process. This multithreaded aspect of Brazos allows greater amount of computation to communication overlap. The use of separate thread to handle incoming replies allows the system to maintain multiple simultaneous outstanding network requests, which can significantly improve performance. Additionally important, the exploitation of multithreaded DSM algorithms proved significant in hiding the communication latencies [Muller F., 1997].

7. CONCLUSION

This paper presents description of an implementation of the framework of DIME-II. This implementation is believed to improve the performance of the whole system in many aspects: saving network resources, reducing data retrieval from user application viewpoint, performing number of tasks per-node simultaneously, and at the same time maintaining consistency by a simple straightforward model. Currently number of experiments are set-up and under way in order to evaluate the performance of the DIME-II system in comparison with the current DIME-I system

8. REFERENCES

- [1] Argile A., Peytchev E., Bargiela A., Kossonen I., "Dime: A Shared Memory Environment for Distributed Simulation, Monitoring and Control Of Urban Traffic", 8th European Simulation Symposium, Genoa, Italy, ISBN 1-565555-099-4, Vol.1, pp. 152-156.
- [2] Peytchev E., "Integrative Framework for Discrete Systems Simulation and Monitoring", Ph.D. thesis, Department of Computing, The Nottingham Trent University, Nottingham, England. Feb. 1999.
- [3] Nitzberg B., Lo V., "Distributed shared memory: A survey of issues and algorithms", IEEE Computer, vol. 24, pp. 52--60, Aug. 1991.

- [4] Khalil M., Peytchev E., "Traffic Telematic Computing Framework based on Non-Locking and Housekeeping Distributed Shared Memory Algorithm", Sixth United Kingdom Simulation Society Conference (UKSim 2003), Apr. 2003, Emmanuel college, Cambridge, UK.
- [5] Lamport L., "How to make a Multiprocessor Computer that correctly executes Multiprocessor Programs". IEEE Trans. Comp., vol. C-29, no. 9, pp. 690-691, Sept. 1979.
- [6] Auld P., Kearns P., "Broadcast Distributed Shared Memory", Proceedings of the ICSA 13th International Conference on Parallel and Distributed Computing Systems, ICSA, pp., 2000.
- [7] Speight E., Bennett J., "Brazos: A Third Generation DSM System", In Proceedings of the 1st USENIX Windows NT Symposium, pp. 95-106, August 1997.
- [8] Mueller F., "Distributed Shared-Memory Threads: DSM-Threads", Workshop on Run-Time systems for Parallel Programming, Apr 1997.
- [9] Amza C., Cox A., Dwarkadas S., Keleher P., Honghui L., Rajamony R., Weimin Y., Zwaenepoel W., "TreadMarks: Shared Memory Computing on Networks of Workstations". Computer, vol.29, no.2, Feb. 1996, pp.18-28. Publisher: IEEE Comput. Soc, USA.
- [10] Lu H., Dwarkadas S, Cox AL, Zwaenepoel W. "Message Passing versus Distributed Shared Memory on Networks of Workstations". Proceedings of the 1995 ACM/IEEE Supercomputing Conference (IEEE Cat. No.95CB35990). ACM. Part vol.1, 1995, pp.865-906 vol.1. New York, NY, USA.



Mr. Mohamed Khalil is a Research Student at the School of Computing and Mathematics, the Nottingham Trent University. He was graduated from Faculty of Mathematical Sciences, University of Khartoum, Sudan

with a bachelor degree (honour) in computer sciences. He worked in the Department of Computer Sciences, Faculty of Mathematical Sciences, University of Khartoum as a teaching assistant for nearly three years. He won two university prizes for the best academic performance while he was student in the academic years 93/1994 and 96/1997. Mr. Khalil started his PhD in August 2001 and the title of his thesis is "Integrative Monitoring and Control Framework Based on Software Distributed Shared Memory Non-Locking Model" under the supervision of Dr. Evtim Peytchev and Prof. Andrzej Bargiela. His research interests are: Distributed shared memory algorithms and prototyping, Distributed Shared memory Applications, and Traffic Telematics Systems.



Dr. Evtim Peytchev is a Senior Lecturer at the School of Computing and Mathematics, the Nottingham Trent University and has been a member of Intelligent Simulation and Modelling group for 10 years. Most of the recent research work in the group, dealing with the traffic control telematics, has been carried out by Dr. Peytchev under the supervision and leadership of the head of the RTTS group Prof. Andrzej Bargiela. As a result of the research work Dr. E. Peytchev has successfully presented his Ph.D. work entitled "Integrative Framework for Discrete Systems Simulation and Monitoring". He worked as a researcher for the successful conclusion of an EPSRC project "Integrative framework for the predictive evaluation of traffic control strategies" (GR/K16593) and most of his publications reflect the work under this project. Dr. Peytchev's interests span: traffics simulation modelling, traffic Telematics, mathematical modelling of the uncertainties in traffic, distributed computing environments, shared memory design, Telematics technology application in the urban traffic control. He is involved in International collaboration with the Transportation Systems Laboratory at the Helsinki University of Technology (Dr. I. Kosonen) and in the DTI funded 'Traffimatics' project.

LATE PAPERS

INTELLIGENT SYSTEM DESIGN FOR KNOWLEDGE STRUCTURE MODELS FROM OBSERVED DATA

VLADIMIR STEPASHKO¹, TATIANA ZVORYGINA²

*International Research and Training UNESCO Center of Information Technologies and Systems, Ukraine, Kiev,
Academic Glushkov Prospect, 40, 03680.*

¹ *Professor, Head of Department, E-mail: step@g.com.ua*

² *Master of Computer Science, Post-graduate Student, E-mail: zvortf@ua.fm*

Abstract: The task of modeling from data observed is considered as a sequence of stages, or subtasks, of solutions choosing. It is shown that at each stage there is some finite subset of possible solutions depending from those accepted on the previous stages. Each of accepted solutions, therefore, restricts subsets of possible solutions at all consequent stages. It allows to organize an "intellectual interlayer" between a user who needs to model something and a modeling software. The intellectuality level of a system of such kind is determined by implementation of knowledge (both theoretical and practical) about a process being modeled as well as methods of modeling and by minimization of requirements to skills of a user.

Keywords: knowledge structuring, modeling, data observed, decision making, dialog shell, intellectual system.

1. INTRODUCTION

There are known modeling tasks to which one or another method of analysis is being applied in practice, based on which the experts consider to have as the most adequate method for the given purposes. However, for the majority of real problems it is not possible to specify in advance the exact line-up of operations, as there is no a priori information on a plant or process being modeled. Most importantly, people who need, for example, to predict some economic or ecological indexes are not experts in the modeling field.

The modeling software existing in the market, for example "Statistics", have one but very essential, in our opinion, shortcoming, they are mainly oriented on users possessing high qualification in the modeling field. An expert in the field can only tell which a method of parameter estimation, or a generator of model structures, or a criterion of model selection should be preferred. An economist or ecologist is forced to choose methods "at random" and then manually check the obtained models with respect to their correspondence to the purposes of research.

Therefore there is a necessity for creation of some "intellectual interlayer" between a user who needs to model something and modern computational software. This "intellectual interlayer" should be capable of advising the user in a dialog mode which method may be better.

It is our aim to create such an intellectual interlayer. The first problem we are faced with is one of classification of the available knowledge in the modeling field. The main part of such expertise exists in the form of practical experience on applying one or another method to specific problems as well as the limitations on their application.

By a method of modeling we mean a set of operations with a given data sample allowing one to build a mathematical relationship between the output variable and the input variables.

2. KNOWLEDGE CLASSIFICATION

Let's assume that the data sample do not contain missing values of variables and are prepared for handling. There are quite well defined mathematical methods for data pre-processing [Duke, Samoilenko, 2001], so we shall not discuss them here.

We claim that each method of modeling contains, in explicit or implicit form, such key elements as a model class, an external criterion of model selection, a generator of model structures and a method of model parameters estimation. For example, it is easy to see that in the classical regression analysis, polynomial functions form the class of used models, inclusion or/and exclusion method is to be a generator of model structures, the

least squares method (LSM) is used as the method of the model parameter estimation, and the Fisher criterion is one for model selection. If we try to identify similar components in the Akaike method, we shall get, accordingly: ARIMA is the class of used models, embedded structures are to be a generator of model structures, the Yool-Walker method (YWM) is used as the method of the model parameter estimation, and the Akaike criterion is one for model selection.

It was also appointed that a choice of a modeling method is affected by such circumstances as the purpose of investigations and the type of plant being modeled.

3. SUB-PROBLEMS

Therefore, in solving the problem of a relevant modeling method choice we define the following sub-problems to be solved sequentially:

1. Choice of the modeling purpose (approximation, interpolation, extrapolation, trend definition, prediction, construction of input-output model etc.)
2. Definition of the plant type (linear static, nonlinear static, linear time series, nonlinear time series, linear dynamic, and nonlinear dynamic)
3. Definition of process stationarity (stationary, with an increasing trend, with a decreasing trend, with an oscillatory trend, with the mixed trend)
4. Choice of a model class (linear regression, autoregression, autoregression with trend, harmonic, logarithmic, polynomial or exponential functions of time, difference equations etc.)
5. Choice of external criterion of model selection (Akaike criterion, "jack-knife", C_p -statistics of Mallows, unbiasedness and/or regularity criterion, Fisher criterion etc.)
6. Choice of a parameter estimation method (LSM, LMM, ridge regression etc.)
7. Choice of structure generation method (a given structure, embedded structures, inclusion, exclusion, inclusion-exclusion, exhaustive search, branches and bounds, combinatorial, combinatorial-selective, multilayer (GMDH)).
8. The obtained model validation (Fisher statistics, precision on control sample, etc.)

We claim that for the solution of problem of choosing a modeling method addressing to this set of subproblems is necessary and sufficient.

This order for solving the problem is well motivated. It was observed that the solution of the first subtask leads to the essential diminishing of the set of subsequent solutions. This happens because the decision making on each of stages introduces implicit limitations on application of these or other techniques the need in which to be decided at the consequent stages.

4. SOLUTION TECHNIQUES

After having determined the order of solving the subtasks, we have encountered the problem of finding such mathematical and dialogue procedures that would facilitate solving the formulated subtasks by an inexperienced (in the modeling field) user. To solve this problem we have used such modern techniques as Data mining [Duke, Samoilenko, 2001; Stepashko, 1991] and knowledge elicitation [Gavrilova, Khoroshevsky, 2000]. By combining these techniques with the dialogue interview of the user about the modeled plant, we have found a solution tree for the problem of observed data modeling. Even to a user at the first time facing the problem of modeling, this procedure allows to construct a more or less acceptable mathematical model.

It is better to give an example to illustrate our findings (see figure 1)

The four stages of decision making chosen for illustration are: definition of a process type, linearity, stationarity, and choice of the model class.

The first stage is choice of process type from the three indicated alternatives. If the user is not able to make a choice on his own, the dialogue and control tools help him. The principle of organization of the dialogue at the stage of deciding the process stationarity is described in detail in [Stepashko, Zvorygina, 2001].

It is easy to see from figure 1 that an investigator has very large set of model classes for choosing at the fourth stage (in the figure, we show only an incomplete class of models). If one attempts to solve the problem by the "brute force", each of possible models needs to be tested, what, in practice, is a sufficiently labour-consuming process. However, if one takes the advantage of decomposing the problem into the proposed subtasks, then, depending on the decision accepted at the first three stages, the set of allowed solutions will be significantly narrowed. For instance, if at

the early stages the decision is taken that the plant is linear and static, a unique solution is to use the linear regression model. If, however, the plant is linear and dynamic, the difference equation models can only be applied. In the case of linear time series without a trend, the model of autoregression is applied. If the trend is increasing or decreasing, it is possible to use a model of a linear trend. And in the case of an oscillatory trend, the choice is limited to a harmonic series or autoregression.

If at the first stage the decision is taken that the process is a "time series", at the second stage that it is non-linear, and at the third stage that it has an increasing trend (non-linear time series), then, at the stage of choosing the basis functions, the intellectual system will recommend the investigator to model the process in auto regression with trend model class. The user will also be given a choice of exponential and logarithmic functions of time. On the other hand, a whole class of models will be eliminated, such as linear regression model, or a model with a linear trend, etc. The considered variant is shown in figure 1 by shadow.

Note that the proposed organization of the intellectual envelope considerably simplifies the checking of correctness and consistency of the accepted decisions.

5. DISCUSSIONS AND CONCLUSIONS

The proposed principle of the intellectual envelope for a computer-aided data modeling system will have the following major advantages: interactive component at all stages of modeling; minimization of requirements to the user qualification; active utilizing the user knowledge base; constant monitoring and testing the accepted decisions; visualisation of the process of problem solving and contextness in perceiving the information; training the user during interaction with the system.

REFERENCES

Duke V., Samoilenko A., 2001 "Data mining", St. Petersburg, Russia (in Russian)

Stepashko V., 1991, "On Expert Knowledge Structuring Task in the field of modeling from empirical data", *Cybernetics and Computer Engineering, Iss. 92*, Kiev, Ukraine (in Russian)

Gavrilova T., Khoroshevsky V., 2000, "Knowledge Bases of Intellectual Systems", St. Petersburg, Russia (in Russian)

Stepashko V., Zvorygina T., 2001, "On Design of a DSS Dialog Shell for Observation Data Modeling",

Modeling and Control of State of Ecologic and Economic Systems of a Region, Kiev, Ukraine (in Russian)

Volodymyr Stepashko



Doctor of Sciences (1994), Head of Department on "Information Technologies of Inductive Modeling" of International UNESCO Center of NASU, Kyiv.

Professor of the Joint Chair of the International UNESCO Center and of the National University

"Kyiv Polytechnic Institute" for preparing of Masters of Sciences in specialty "Intellectual Systems of Decision Making", special course "Inductive Approach to Complex Systems Modeling".

Field of interests: data-based modeling of complex systems, system identification, control systems, intellectual knowledge-based systems, decision making.

Tetyana Zvorygina



Graduated from Kiev State University in 1992, specialty physicist. At 1999 entered to International Research and Training UNESCO Center of Information Technologies and Systems where get a Master degree in specialty intellectual system of

decision making. Field of interest - intellectual system of knowledge extraction, data based modeling, decision making procedures.

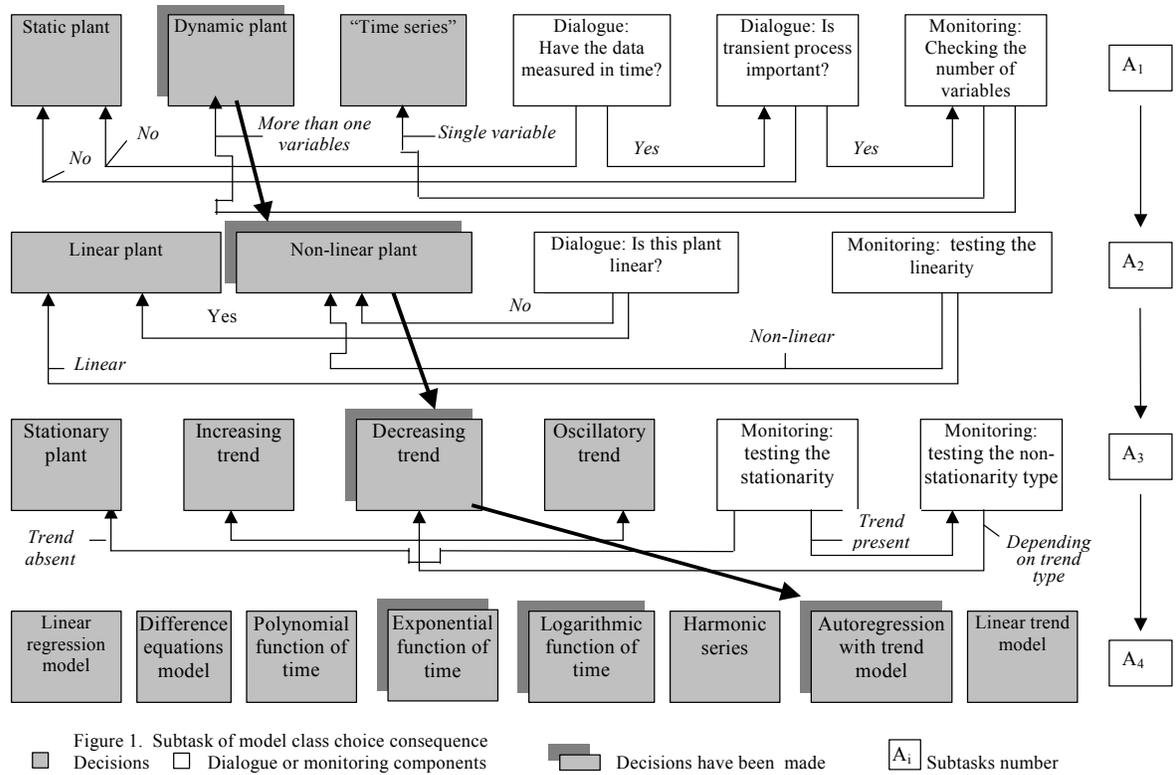


Figure-1: The 4 stages of decision making

Flow Control in Optimistic Simulation

Luiza Solomon and Carl Tropper

School of Computer Science, McGill University, Montreal, Québec, Canada.

contact: lys@cs.mcgill.ca, carl@cs.mcgill.ca

Abstract

Overly optimistic processing in Time Warp can threaten the stability of the simulation due to large memory consumption and explosive rollback growth. To address the stability concerns of optimistic simulation, Choe and Tropper proposed a learning-based flow control algorithm which throttles over-optimistic execution by regulating the flow of events between pairs of processors throughout the course of the simulation. This flow control algorithm has been shown to effectively improve simulation stability for certain applications in a shared-memory environment.

In this paper we present an analysis and experimental verification of the performance of this flow control algorithm in a distributed-memory environment. Results show that the flow control algorithm reduces the memory usage, the number of rollbacks and the number of antievents at the expense of the simulation time. Thus it becomes apparent that the behaviour of the flow control algorithm is not a consequence of learning, but it is highly dependent on the type of simulation platform, event granularity and communication latency. Taking these results into account, we discuss a number of approaches to learning and flow control using the outlines of the flow control algorithm, and we consider the extent of the performance improvement to be expected from memory-based schemes for limiting Time Warp optimism in a distributed-memory environment.

1 Introduction

The Time Warp optimistic simulation technique is designed to exploit the maximum achievable parallelism in a discrete event simulation system; thus, it has the potential to obtain excellent performance and scalability results. Unfortunately, the optimistic behaviour of Time Warp brings with it its own hazard: instability and excessive use of memory. For a system to be stable, it should be able to adapt quickly to any perturbations in the environment and maintain an acceptable level of performance. In Time Warp, however, per-

turbations such as sudden bursts of incoming events, stragglers and anti-events may cause a host to exceed its memory capacity, degrade its performance, propagate its adverse effects to the neighbors, and finally result in the congestion of the simulation system with work that would soon be rolled back. In the most extreme cases, the number of rollbacks increases without bound, making it impossible for the simulation to finish in a finite amount of time [10].

Hence, for the best performance of a Time Warp simulation system, the instability must be kept to a minimum. Numerous methods for reducing the cost and the number of rollbacks have been proposed to control instability. The approaches to reducing the number of rollbacks can be classified in two categories: the direct control approach, which aims to slow down the processes further ahead in simulated time, and the indirect control approach, which aims to limit memory consumption, in turn limiting the advance of a processor in simulated time. The earlier direct control algorithms used windowing techniques to bound the progress of all processors [10, 19, 15]. Currently the focus is on adaptive protocols, which dynamically change specific control parameters to influence the degree of throttling [7, 14, 18, 16].

The algorithms aiming for direct control do not actively deal with the possibility of a processor poorly managing its allotted memory space; such concerns, however, are the primary consideration of the indirect control algorithms.

An adaptive protocol for a shared-memory machine based on the Cancelback mechanism is presented in [5]. This protocol manages the pool of shared memory for the entire simulation and adjusts the amount of memory provided to the parallel simulator to maximize performance. An adaptive flow control mechanism is proposed in [12], also intended for a shared-memory environment. This mechanism limits the number of uncommitted events generated by a processor, thus preventing overly-optimistic execution. A window of events is used to set an upper bound on the number of uncommitted events to be scheduled in a time period; the fossil collection commits events and thus serves as acknowledgments.

In the indirect control category, Choe and Tropper [4] presented an algorithm targeted towards a distributed-memory environment which uses flow control to improve the stability and performance of Time Warp. This flow control algorithm attempts to maintain the standard deviation of the load of the processors participating in the simulation below a small bound by continuously regulating the flow of events between processors. The authors have presented results from an implementation of this algorithm on a shared-memory multiprocessor; message passing routines were used for inter-processor communication and direct use of shared memory was avoided. This paper discusses the results of the implementation of the flow control algorithm on a Beowulf cluster. To our knowledge, this is the first time an indirect optimism control algorithm has been implemented in a distributed memory environment.

The remainder of this paper is structured as follows: section 2 describes the flow control algorithm, section 3 analyses the results of its implementation on a Beowulf cluster and section 4 presents a modification to the flow control algorithm and its results. In section 5 we discuss a set of alternative approaches to learning for the flow control algorithm, and in section 6 we consider the effects of memory-based optimism limiting schemes in distributed-memory environments.

2 The Flow Control Algorithm

2.1 Motivation

It is well known that optimistic simulations can consume a large amount of memory. The large demands on memory stem from the information maintained to allow rollbacks: checkpointing the state, storing an antievent for each output event, and sending input events that are later canceled. Choe [3] provides experimental results that indicate the correlation between the rate of memory usage and the rate of increase in local virtual time for each processor during the simulation of a shuffle-ring network. The results in [3] imply that rapid progress of a processor ahead of the GVT results in larger consumption of memory and a larger number of rollbacks and antievents compared to the processors whose time advance is closer to GVT. In this case larger than average memory consumption is more than just a threat to the completion of the simulation: it is also a sign of instability. The goal of the flow control algorithm is to increase the stability and improve the performance of the simulation by ensuring that memory utilization, and by extension simulation progress, is approximately the same for all processors and that no processor runs out of memory.

Respecting these conditions requires that local load infor-

mation is frequently disseminated and shared among processes.

2.2 Description of the Algorithm

The flow control algorithm proceeds as follows: each processor is first assigned a number of permits (called tokens) by means of a stochastic learning automaton (SLA). The tokens are allocated in individual pools for each outgoing link (see also [8]). Every event sent to a neighbouring processor consumes a token from the pool allocated to that neighbour. The token pool size varies dynamically throughout the course of the simulation as a function of the differences in load (memory utilization and/or virtual time progress) between processors. A uni-directional link between a lightly loaded sending processor and a heavily loaded receiving processor is assigned less tokens in an attempt to reduce the load on the receiver; in contrast, a link between a heavily loaded sender and a lightly loaded receiver is assigned more tokens to increase the load of the receiver and reduce the load of the sender. When a processor runs out of tokens for a particular neighbour, that neighbour is considered to be fully loaded, i.e. far in memory consumption and simulation time. In this case the processor slows down the outgoing event flow to the loaded neighbour while learning the appropriate number of tokens to assign to that link in the future.

2.2.1 Control Mechanism

The control model of the flow control algorithm consists of a collection of automata such that each automaton resides within a processor and cooperates with the remaining automata to control the flow of events. The stochastic learning automaton residing at each processor regulates the outgoing flow towards the rest of the processors with the express purpose of keeping the processors close in memory usage and local virtual time. To achieve this goal, the principle of conservation of memory is used to relate the memory utilization at a processor to the memory space occupied as a result of the incoming event flow. The principle of conservation of memory states that the number of memory buffers occupied during a time interval is equal to the number of memory buffers occupied at the start of the interval together with the amount of memory buffers occupied by the events received during this interval, minus the number of buffers released by the events sent during this interval.

The stochastic learning automaton at each processor takes as inputs the load of all processors in the simulation and outputs an outgoing flow regulation factor λ . This flow regulation factor, multiplied by the number of events sent during the previous update interval, determines the number of

```

1: variables for processor  $n$ 
2:  $load_n$ : integer init 0 {current processor load}
3:  $oldload_n$ : integer init 0 {previous processor load}
4:  $token_n$ : integer init 500 {number of available tokens}
5:  $loadList[0 \dots N - 1]$ : integer init 0 {list of space-time products of all  $N$  processors}

6: if sending event  $\langle msg \rangle$  then
7:   compute  $load_n(t)$ 
8:    $load_n(t+1) \leftarrow \alpha \times load_n(t) + (1-\alpha) \times oldload_n(t)$ 
   { exponential smoothing with  $\alpha$  0.15}
9:    $oldload_n(t+1) = load_n(t+1)$ 
10:  piggyback  $load_n(t+1)$  onto a basic event:  $\langle msg, load_n \rangle$ 
11:  if  $token_{n,i} > 0$  then
12:    send  $\langle msg, load_n \rangle$  to the receiving processor  $i$ 
13:     $token_{n,i} \leftarrow token_{n,i} - 1$ 
14:  else
15:    update action probabilities
16:    compute token number
17:    send  $\langle msg, load_n \rangle$  to the receiving processor  $i$ 
18:     $token_{n,i} \leftarrow token_{n,i} - 1$ 
19:  end if
20: else if receiving event  $\langle msg, load_i \rangle$  then
21:    $loadList[i] \leftarrow load_i$ 
22:   if updating interval then
23:     update action probabilities
24:     compute token number
25:   end if
26:   process  $\langle msg \rangle$ 
27: end if

```

Algorithm 1: Flow Control at Processor n

events to be sent during the next interval. Note that the automaton computes a number of tokens individually for every outgoing link of a processor.

2.2.2 Load Metrics

A key element of the flow control algorithm is its definition of load. Occupied memory is the most obvious way of defining load, as the outgoing flow regulation at a processor depends on the principle of conservation of memory. In Time Warp, memory is consumed by state saving and the event queues, so the space metric measures the memory space occupied by events and states.

The metric employed in [3] is the space-time product, defined as the product between the occupied memory and the minimum logical virtual time of the processor at the time of calculation. The intuition behind the use of this metric is keeping the processors close in both memory consumption and simulated time. The space-time product is the load metric used in our description of the learning scheme of the

automaton.

In our experiments we also tested the effects of using time as a metric, as increases in memory consumption are postulated to mirror increases in simulated time. Time is measured as the minimum logical virtual time of the processor at the time of calculation.

Processors piggyback the local load information onto the events sent to neighbouring processors. Since every processor does not necessarily send an event to every other processor, load information is also collected from the processors in the course of the GVT calculation and broadcast to all processors together with the new GVT value.

2.2.3 Update Interval

The action probabilities of the stochastic learning automaton are periodically updated to reflect the current load of the processors involved in the simulation. Updating the probabilities frequently provides the finest control since the learning automaton keeps track of the smallest variations in the memory utilization and local virtual time, but each update takes time thus slowing down the simulation. The action probabilities are updated and the tokens are recalculated when a fixed number of events is received by a processor or whenever the processor runs out of tokens for one of its outgoing links. Currently an estimate of the best updating interval is experimentally obtained.

3 Performance Analysis

3.1 Experimental Setup

The flow control algorithm was tested on two types of applications: a queuing network application and a Personal Communication Services (PCS) network application. This section describes the behaviour of these applications.

The queuing network application simulates the behaviour of a set of computer servers connected by a network. The network is configured as a torus which has many cycles and hence induces a large amount of instability into the system.

A fixed number of messages randomly circulates through the network. Each network node spends some simulation time processing messages and generating for each input message an output message which is sent to one of the neighbouring network nodes. The outgoing link is selected using a uniformly distributed random variable. The service time for each node is constant; in our tests the service time is 3 simulation time units.

Our second application is a Personal Communication Services (PCS) network, a wireless communication network

that provides services to mobile phone users. We are using a call-initiated model as described in [2], where the objects traveling through the system are calls, each representing an active phone conversation. The channel allocation strategy is fixed: the number of channels per cell is constant. The cells are in the shape of a hexagon and are grouped in a hexagonal mesh. We use Lin and Mak’s strategy [9] to eliminate the disappearance of calls at the mesh boundary: if a call crosses outside the simulated area, it appears at the boundary edge in the opposite direction. The PCS simulation is self-initiating: each cell generates its own incoming calls. One new call is generated every time a call is started.

In our simulation, the cell diameter is 1 km and has 500 channels. A call can have 6 directions: east, south-east, south-west, west, north-west and north-east corresponding to the neighbors of each cell. The velocity and direction are determined by a uniform distribution, the call completion time is determined by an exponential distribution with a mean time of 300 seconds and the call move time is determined by an exponential distribution with a mean time of 120 seconds. Calls are generated at each cell following an exponential distribution with a mean time of 10 seconds.

The simulations were run on a 16-node Beowulf cluster. Each computer has a dual processor Intel PIII 700MHz CPU on Asus PII-BD motherboards with 384MB RAM. The network hub for the cluster is a Cisco 100Mb/s switch. The computers are running the Linux RedHat operating system. The PVM library is used for interprocessor message passing. The flow control algorithm was implemented on top of Time Warp using the TWSIM Time Warp simulator developed by our laboratory.

3.2 Experimental Results

To compare the behaviour of Time Warp with the behaviour of Time Warp with flow control, we used the following performance measures: (1) simulation time, (2) memory used, (3) number of rollbacks and (4) number of antievents. Graphs present the effect on each performance measure using three different load metrics: space, time and space-time product. Each graph also has data points labeled “Data Pass” to indicate the overhead incurred by calculating and transmitting the necessary metrics for the flow control algorithm on top of the regular Time Warp computation but without employing the flow control mechanism. The results presented are an average over ten consecutive runs of the program. The starting number of tokens is 500 and the updating interval is set to 100 received events.

Figure 1 presents the performance results for a queuing network simulation on a 12-node torus with 75 starting events per queuing network node. The total of good events processed in 500,000 units of simulated time is 28,085,587.

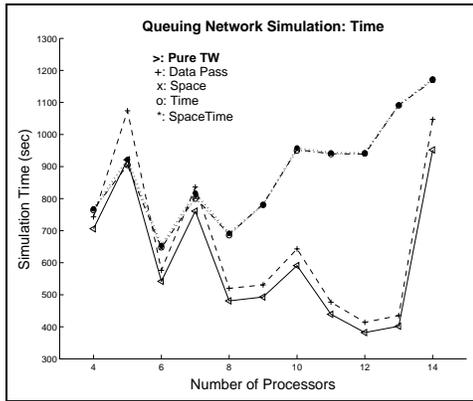
The simulation shows a better performance on some numbers of processors than on others. This phenomenon is caused by cycles of the torus network and exacerbated by the logical process partitioning and the communication delays. Figure 2 shows the performance results for the PCS simulation on a hexagonal wrap-around mesh of side 80. Each cell initiates two calls at the start of the simulation, resulting in a total of 31,325,453 good events processed in 2,000 units of simulated time. In some of the figures, notably the ones showing time progress, the Space, Time, and Space-Time data points are very close to each other and often coincide.

The graphs show that the flow control algorithm does reduce the number of rollbacks and antievents if it is large; the memory usage is also reduced, the reductions being more significant as the number of processors increases. However, this decrease in memory usage occurs at the expense of the simulation time, showing yet again the space-time trade-off in distributed simulation. None of the metrics tried appears to perform better than the others.

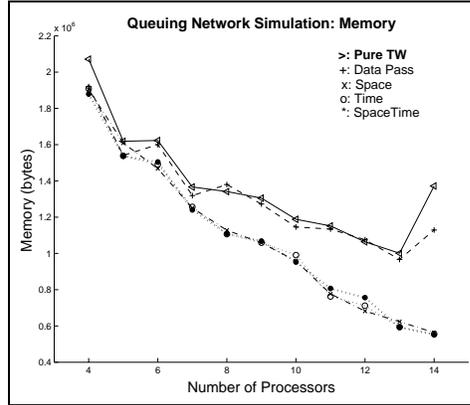
3.3 Analysis of Results

A close look at the particulars of the flow control algorithm shows that the observed results are not a consequence of learning. The learning mechanism is not engaged because recalculating the tokens for all outgoing links as soon as one link runs out of tokens causes the number of assigned tokens to decrease to 1 (our lower bound) after only a few token re-computation steps. According to the control mechanism, the maximum number of tokens allowed to depart from processor P_i to processor P_d in the time interval $[t, t + 1)$ is $\lambda_{i,d}(t)D_{i,d}(t - 1)$, where $D_{i,d}(t - 1)$ is the number of events sent during the previous time interval $[t - 1, t)$ and $\lambda_{i,d}(t)$ is the regulation factor as computed by the learning automaton. If the tokens for all links are recalculated when the tokens for one link are consumed, then the time interval $[t, t + 1)$ does not have the same length as the time interval $[t - 1, t)$, and the number of tokens used in the time interval $[t - 1, t)$ is not representative for the next time interval.

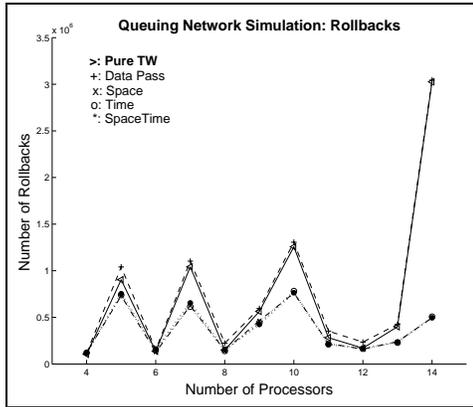
To illustrate, assume that processor P_1 has two outgoing links to processors P_2 and P_3 , with the link to P_2 receiving 10 tokens and the link to P_3 receiving 5 tokens for the next time interval. Furthermore, assume that at the beginning of the interval more events are sent to P_3 and fewer events are sent to P_2 than expected; at the time when the tokens for the link to P_3 are exhausted, only one of the tokens for the link to P_2 has been used. If the regulation factor for P_2 as determined by the learning automaton using the load information is less than 1, then the link to P_2 will be assigned only one token during the next interval (since 1 is the lowest bound on the the number of tokens). This single token will be used



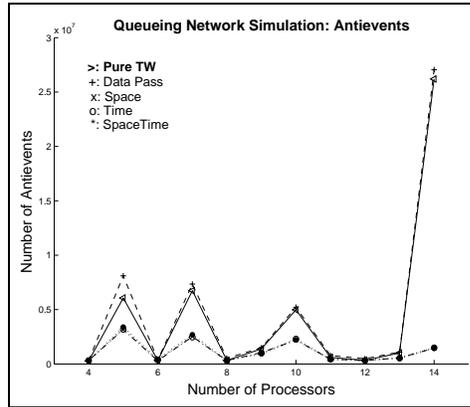
(a)



(b)



(c)



(d)

Figure 1: Performance of the flow control algorithm for a queuing network model on a torus-shaped network.

very fast, prompting P_1 to also decrease the tokens for its link to P_3 . In just a few iterations the tokens on all outgoing links of all processors decrease to 1.

The result of assigning only one token per outgoing link is that the token recalculation occurs very frequently, approximately once for every two events sent. As such, its only effect is to increase the granularity of the event computation. Since the simulations applications used have very small event granularity, the token recomputation time is significant in comparison. In the PCS simulation, the average event processing time on our system is 0.38 milliseconds, while the average token computation time is 0.16 milliseconds. The average rollback time is 0.39 milliseconds. Therefore, increasing the granularity by such a small amount results in these circumstances in a partial ordering of the events during the simulation and a reduced number of rollbacks and antievents.

3.4 Consistency Check

The performance results of the flow control algorithm on a shared-memory machine presented by Choe and Troller [4] are consistent with these findings. The authors did obtain a small reduction in the simulation time (3 to 10 percent) as compared with the Time Warp simulation with no flow control. This reduction is expected to be caused by the more significant contribution of rollbacks to the total simulation time in a shared-memory environment; in a distributed-memory environment, the communication costs dilute the effect a considerable reduction in the number of rollbacks and antievents can have on the execution time of the simulation. The biggest reduction in the simulation time was obtained in the case of a stress test which directed a large percentage of the messages sent by each processor to one designated processor during a fixed time interval (15% to 28%). Since the number of uncontrolled rollbacks is very

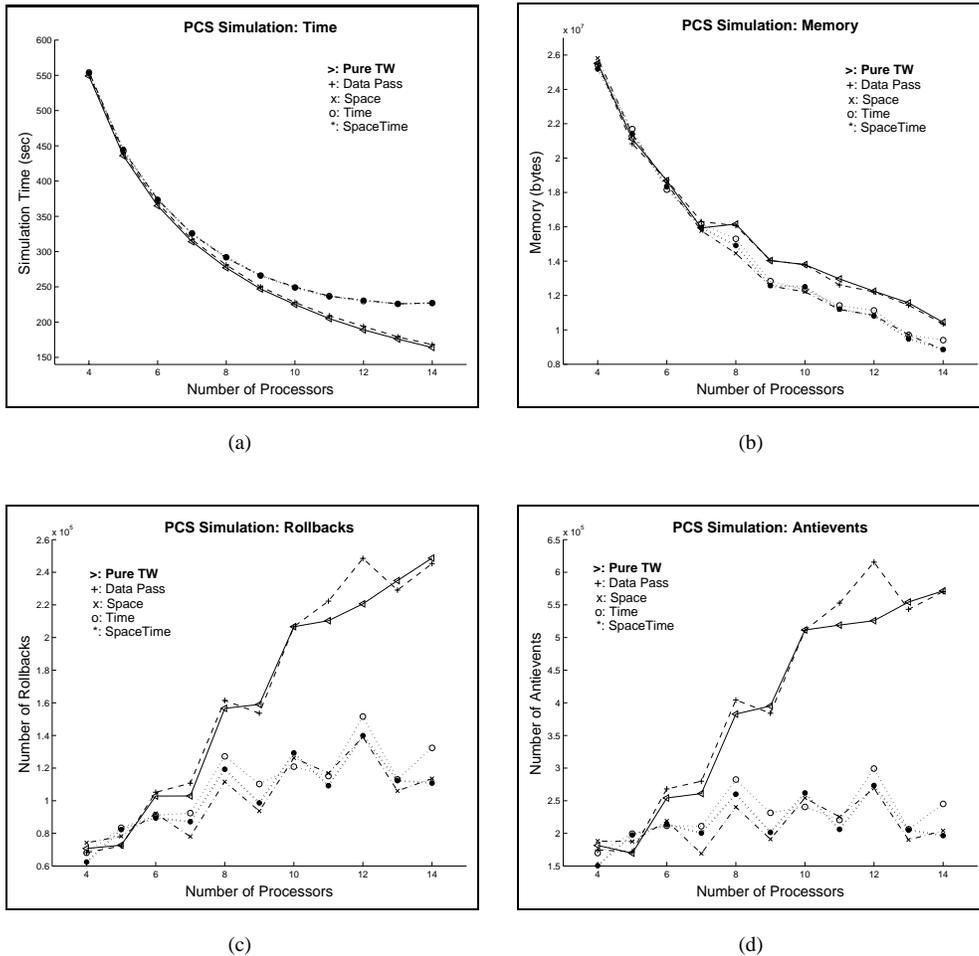


Figure 2: Performance of the flow control algorithm for a PCS model on a hexagonal mesh network.

high in this case, any technique that will reduce the number of rollbacks by a significant percentage is likely to obtain very good results in terms of execution time. Our graphs show that for the queuing network simulation a very large reduction in the number of rollbacks for 14 processors resulted in a much smaller increase in the simulation time.

In addition, it is probable that the performance in Choe and Tropper's case was further improved by his implementation of the GVT algorithm. He uses an election mechanism to select the GVT initiator based on the simulation load: the processor with the highest memory consumption starts the new GVT round. Since computing the GVT is a time intensive process, through this election process the busiest processor is slowed down to the advantage of the rest of the simulation. Our current implementation of the GVT algorithm dispenses with the election process and assigns a single GVT initiator at the beginning of the simulation.

Another factor that suggests that the results presented by Choe and Tropper are not a consequence of learning is the very small reduction obtained in the variability of the space-time product: the flow control scheme generated a reduction of 4.4% in the mean of the space-time product and a reduction of 9% in the standard deviation of the space-time product. This small reduction is likely to be the effect and not the cause of the decreased number of rollbacks and antievents.

4 Modifications to the Flow Control Algorithm

The assigned tokens act as a window indicating the number of events that can be sent to the neighbouring processors without delay between updates, and the total number of events sent between updates (with or without delay) is

used to compute the number of tokens for the next interval. For this computation to be meaningful, the time interval between updates has to be approximately the same from one interval to the next; hence, the tokens should be recalculated only every *updating interval* and not every time a link runs out of tokens. A possible implementation of the algorithm taking this issue into consideration would be as follows: only learning (probability recalculation) occurs when a link runs out of tokens (deleting line 16 of algorithm 1) and no token counter is decremented as the tokens have already reached 0 (deleting line 18). A sent event counter is incremented for every outgoing event, and this sent event counter is used to compute the number of tokens for the next interval. The probability recalculation when the tokens for the current interval have been exhausted would then serve the double purpose of accelerating learning and providing a respite for the receiving processor via a delay in sending the outgoing event. We studied the effect this modification of the flow control algorithm had on the performance of the simulation.

Figures 3 and 4 present the performance of the modified flow control algorithm for the queuing network simulation and the PCS simulation with the same parameters as in the previous section¹. The algorithm did not improve the performance of the simulation; on the contrary, it increased the execution time together with the number of rollbacks and antievents. It is interesting to note that in the case of the queuing network simulation on 14 processors the modified flow control algorithm did reduce the execution time, the memory consumed and the number of rollbacks and antievents, illustrating that when communication delays and cycles induce an ever-increasing number of rollbacks any technique that slows down the simulation is likely to improve the simulation performance.

Since the probability recalculation when the tokens for the current interval have been exhausted serves as a delay for the outgoing events, it seemed possible that the time spent in calculations is not significant compared with the communication time. Experiments have been done to determine whether increasing the delay preceding the sent event when all the tokens have been used has an effect on the performance of the flow control algorithm. However, the additional time spent waiting worsens the performance of the simulation.

It is worth noting that for small queuing network simula-

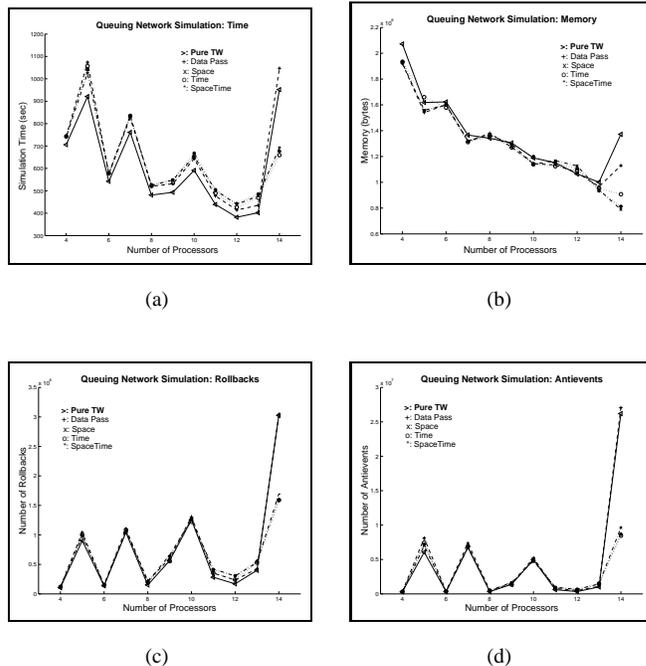


Figure 3: Performance of the modified flow control algorithm for a queuing network model on a torus-shaped network.

tions in which only one logical process is assigned per processor the additional delay does decrease the number of rollbacks and antievents despite increasing the simulation time. We conjecture that this effect occurs because in a distributed-memory environment the speed of propagation of events within a processor is much faster than the speed of propagation of events between processors. Since only the flow of inter-processor events is controlled, the bad computation among logical processes located on the same processor is allowed to proceed unimpeded, and a rollback caused by an outside event has far-reaching effects. However, if no new events are generated and no inter-processor rollbacks can occur, as in the case of small queuing network simulations, an increased delay gives the antievents an opportunity to catch up with the original events before they are processed.

Furthermore, we observed experimentally that varying the length of the update interval made no difference to the performance of the modified flow control algorithm. The insensitivity of the algorithm to the interval length, combined with its bad performance, suggests that no learning actually takes place.

¹Figure 3 shows the performance results for a queuing network simulation on a 12-node torus with 75 starting events per queuing network node. The total of good events processed in 500,000 units of simulated time is 28,085,587. Figure 4 shows the performance results for the PCS simulation on a hexagonal wrap-around mesh of side 80. Each cell initiates two calls at the start of the simulation, resulting in a total of 31,325,453 good events processed in 2,000 units of simulated time. The starting number of tokens is 500 and the updating interval is set to 100 received events.

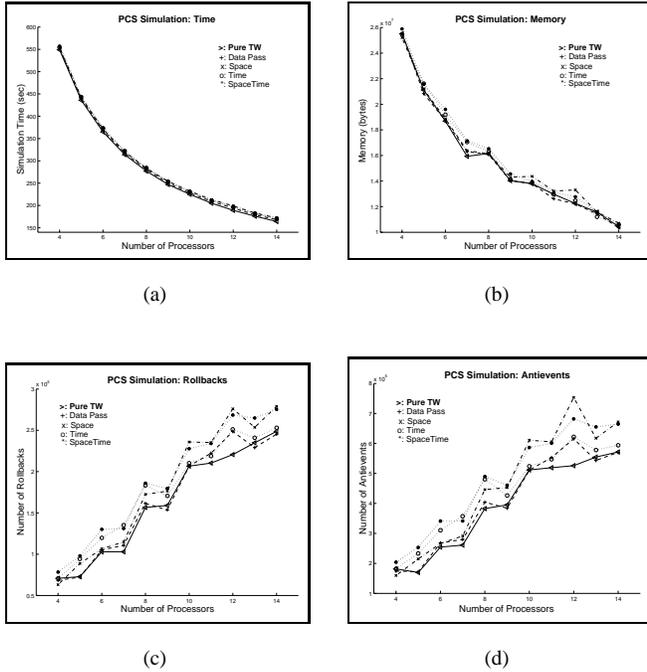


Figure 4: Performance of the modified flow control algorithm for a PCS model on a hexagonal mesh network.

5 Alternative Approaches to Learning and Flow Control

The stated goal of the flow control algorithm is to keep in close proximity the memory utilization of the processors involved in the simulation, and by extension keep in close proximity their virtual time as well. This goal is intended to be achieved by managing the flow of memory buffers between processors, delaying events whose processing is likely to cause rollbacks and allowing unimpeded passage to events that are likely to be on the critical path of the simulation. These concepts have been successfully exploited in other optimism-limiting algorithms. Panesar and Fujimoto [12] proposed a memory-based flow control mechanism which improved Time Warp performance in a shared-memory environment by throttling over-optimistic event execution. Tay et al. [17] demonstrated that bringing the sender and receiver logical processes closer in virtual time resulted in reduced number of rollbacks, as no logical process is allowed to dramatically increase its simulation time and flood the simulation with events that will soon be rolled back. The algorithm of Srinivasan and Reynolds [14] controlled optimism by delaying executions of events according to their error potential computed from global information. Therefore, it seems probable that an algorithm can be built along the outlines of Choe and Tropper’s flow control algo-

rithm which could control Time Warp optimism.

The succeeding sections examine several aspects of the flow control algorithm and discuss the changes and issues to consider in order to effectively harness the power of the learning automata and increase simulation stability.

5.1 Token Computation

Stochastic learning is a theoretically simple technique which is difficult to implement efficiently. A learning automaton must be fed information in a timely manner in order for its control to be effective. For this reason, the amount of instability in the system can have a large impact on the learning techniques employed. Simulations run on distributed-memory systems are inherently more unstable than those on shared-memory systems; for example, Choe and Tropper obtained 3,449 rollbacks in 330 seconds for his pure Time Warp simulation on 6 processors for a PCS application with a wrapped hexagonal mesh network of side 80. In contrast, our pure Time Warp simulation of a similar application over 6 processors produced 85,435 rollbacks in 206 seconds, a sizable difference. Next we examine the challenges posed by an unstable environment and propose ways in which the learning scheme can be modified to cope with them.

A particularly unstable simulation (for example, one using a configuration with many cycles on a distributed system with large communication costs) can enter into a phase of cascading rollbacks very soon after start-up, preventing the automaton from acquiring any notion of stable behaviour. Even when the simulation achieves some stability of its own after the initial chaotic starting phase, the automaton must not give too much importance to what it learned during this phase. Hence, either the automaton should delay learning until the system stabilizes to some extent, or it should keep the maximum gain low enough not to give too much credence to the initial data; the flow control algorithm, with its maximum gain of 1, does not follow either course. As well, the flow control algorithm bases the token calculation for the next interval on the number of sent events of the previous interval *only*. However, if in the previous interval the outgoing traffic had an uncharacteristic pattern (for example, a large number of new events have been injected into the simulation), the number of tokens for the next interval will be calculated based on non-representative information and the automaton control will not be efficient. A better course of action would be to obtain an estimate of a representative number of sent events over several past intervals, giving the most recent interval the largest weight.

The accuracy of the information the automaton uses to update its probabilities is also called into question. The most recent information about the load of the other processors

is obtained from data piggybacked on the incoming events; otherwise, a processor is guaranteed new data only every GVT calculation. If the one-directional traffic between two processors is intermittent – or nonexistent – the automaton uses old information and loses efficiency. The problem of obtaining accurate and timely global data has no easy answer; one possibility could be the reduction model for computing near-perfect state information presented in [13], which has been implemented on a network of workstations connected by a Myrinet switch and shown to be feasible for simulations with medium to large event granularity.

5.2 Token Utilization

However, no matter how accurately the learning automata estimate the number of tokens required for the next interval, these tokens must be used in an efficient manner in order to control the simulation. The flow control algorithm attempts to keep all processors close in terms of memory usage and virtual time by starving the processors with large memory consumptions. This starving process is accomplished by delaying the exit of events from the lightly loaded processors to the heavier loaded ones through a waiting loop, with the effect that the lightly loaded processors are slowed down themselves. This method could potentially restrain the spread of bad computation from the highly loaded processors, giving the loaded processors an opportunity to roll-back and send out antievents before the original events have gone too far. On the other hand, the same method could slow the spread of antievents as well, since the flow control algorithm deals with memory buffers only and does not consider the type of event to be delayed. Moreover, the optimal size of these delays is platform and application dependent, since they must be significant compared to the event granularity and the inter-processor communication time.

The disruptive effect of delays could be minimized by allowing the lightly loaded processor to continue activity while delaying the exit of events lacking tokens. The event delay could be measured in terms of a specific number of processed events or a fixed time period. The events can also be held hostage until the next updating period when more tokens are assigned. This approach of delaying outgoing events presumed to be bad reduces risk; alternatively, aggressiveness can be reduced as in the case of the Adaptive Flow Control algorithm [12] by suspending event execution and communication until the next token updating period. Aggressive blocking has also the potential to reduce the spread of bad computation within a processor as well as the length of rollbacks caused by out-of-processor events, making it a more suitable strategy for distributed-memory environments.

An implementation of the risk-reducing version described

above, where events lacking tokens are held until the next updating interval, shows that such an approach leads to deadlocks if the simulation is lightly populated and to stalling if the simulation is densely populated. The stalling occurs because the event with the smallest timestamp gets caught in one waiting queue after another, and the same happens to the antievents. When the tokens are recalculated at the end of the update interval and the events are released from the waiting queues, a large majority of the events processed during the last GVT interval are rolled back and the GVT cannot advance. The introduction of a cancellation mechanism between events and their respective antievents in the waiting queue did not have any effect. The aggressiveness-reducing version can also deadlock, and it is not obvious how to break the deadlock and resume the simulation in the context of the learning automata with minimal time expenditure without voiding the learning that has occurred up to that point.

An alternative approach could be to change the placement of the learning automata. If the automata reside at the destination and not the source processors, the processing delays consisting of waiting loops happen at the heavily loaded processors, which seems a more desirable course of action than delaying the lightly loaded processors. However, there are serious implementation complications with this technique as well. If the events are allowed to queue at the heavily loaded processor until they get enough tokens to be processed, that processor will have an even higher memory consumption, contrary to the goal of lightening the load. A sendback mechanism might alleviate this concern, but this approach would also provide additional work for the lightly loaded processors who would have to deal with the events sent back.

5.3 Space-Time Correlation

The correlation between the rate of memory usage and the rate of increase in local virtual time for each processor during the simulation of a shuffle-ring network is indicated by experimental results presented by Choe [3]. As a consequence of these results, the case was made that overconsumption of memory is a sign of instability indicating a disproportionate progress in virtual time compared to other processors. However, this conjecture has not been formally proven; a negative proof may have implications for the flow control algorithm. Intuitively, if processor P_1 is ahead of processor P_2 in memory consumption, then P_2 should withhold events from P_1 to not increase P_1 's memory consumption; as well, P_1 should be free to send events to lower its memory usage. In contrast, if processor P_1 is ahead of processor P_2 in simulated time, a better course of action for P_2 would be to send events to P_1 to roll it back as soon

as possible. In this case, allowing P_1 to send an unlimited number of events risks flooding the simulation with computation that would need to be rolled back. The correlation between memory usage and virtual time progress requires further analysis, especially as related to the principle of conservation of memory used by the flow control algorithm.

Our experiments were not conclusive with regard as to which load metric shows the best promise for future research. However, it appears that the space-time product as it is currently calculated mirrors in behaviour the space metric. The reason is that the memory size used to calculate the product is measured in bytes. If the virtual time advances very slowly compared to the increase of memory usage, the space-time product is heavily weighed in the favor of memory and might not offer any new information.

6 Memory-Based Optimism Limiting Schemes and Distributed Memory

Memory-based optimism limiting schemes have been successfully implemented up to now on shared-memory multiprocessors. In shared-memory environments controlling memory consumption serves a dual purpose: first, cache performance is improved by increasing locality and decreasing false sharing of virtual memory pages, and second, harmful optimism is eliminated through the equivalent of a simulated time window. To our knowledge no experiments have been done to determine which of these two factors results in the biggest performance improvement.

However, the negative consequences of loss of spatial locality and false sharing of memory pages for optimistic simulation have been extensively documented and found to be significant [8, 5, 6]. The effects of poor cache performance are exacerbated by the increasing gap between the memory and CPU speed [11]. Furthermore, experiments have shown that in shared-memory environments simulation performance for a memory-intensive application is considerably affected by the dense bus traffic. In contrast, the same simulation in a distributed memory environment, lacking the bus overcrowding, outperformed its shared-memory counterpart by approximately 50% [1].

In the light of these results, it seems likely that simulations executed in shared-memory environments will benefit more from memory-based approaches to reducing Time Warp instability than simulations in distributed-memory environments. Before extensive research is undertaken to design a memory-based optimism limiting algorithm targeted towards a distributed-memory environment, it would be useful to ascertain the degree of performance improvements that can be expected from such an algorithm. The implemen-

tation in a distributed-memory environment of a memory-based algorithm which has been proven successful at limiting optimism in shared-memory environments would provide valuable information in this regard.

7 Final Remarks

There are many options to be explored regarding the best way to implement the learning automata and use their results, and each one has its own advantages and drawbacks. Clearly more experimentation is necessary before an effective version of the flow control algorithm can be implemented on a network of workstations. But before this work can be undertaken it has to be established the extent to which controlling the memory consumption in optimistic simulation can improve stability and performance in a distributed memory environment. It is possible that in such an environment methods that directly limit optimism have the best chance of success.

References

- [1] C. J. M. Booth, D. I. Bruce, P. R. Hoare, M. J. Kirton, K. R. Milner, and I. J. Relf. Dynamic memory usage in parallel simulation: a case study of large-scale military logistics application. In *Proceedings of the 1996 Winter Simulation Conference*, pages 975–982, 1996.
- [2] C. D. Carothers, R. M. Fujimoto, and Y-B. Lin. A case study in simulating pcs networks using time warp. In *Proceedings of the 9th Workshop on Parallel and Distributed Simulation*, pages 87–94, 1995.
- [3] M. Choe. *Distributed Process Cooperation in Time Warp*. PhD thesis, McGill University, 1999.
- [4] M. Choe and C. Tropper. On learning algorithms and balancing loads in time warp. In *Proceedings of the 13th Workshop on Parallel and Distributed Simulation*, pages 101–108, 1999.
- [5] S. Das and R. Fujimoto. Adaptive memory management and optimism control in time warp. *ACM Transactions on Modeling and Computer Simulation*, 7:239–271, 1997.
- [6] S. R. Das and R. M. Fujimoto. An empirical evaluation of performance-memory trade-offs in time warp. *IEEE Transactions on Parallel and Distributed Systems*, 8:210–224, 1997.

- [7] A. Ferscha and J. Luthi. Estimating rollback overhead for optimism control in time warp. In *Proceedings of the 28th Annual Simulation Symposium*, pages 2–12, 1995.
- [8] R. Fujimoto and K. Panesar. Buffer management in shared-memory time warp systems. In *Proceedings of the 9th Workshop on Parallel and Distributed Simulation*, pages 149–156, 1995.
- [9] Y-B Lin and V. W. Mak. Eliminating the boundary effect of a large-scale personal communication service network simulation. *ACM Transactions on Modeling and Computer Simulation*, 4:165–190, 1994.
- [10] B. Lubachevsky, A. Weiss, and A. Schwartz. An analysis of rollback-based simulation. *ACM Transactions on Modeling and Computer Simulation*, 1:154–193, 1991.
- [11] R. A. Meyer, J. M. Martin, and R. L. Bragodia. Slow memory: the rising cost of optimism. In *Proceedings of the 14th Workshop on Parallel and Distributed Simulation*, pages 45–52, 2000.
- [12] K. Panesar and R. Fujimoto. Adaptive flow control in time warp. In *Proceedings of the 11th Workshop on Parallel and Distributed Simulation*, pages 108–115, 1997.
- [13] S. Srinivasan, M. J. Lyell, P. F. Reynolds Jr., and J. Wehrwein. Implementation of reductions in support of pdes on a network of workstations. In *Proceedings of the 12th Workshop on Parallel and Distributed Simulation*, pages 116–123, 1998.
- [14] S. Srinivasan and P. F. Reynolds. Elastic time. *ACM Transactions on Modeling and Computer Simulation*, 8:103–139, 1998.
- [15] J. S. Steinman. Breathing time warp. In *Proceedings of the 7th Workshop on Parallel and Distributed Simulation*, pages 109–118, 1993.
- [16] L. Suppi, F. Cores, and E. Luque. Improving optimistic pdes in pvm environments. In *Recent Advances in Parallel Virtual Machine and Message Passing Interface. 7th European PVM/MPI Users' Group Meeting*, pages 304–312, 2000.
- [17] S. C. Tay, Y. M. Teo, and R. Ayani. Performance analysis of time warp simulation with cascading rollbacks. In *Proceedings of the 12th Workshop on Parallel and Distributed Simulation*, pages 30–37, 1998.
- [18] S. C. Tay, Y. M. Teo, and S. T. Kong. Speculative parallel simulation with an adaptive throttle scheme. In *Proceedings of the 11th Workshop on Parallel and Distributed Simulation*, pages 116–123, 1997.
- [19] S. J. Turner and M. Q. Xu. Performance evaluation of the bounded time warp algorithm. In *Proceedings of the 6th Workshop on Parallel and Distributed Simulation*, pages 117–126, 1992.

CGI CONTROL OF REMOTE TELECOMMUNICATION EQUIPMENT

J.C. SIMNER*, S. BECK**, M. WUWER**, T. OSMAN*** and D. AL-DABASS***

* Siemens Communications
Technology Drive, Beeston,
Nottingham, NG9 1LA.
john.simner@siemens.com

** Siemens Communications
Munich
Germany.

*** School of Computing & Mathematics
The Nottingham Trent University
Nottingham, NG1 4BU.
taha.osman@ntu.ac.uk

Abstract: Cordless handsets allow a user to make and receive calls anywhere within the range of the base stations. The base stations provide the low power cellular radio communications to the cordless handsets. The performance of the cordless equipment should be monitored to ensure that calls are not being lost. Users may be aware of some lost calls because they were talking at the time the call failed. They would not be aware of any incoming calls that fail to ring their handsets. Any lost calls could result in loss of business. This paper highlights the limitations of local monitoring. It explores some examples of remote monitoring connected with network management and rail transportation to see whether the technologies used can enhance the collection of cordless statistics on Hicom. It successfully combines technologies from the different examples to control the collection of cordless statistics on the remote telecommunication equipment. It uses a well-established client-server technology in a new way. It does not use it in the normal way to return information in a web page. Instead, it uses it to control the starting and stopping of the statistics collection.

Keywords: network management, remote monitoring

1. INTRODUCTION

Siemens Information and Communication Networks designs, develops, manufactures, and markets telecommunication equipment (termed "switch") that supports cordless equipment. Figure 1 shows a typical Hicom switch with private cordless equipment, telephones, and a local administration and service terminal connected to the Public Network.

Cordless handsets allow a user to make and receive calls anywhere within the range of the base stations. The base stations provide the low power cellular radio communications to the cordless handsets. The performance of the cordless equipment should be monitored to ensure that calls are not being lost. Users may be aware of some lost calls because they were talking at the time the call failed. They would not be aware of any incoming calls that fail to ring their handsets. Any lost calls could result in loss of business.

The goal of this paper is to investigate alternative monitoring scenarios and subsequently, produce a control mechanism that starts and stops the monitoring, which can be applied to a commercial product.

Section 2 highlights the limitations of local monitoring and investigates remote monitoring as an alternative to local monitoring. It explores two examples of remote monitoring (network

management and rail transportation) to see whether the technologies used can be applied to cordless.

Section 3 proposes remote collection of the statistics using a well-established client-server technology. It focuses on the control aspects of the remote collection system and shows how the chosen technology can be used to convey the user's requests between different parts of the system. To protect customers' investment, the chosen technology must run on both old and new Hicom switches. This limits the choice to those technologies that are supported on old switches.

Section 4 surveys different variants of the client-server technology and considers how appropriate they are to the proposed solution.

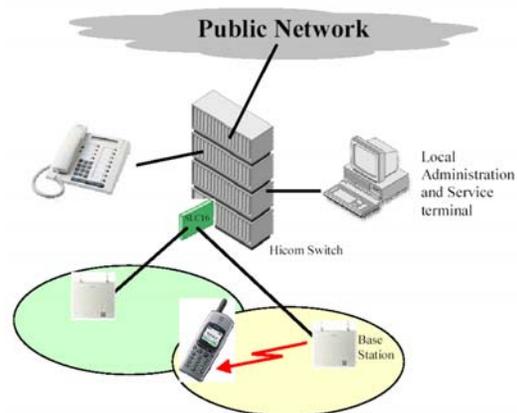


Figure 1 - Typical Private Cordless Telecommunication Equipment

Section 5 outlines tests performed and tools used to gauge how successful the chosen client-server technology is in meeting its requirements. It identifies security issues with the chosen technology and proposes using a more secure form for telecommunication equipment.

2. LIMITATIONS OF LOCAL MONITORING

Section 1 highlighted the importance of monitoring the cordless equipment to ensure that it provides the best service and that cordless calls are not lost.

The switch can measure and record many different aspects of cordless calls. It records the number of times a particular aspect has occurred since the switch was last reset. This provides an absolute measurement rather than a historical record over time.

The recorded information is accessed using a proprietary management interface. An engineer can generate historical information by manually invoking the collection commands on a periodic basis, collecting their output, extrapolating the current values, and comparing them with the previous values. Even on a small switch, the collection commands can generate hundreds or thousands of lines of textual output.

There are two ways of accessing the management interface; locally or remotely through a modem. The main problem with the remote connection is the speed of the modem link especially with the amount of textual output generated. Sometimes, the periodic period is increased because of the time it takes to receive the generated output. This could affect the worth of the historical data.

Hence, when monitoring produces a large amount of data, the most efficient way to collect it is to send an engineer to site with a laptop to locally collect the statistics. This process is very expensive in time and manpower.

The statistics are collected in blocks of 15 minutes. Whilst, the statistics are being collected, the engineer must remain on site. A typical site visit to collect the statistics is 3 hours plus travel time.

However, when monitoring produces a small amount of data, it can be remotely collected through a modem.

ALTERNATIVE MONITORING APPROACH

The previous section highlighted the cost of local monitoring. This section investigates remote monitoring as an alternative to local monitoring. There are many published examples of remote

monitoring. This section considers two examples; network management and rail transportation.

Remote Monitoring Example - Network Management

Network Management Systems are a typical example of remote monitoring. Typically, a central network manager polls the nodes, collects data from them, processes it, and presents it in a visual form to a human operator.

Kooijman (1995) and Gavalas et al. (2000) propose using agents to reduce the amount of data passed between the nodes and the server and the amount of processing done by the server.

The current cordless monitoring approach is inefficient because it transfers so much data.

Network management and agent technology have not been explored further because agent technology is not supported on old switches.

Remote Monitoring Example – Rail Transportation

Nieva, Fabri, and Wegmann (2001) and Fabri, Nieva, and Umiliacchi (1999) developed “ a web-based monitoring tool for trains ... [that] allow[ed] maintenance staff to supervise railway equipment from anywhere at anytime.” (Nieva, Fabri, and Wegmann 2001, p1)

They identified the significant benefits including; reduced development, installation, and maintenance personal travel costs. These cost savings are just as pertinent for a service organisation.

They developed and compared three prototypes based upon different technologies; HTTP with CGI, Java RMI, and HTTP with XML. The prototypes were used to monitor a single device on a train, all devices on a single train, and all devices on a fleet of trains, respectively.

They found that the CGI approach was a “fast-to-develop and elegant [solution]” (Fabri, Nieva, and Umiliacchi 1999, p12) that suffered from using a proprietary protocol between the client and the server.

The Java RMI approach pushed the data from the server whilst both HTTP approaches pulled it. Nieva, Fabri, and Wegmann found firewall security problems with pushing the data. The main advantage of pushing over pulling is the reduction in communication overhead because the data is only sent when it changes.

The XML approach enabled the data and its meaning to be sent to the client. This allows the data to be interpreted by the client.

Nieva, Fabri, and Wegmann compared the performance of the three prototypes for one and ten

updates. They found that the HTTP approach is slower than the Java RMI approach, and “the difference between the performances of Java RMI against HTTP will increase as we increase the number of updates.” (Nieva, Fabri, and Wegmann 2001, p5).

The two HTTP approaches, HTTP with CGI and HTTP with XML, could both be used to collect the cordless statistics. The CGI approach is quicker to develop than the XML approach but the XML approach would allow for future enhancements as the data and its meaning are both collected. However, the CGI approach operates faster than the XML approach as less data is being transferred between the switch and the control centre.

The Java RMI approach is not appropriate to cordless because it uses Java technology on both the client and the server. The Java technology could be added to new switches but is not supported on old switches.

3. THEORY OF NEW IDEA

The previous sections highlighted limitations with local monitoring and investigated remote monitoring as an alternative. This section proposes a remote collection system for Hicom using a client-server technology.

There are a number of aspects to the remote collection system; controlling the starting and stopping of the collection process, running the collection commands, processing and transferring the collected data. Section 3.1 describes the remote cordless collection scenario that the new idea must work within.

The remainder of this paper focuses on the control aspect between the Manager and the Hicom switch using a client-server technology. It does not address the collection process, which uses an established mechanism.

3.1 Remote Cordless Collection Scenario

Figure 2 shows the remote cordless collection scenario. There are three areas; the Administration and Service (A&S) Client, the A&S Platform on an Intel-based Server (termed “the Manager”), and the A&S Platform on a proprietary card (ADP) installed within the Hicom switch (termed “the Assistant”). The Manager can remotely access one or more Assistants.

A user logs onto the Manager to control and view the statistics on the Hicom switch. When the user starts or stops the data collection, the request is conveyed through the Manager to the Assistant. The Assistant periodically collects the data from the Hicom switch by invoking the collection commands

and analysing their output. Subsequently, the Manager remotely collects the processed data from the Assistant.

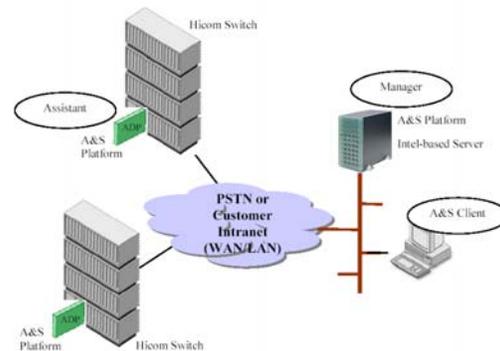


Figure 2 - Remote Cordless Collection Scenario

Overview of Control using Client-Server Technology

The control aspects of the remote collection system are the passing of the user’s requests (i.e. start collection and stop collection) from the Manager to the Assistant. To protect customers’ investment, the chosen client-server technology must run on both old and new Hicom switches. This limits the choice to those technologies that are supported on old switches. This means that it must be a well-established technology rather than one developed in the last few years.

The Apache web server and Common Gateway Interface (CGI) were chosen. They are freely available, UNIX-based, non-proprietary and widely used.

CGI is used to convey the user’s requests to the Assistant. Normally, CGI is used to return a web page to the client but the remote cordless collection system uses it to control the collections. The alternative CGI invocations that can be used to convey these requests are described later.

Using CGI allows remote access through the Internet, Intranet, or any other open network to the telecommunication equipment. There is concern that such access will allow the telecommunication equipment to be more open to attack. As it is impossible to eliminate these attacks, their effects must be minimised.

Whilst, the CGI script is being invoked, there is very limited feedback. It does not return success or failure. This minimises the information returned to a potential hacker. In addition, a barrier is required between the CGI script interface and the rest of the collection process. The barrier must utilise

minimum resources; memory and processor. A natural barrier is the creation and deletion of a control file. To ensure that the barrier is effective, the CGI script interface must not allow; any user input to be executed by the system or the user to interrupt it and take control.

Figure 3 shows an overview of the remote collection system focusing on the control aspects.

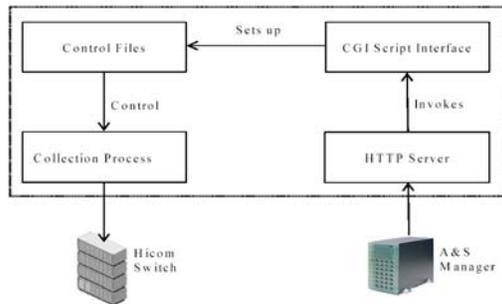


Figure 3 - Remote Collection Overview – Control Aspects

CGI Invocations

CGI is used to convey the user’s request from the Manager to the Assistant. Figure 4 shows two alternative invocations.

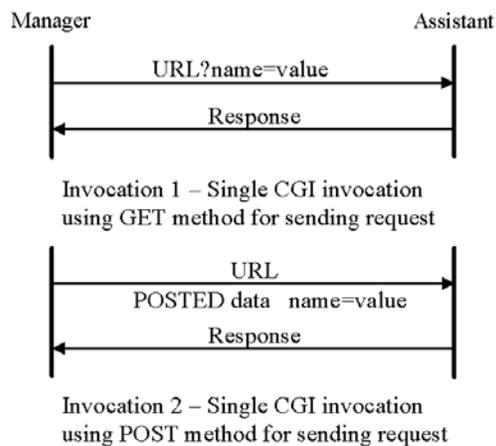


Figure 4 - Alternative CGI Invocations

The simplest CGI invocation is the GET method. The user’s request is appended to the URL. The GET method is very insecure. The URL and the user’s request can appear in the browser location bar and be logged by any system the request travels through. With no GUI, there is no browser location bar so the only concern is the logging by other systems.

An alternative CGI invocation is the POST method. The user’s request is transmitted immediately after the URL. One advantage of this method is the unseen data. The Manager requires unseen data to hide the security measures.

Kargl, Maier, and Weber (2001) identify that a system can be attacked at different levels. This section focuses on the user’s request protocol between the Manager and the Assistant. Eronen (2001) and Moore, Voelker, and Savage (2001) identify the difficulties in detecting and tracing flood attacks, which consume CPU and memory resources. Eronen and Meadows (2000) identify countermeasures to distinguish the valid requests from the rogue requests.

As the GET and POST invocations have no protection against denial of service attacks, security measures or a commercial product (e.g. Password Hurler Protection) are required to reduce the effect of flood attacks.

The security measures enable the Assistant to validate the user request with minimum processing and memory utilisation. It can check IP address of the visitor, the content length, and key value pairs. If the request is invalid, it is immediately ignored and nothing is returned. This behaviour is loosely based on Gong and Syverson’s (1995) fail-stop protocol.

Gong and Syverson state that “A fail-stop protocol automatically halts when there is any derivation from the designed protocol execution path.” (Gong and Syverson 1995, p2). The user’s request protocol conforms to Gong and Syverson’s Definition 1 (Fail-Stop Protocol) (Gong and Syverson 1995, p3) because it returns nothing if the request is invalid. However, this is the only conformance because it uses weak not strong authentication.

Eronen and Moore, Voelker, and Savage report that attackers will often forge or “spoo” the source IP address so that they can not be traced. Therefore, the source IP address must be checked.

Three different levels of checks can be performed:

1. Does the source IP address exist in the HTTP request? Some surfers forcibly remove their address from their request. As the user’s request protocol always sends the address, any HTTP request received without an address must be an attack.
2. Is the IP address valid? It is very difficult for the recipient to check the validity of an IP address. It can check it is in the right range but it can not check that it equates to a valid location. The attacker probably selected the address at random. The location may not exist or it may be the address

of an innocent third party. Hence, this check is fallible and should not be used.

3. Is the IP address trusted? The Assistant could hold a list of the Manager IP addresses. Whenever, it received an HTTP request, it could validate the source IP address received against the list of Manager IP addresses. If there was no match, the HTTP request must be an attack. Unfortunately, it could also mean that a valid request was received from a Manager but the list has not been updated yet. The validity checks consume time, CPU, and memory resources. The amount of resources used depends on the number of Managers and the position of the received address in the list. Therefore, this check should be used as a last resort.

Meadows and Eronen identify that alternative techniques are required “to prevent attacks which employ IP spoofing.” (Eronen 2001, p4). Meadows proposes authentication whilst Eronen proposes cookies.

The HTTP request could contain an additional key value pair, which is authenticated by the Assistant. If the Assistant detected an invalid key value pair, the request must be an attack. This approach is not appropriate to the GET method because the security measure could be logged and sent in subsequent attack requests.

The website for the Password Hurler Protection (www.passwordhurlerprotection.com) states that “[it] stops brute force attacks on your web site... [it] works by logging the IP address of all failed logins (401 Errors) and then it blocks users based upon the number of failed logins within a specific period of time.”

This level of protection is valid for many commercial web sites and may be appropriate for telecommunication equipment. In the case of the Assistant, the alternative approach of validating known Manager IP addresses and blocking all other addresses should consume fewer resources than logging and blocking failed logins especially if the failed logins are mounting a distributed denial of service attack.

4. DEVELOPMENT OF CGI SCRIPT INTERFACE

There are a number of alternative CGI approaches. This section provides a comparative survey of the different approaches and considers how appropriate they are to the proposed solution.

Shah and Darugar (1998), Venkitachalam and Chiueh (1999), Wu, Wang, and Wilkins (2000) and Dumitrescu (1998) all identify that CGI has an inherent performance problem because separate processes are created to handle each client request.

They all highlight the overheads incurred in forking a new process.

New approaches have been developed to overcome the performance problems. Some are proprietary Server APIs (e.g., mod_perl) whilst others are modifications to the CGI execution architecture (e.g., FastCGI, LibCGI, and VEP).

Mod_perl (<http://perl.apache.org/guide>) brings together the PERL application and the Apache web server into one process.

FastCGI is described by Venkitachalam and Chiueh and on the FastCGI web site (<http://www.fastcgi.com>, Brown 1996a, Brown 1996b & Open Market 1996). It runs as a persistent process thereby eliminating the overheads of creating a new process.

Venkitachalam and Chiueh advocate a high-performance CGI architecture, LibCGI. The CGI script is compiled into a shared library that executes in the web server’s address space. It avoids the overhead of executing the forked process.

Shah and Darugar cite a high performance architecture using Binary Evolution’s VelociGen™ interface (see Shah and Darugar 1998, p2). They describe VelociGenforPerl™ (VEP) which “combines the performance associated with server APIs with the benefits of CGI.” (Shah and Darygar 1998, p2).

Table 1 shows a comparative summary analysis of the five approaches. The information has been extracted from the referenced papers.

Attributes	CGI	Mod_perl	FastCGI	libCGI	VEP	Server API
Performance	Poor	Fast	Fast	Fast	Fast	Fast
Separate Isolated Process	Yes	No	Yes	No	Yes	No
Persistent Process	No	Yes	Yes	No (Shared library)	Yes	Yes
Language Independence	Yes (Perl, C, C++)	No (Only Perl)	Yes	No (Only C, C++)	No (Only Perl)	No (Usually C, C++)
Proprietary	No	Yes	No	Yes	Yes	Yes
Architecture Independence	Yes	No	Yes	No	Yes	No (Same as web server)
Ease of Use	Simple	Simple (Can run existing Perl scripts)	Simple (Easy migration from CGI)	Simple (Similar to conventional CGI)	Simple (Can run existing Perl scripts)	Usually Complex

Table 1 - Comparative Summary Analysis of CGI Approaches

Two of the important attributes in the above table which significantly effect performance and security are separate isolated process and persistent process.

With a separate isolated process, a CGI based application crash will not bring down the entire web server. If they shared the same process space, the application can corrupt, crash, or compromise the web server. The application could even access the session keys for the encryption. Venkitachalam and Chiueh describe LibCGIs solution to sharing the same process space so that the application does not corrupt, crash, or compromise the web server.

Persistent processes do not die when they have finished handling a request. Instead, they wait around for a new request.

Venkitachalam and Chiueh compare LibCGI with two alternative solutions, FastCGI, and mod_perl. They conclude "LibCGI improves the CGI script execution throughput over FastCGI by a factor of 2.3, and over conventional CGI model by a factor of 3.9 to 4.6." They compared performance at the machine level and across the network.

Kothari and Claypool (1999) also measured and analysed the performance of CGI and FastCGI for input data size, output data size, disk read, disk write, and computation. They found that "CGI and Fast CGI perform effectively the same under most low-level benchmarks."

A Technical White Paper on FastCGI (Open Market 1996) compared FastCGI with CGI and concluded that FastCGI was 5 times faster than CGI.

Shah and Darugar compared VEP with CGI and concluded that VEP was up to 20 times faster than CGI.

In contrast, Wu, Wang, and Wilkins conclude that "CGI solutions are appropriate for small applications with a limited amount of client access ... with the trade-off being the performance penalty." (Wu, Wang, and Wilkins 2000, p10)

This implies that CGI can be used to control remote monitoring (i.e. starting and stopping) because the requests occur very infrequently. Typically, there will be one request to start the monitoring and another some time later to stop it.

The High Performance Common Gateway Interface Invocation paper by Venkitachalam and Chiueh covering CGI performance problems, LibCGI, FastCGI, and mod_perl were evaluated.

The conclusions were;

- There are recognised performance problems with CGI,
- New approaches have been developed that overcome these problems,

- The new approaches are not appropriate to the control of remote collection of cordless statistics. As there are minimal requests between the Manager and the Assistant, any performance problems with CGI are not seen as an issue.

- Therefore, CGI will be used to convey the user's request from the Manager to the Assistant.

The CGI script interface only runs on the ADP (see Figure 2). The operating system on the ADP is UnixWare. The UnixWare operating system provides a comprehensive environment with many shells, commands, functions, and tools.

The environment influences the form (e.g. executable or shell script) and choice of the programming language (e.g. C, C++, or Perl) for the CGI program.

An important aspect of the control using client-server technology proposal is the barrier between the CGI script interface and the rest of the remote collection system. The barrier is achieved through the creation and deletion of a control file.

Two Bourne shell scripts are used; one to create the control file and the other to delete it. The Bourne shell scripts were used in preference to an executable because when they are called, a new process is not created so there are no additional overheads.

This leaves the choice of the programming language for the CGI program.

Gundavaram states that "Perl is by far the most widely used language for CGI programming!" (1996, p11). He cites one of the advantages of Perl as "It makes calling shell commands very easy, and provides some useful equivalents of certain UNIX system functions." (1996, p11).

With this recommendation and the two Bourne shell scripts already developed, Perl was the obvious choice for the CGI program especially as Gundavaram (1996, p65) has a standard CGI PERL script that can be used. The only additions required are the recognition of the key-value pairs and the calling of the Bourne shell scripts.

5. RESULTS AND DISCUSSION

With the controlling of the remote collection system, two separate functional tests are required to test the CGI script interface; one to start the collection and the other to stop it. In both cases, the URL is invoked and the correct operation was tested and checked. The start request created the control file whilst, the stop request deleted it.

The tests were successful. They clearly demonstrated that the cordless collection could be remotely started and stopped across the Siemens

network. The switch was located in the lab and the testing was carried out from a PC in the office.

The CGI invocations using the GET method were easily tested using the Internet Explorer browser to invoke the URLs from the address line. As this approach does not work for the POST method, alternative approaches were investigated. A free command line tool, cURL (<http://curl.haxx.se>), was found which can transfer files with URL syntax. It supports many different aspects of client-server technology. It was successfully used to test the normal and error behaviour of the CGI script interface.

Although the tests were successful, they did identify security issues with the simple CGI script interface. It minimised attacks but did not prevent unauthorised access. Hence, a more secure CGI invocation is required for telecommunication equipment.

The following paragraphs outline the principles of secure CGI within the context of conveying user's requests from the Manager to the Assistant.

First, all communication between the Manager and the Assistant uses HTTPS (HTTP over SSL) rather than the standard HTTP. Therefore, all information exchanged between the Manager and the Assistant is encrypted. This includes the header, URL, posted data, and any cookies. The name of the server is not encrypted because it is used to route the request. Encryption does not stop any system the request travels through from seeing the information; it just makes it difficult for them to decode. With HTTPS and no GUI, the security measures can be placed in the URL and/or the posted data.

Secondly, authentication and/or cookies are required to distinguish the authorised accesses from the unauthorised ones.

With secure CGI invocation, the user is expected to login before the URL is invoked. If the user attempts to invoke a URL before they have logged in, they are automatically redirected to a login page. When they have successfully logged in, the original URL is automatically invoked. Therefore, the Assistant must be able to determine if the user is already logged in.

There are two possibilities; the user name is sent in every request and the server checks that the particular user has already logged in, or a cookie is sent in every request after the user has logged in. As the initial request has no cookie, the automatic redirection to the login page occurs. When the user has successfully logged in, the server puts a cookie onto the client, which is returned in subsequent requests.

As the cookie is a simpler and more efficient approach than searching for logged on user names, they are used in this invocation.

Unfortunately, with the user's requests, there is no browser or logged on user, there is only an executable running on the Manager invoking a URL on the Assistant. The executable could detect the login page and login but the user name and password would have to be hard coded or easily available. This presents a number of security problems.

For example, hard coded passwords can be easily detected and difficult to change. As the password should be changed on a regular basis to avoid misuse, hard coded passwords should not be used.

Therefore, an alternative approach to user name and password is required.

The Photuris Specification (RFC 2522) outlines some basic requirements for cookie generation. They include:

1. The cookie MUST depend on the specific parties ...
2. It MUST NOT be possible for anyone other than the issuing entity to generate cookies that will be accepted by that entity. This implies that the issuing entity will use local secret information in the generation and subsequent verification of a cookie. ...
3. The cookie generation and verification methods MUST be fast to thwart attacks ...” (Karn and Simpson 1999, p19).

These requirements can be adapted to allow a Manager to open a session to the Assistant. Figure 5 shows a secure CGI invocation between the Manager and the Assistant.

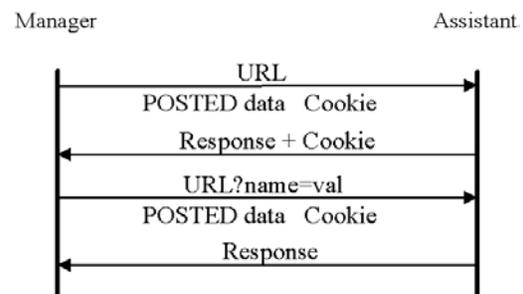


Figure 5 - Secure CGI Invocation between Manager and Assistant

The initial request can include a cookie, which identifies the Manager, the Assistant, and type of request. The Assistant can use these details to verify that the request has come from a Manager. Subsequently, the Assistant can return a cookie,

which is sent in subsequent requests from the Manager.

Therefore, Managers and Assistants can easily distinguish between valid requests and responses, and can determine unauthorised access and potential attacks by the existence or not of valid cookies. If an invalid cookie is detected, it must be an unauthorised access or attack and the request is simply ignored.

The tests were successfully repeated using the secure CGI invocations. The Manager detected and ignored unauthorised accesses.

6. CONCLUSIONS

This paper highlighted limitations with local monitoring and found that remote monitoring was a viable alternative.

This paper explored two examples of remote monitoring; network management and rail transportation. These examples were chosen because they are monitoring real time systems, which have similar characteristics to telecommunication equipment. The examples identified a number of underlying technologies that could be used for the collection of cordless statistics; agents, HTTP with CGI, Java RMI, and HTTP with XML.

To protect customers' investment, the remote collection solution had to work on both old and new Hicom switches. Some technologies (e.g. agents and Java RMI) had to be discarded because they were not available on old switches. Other technologies (e.g., HTTP with XML) were not suitable because of large memory footprints or performance problems.

The chosen underlying technology was HTTP with CGI. The remote collection solution did not use CGI in the normal way to return the collected statistics in a web page. Instead, it used it to control the starting and stopping of the cordless statistics. No feedback was given in order to confuse any potential hackers.

This paper investigated the performance problems of CGI including surveying alternative CGI. It concluded that any performance problems were not an issue because there are minimal CGI requests when CGI is used to control.

This paper recognised that CGI is not a secure technology. It investigated alternative CGI invocations with different security measures that included verifying the source IP address, using HTTPS, adding key value pairs and cookies to distinguish between valid and invalid requests between Managers and Assistants.

Finally, the goals of this paper were met as the proposed solution was adopted in the Siemens HiPath 4000 Administration and Service Product to control the collection of the cordless statistics from switches.

ABBREVIATIONS

The following abbreviations have been used in this paper:

- A&S - Administration and Service
- ADP - Administration Data Processor
- CGI - Common Gateway Interface
- CPU - Central Processing Unit
- GUI - Graphical User Interface
- HTTP - HyperText Transfer Protocol
- HTTPS - HyperText Transfer Protocol Secure
- ICN - Information and Communication Networks
- IP - Internet Protocol
- PSTN - Public Switch Telecommunication Network
- RFC - Request For Comment
- RMI - Remote Method Invocation
- SLC - Subscriber Line Cordless
- SSL - Secure Sockets Layer
- URL - Uniform Resource Locator
- XML - Extensible Markup Language

ACKNOWLEDGMENTS

I would like to thank Siemens ICN and especially my immediate management; Mr. Roger Andrews, Mr. Paul Erckens, Mr. Jeff Conway, and Mr. Graham Underwood for giving me the opportunity, time, and support to do the MSc and this paper.

Finally, I owe a lot of gratitude and thanks to my wife Mrs. Janet Simmer and children David and Andrew for their love, support, and encouragement whilst doing this paper.

AUTHOR



John Simner is a senior software engineer in Siemens' Design Services at Nottingham, U.K.. He graduated from the University of Birmingham in 1978 with a BSc with Honours Class I in Electronic and Electrical Engineering. He has worked in the Telecommunication Industry for over 25 years, working on real-time embedded and application software in C, C++, and Java. Currently, he is part of a team enhancing a web-

based administration and service (A&S) product developed by Siemens Information and Communication Networks. He was part of the first cohort on a MSc course set up between NTU and Roger Andrews Siemens' Head of Engineering. This paper is taken from his MSc. project which developed an application that remotely collected Cordless Telecommunication statistics from HiPath 4000 telecommunication equipment.

REFERENCES

BROWN, M.R., 1996a. **FastCGI: A High-Performance Gateway Interface**. Open Market, Inc. <<http://www.fastcgi.com/devkit/doc/www5-api-workshop.htm>> (3 July 2002)

BROWN, M.R., 1996b. **Understanding FastCGI Application Performance**. Open Market, Inc. <<http://www.fastcgi.com/devkit/doc/fcgi-perf.htm>> (3 July 2002)

DUMITRESCU, R.A., 1998. **Two-stage Programming via the Client-Servlet-Coprocess Interaction Model**. University of Basel, Switzerland.
(Source <http://citeseer.nj.nec.com/77511.html>)
Cached: PDF, 12 June 2002)

ERONEN, P., 2001. **Denial of service in public key protocols**. Helsinki University of Technology.
(Source <http://citeseer.nj.nec.com/eronen01denial.html>)
Cached: PDF, 11 May 2002)

FABRI, A., NIEVA, T. & UMILACCHI, P., 1999. **Use of the Internet for Remote Train Monitoring and Control: the ROSIN Project**. Paper appeared in the Proceedings of Rail Technology '99, London, September 1999.
(Available <http://icawww.epfl.ch/nieva/thesis/Conferences/RailTech99/article/RailTech99.pdf>)

GAVALAS, D., GREENWOOD, D., GHANBARI, M. & O'MAHONY, M., 2000. **Advanced Network Monitoring Applications Based on Mobile/Intelligent Agent Technology**. University of Essex, Colchester, UK & Fujitsu Telecommunications Europe Ltd., UK.
(Source <http://citeseer.nj.nec.com/268291.html>)
Cached: PDF, 18 July 2002)

GONG, L. & SYVERSON, P., 1995. **Fail-Stop Protocols: An Approach to Designing Secure**

Protocols. SRI international, Menlo Park, California.
Paper to appear in Proceedings of IFIP DCCA-5, Illinois, September 1995.
(Source <http://citeseer.nj.nec.com/49099.html>)
Cached: PDF, 11 May 2002)

GUNDAVARAM, S., 1996. **CGI Programming on the World Wide Web**. 1st ed. Sebastopol, CA: O'Reilly & Associates, Inc.

KARGL, F., MAIER, J. & WEBER, M., 2001. **Protecting Web Servers from Distributed Denial of Service Attacks**. University of Ulm, Germany.
(Source <http://citeseer.nj.nec.com/444367.html>)
Cached: PDF, 11 May 2002)

KARN, P. & SIMPSON, W., 1999. **Photuris: Session-key Management Protocol**. Network Working Group, Request for Comments 2522 (RFC 2522), Category: Experimental.
(Source <http://rfc.sunsite.dk/rfc/rfc2522.html>, 8 September 2002)

KOOIJMAN, R., 1995. **Divide and conquer in network management using event-driven network area agents**.
(Source <http://citeseer.nj.nec.com/Kooijman95divide.html>)
Cached: PDF, 18 July 2002)

KOTHARI, B. & CLAYPOOL, M., 1999. **Performance Analysis of Dynamic Web Page Generation Technologies**. Computer Science Technical Report Series. WPI-CS-TR-99-12 Worcester Polytechnic Institute, Massachusetts.
(Source <http://citeseer.nj.nec.com/119628.html>)
Cached: PDF, 12 June 2002)

MEADOWS, C., 2000a. **A Cost-Based Framework for Analysis of Denial of Service in Networks**. Naval Research Laboratory, Washington, DC 20375.
(Source <http://citeseer.nj.nec.com/375643.html>)
Cached: PDF, 11 May 2002)

MEADOWS, C., 2000b. **A Framework for Denial of Service Analysis**. Naval Research Laboratory, Washington, DC 20375.
(Source <http://citeseer.nj.nec.com/484887.html>)
Cached: PDF, 11 May 2002)

mod_perl guide.
<<http://perl.apache.org/guide/intro.htm>> (3 July 2002)

MOORE, D., VOELKER, G.M. & SAVAGE, S. 2001. **Inferring Internet Denial-of-Service Activity**. University of California, San Diego. (Source <http://citeseer.nj.nec.com/moore01inferring.html> Cached: PDF, 11 May 2002)

NIEVA, T., FABRI, A. & WEGMANN, A., 2001. **Remote Monitoring of Railway Equipment using Internet Technologies**. Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland & ABB Corporate Research Ltd., Baden, Switzerland. (Available http://icawww.epfl.ch/nieva/thesis/TechnicalReports/RMREIT/TR01_018.pdf)

Open Market, Inc., 1996, Technical White Paper. **Fast CGI: A High-Performance Web Server Interface**. <<http://www.fastcgi.com/devkit/doc/fastcgi-whitepaper/fastcgi.htm>> (3 July 2002)

SHAH, A. & DARGAR T., 1998. **Creating High Performance Web Applications Using Perl, Display Templates, XML, and Database Content**. Binary Evolution, Inc. (Source <http://citeseer.nj.nec.com/112243.html> Cached: PDF, 11 June 2002)

VENKITACHALAM, G. & CHIUEH, T., 1999. **High Performance Common Gateway Interface Invocation**. State University of New York at Stony Brook, Stony Brook, NY. (Source <http://citeseer.nj.nec.com/77638.html> Cached: PDF, 11 June 2002)

WU, A.W., WANG, H. & WILKINS, D., 2000. **Performance Comparison of Alternative Solutions For Web-To-Database Applications**. Proceedings of the Southern Conference on Computing. The University of Southern Mississippi, October 26-28, 2000. (Source <http://citeseer.nj.nec.com/428587.html> Cached: PDF, 11 June 2002)

CASE STUDY OF 100% TEST COVERAGE

J.C. SIMNER*, J. CONWAY*, T. OSMAN** and D. AL-DABASS**

* Siemens Communications
Technology Drive, Beeston,
Nottingham, NG9 1LA.
john.simner@siemens.com

** School of Computing & Mathematics
The Nottingham Trent University
Nottingham, NG1 4BU.
taha.osman@ntu.ac.uk

Abstract: Telecommunication software is expected to have a long lifespan during which many developers will add new features, modify existing features, or correct bugs. The software must be understandable, reliable, and maintainable otherwise the additions and modifications will take longer to develop and introduce further errors. Siemens has a software development process, which includes Fagan inspections, module testing, integration testing, internal, and customer field trials. The Quality Plan for a large software development stated that the developed software will be subject to either “100% code reviews and normal levels of testing” or “No code reviews and 100% testing”. This paper tries to determine whether “100% testing with no code reviews” is a viable alternative to “100% code reviews with testing” for commercial software products. It provides a case study of a recently developed application that was subject to “100% testing and no code reviews”. It uses a commercial tool to demonstrate the test coverage.

Keywords: quality assurance, telecom software testing.

1. INTRODUCTION

Telecommunications plays an important role in any company’s business. Modern private telecommunication equipment provides an extensive range of features which allows companies to manage their business more efficiently and effectively. As their business, organisation, or needs evolve, it is imperative that the configuration of the installed equipment and network is changed to meet the new circumstances otherwise, the efficiency and effectiveness of their business may be less than it should be.

Over the last two years, Siemens Information and Communication Networks have developed an enhanced Administration and Service (A&S) product which allows easier access to the configuration and performance data on Siemens’ range of telecommunication equipment. Figure 1 shows an overview of the A&S product (known as “HiPath 4000 Manager”).

The HiPath 4000 Manager was developed by a multi-national and multi-site team, involving 800 people worldwide, including Beeston, Munich, Berlin, Graz, and Boca Raton.

The goal of this paper is to provide a case study of a recently developed application that was subject to 100% testing and no code reviews.

One of the applications developed for the HiPath 4000 Manager (PmAmoProc) was chosen to be subject to 100% testing and no code reviews. It is a standalone application that runs on the HiPath 4000. It periodically invokes collection commands, parses their output, and generates statistic reports,

which are collected and handled by two other applications on the HiPath 4000 Manager.

Section 2 reviews software inspection and testing techniques to determine whether “100% testing with no code reviews” or “100% code reviews with testing” are viable alternatives.

Section 3 identifies a testing strategy that should achieve 100% test coverage using both black-box and white-box testing approaches.

Section 4 uses the same to dynamically analyse the product and determine what percentage of the product was actually tested. Initially, it identified a high percentage of untested paths. The reasons for the untested paths were determined. Additional tests were carried out until there was no further increase in test coverage. 100% test coverage was not achieved because some exception handling code could not be executed. As this situation is not unique to this particular product, an additional utility is available to mark the exception branches. This effectively removes them from the metrics thereby achieving higher overall test coverage.

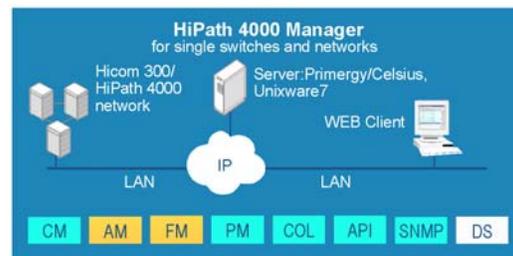


Figure 1 – HiPath 4000 Manager Product Overview

Section 4 concludes with a fault analysis of the errors found during testing and field trials to determine whether the decision made to perform no code reviews was correct or not.

Section 5 summarises how well the chosen application met its goal and demonstrated reliability and maintainability. It identifies that no code reviews and 100% testing leads to coding errors being detected in field trials that should have been found earlier.

2. PRODUCT QUALITY

Telecommunication software is expected to have a long lifespan during which many developers will add new features, modify existing features, or correct bugs. The software must be understandable, reliable, and maintainable otherwise the additions and modifications will take longer to develop and introduce further errors.

Siemens has a software development process, which includes Fagan inspections, module testing, integration testing, internal, and customer field trials. The Quality Plan for the HiPath 4000 Manager states that the developed software will be subject to either “100% code reviews and normal levels of testing” or “100% testing and no code reviews”. This section explores how realistic this is.

Table 1 shows the actual code review statistics for three modules in the HiPath 4000 Manager. In summary 1454 lines of code were reviewed in 3 hours consuming 16¼ person hours of effort and finding 16 defects.

Module Name	NLOC	Defects Found	Inspection Duration (mins)	Preparation Effort (pmins)	Execution Effort (pmins)
DNS	153	2	30	60	90
Server	159	1	20	45	60
LanCard	1142	13	130	330	390
Totals	1454	16	180	435	540

Table 1 - Code Review Statistics

Laitenberger and DeBaud (1998) carried out a survey of Software Inspection Technologies. He cites the work of Ackerman et al. (see Laitenberger and DeBaud 1998, p19) which reports the individual preparation and meeting time, per thousand lines of code, for code reviews, by two different development groups; 7.9 and 4.4, and 4.91 and 3.32.

The figures for HiPath 4000 Manager are 1.66 and 2.62. This shows that the review rate on HiPath 4000 Manager compares favourable with other organisations whilst the preparation rate is

substantially higher. The difference may be due to knowledge, familiarity, or complexity of the code.

In the HiPath 4000 Manager, there are approximately 14000 modules with 3.5 million lines of code. At the above review rate, it would take 20 person years of effort to review all the code (assuming an 8 hour day and 20 working days per month).

Laitenberger and DeBaud report that “part of the problem [with software inspections] is the perception that ... [they] cost more than they are worth.” (Laitenberger and DeBaud 1998, p21).

For example, is it worth spending 20 person years (@ £85K per person per year, total £1.7 million) inspecting 3.5 million lines of code. Expressed in this way the general answer is no. It is not economical or viable to do so.

So, is 100% testing more realistic?

Rushby (1991), Watson (1996), and Watson and McCabe (1996) all describe different testing methodologies. They include; random, regression, thorough, and functional.

Watson and McCabe identify that “[a] common approach to testing is based on requirements analysis. A requirements specification is converted into test cases, which are then executed ... “ (Watson and McCabe 1996, p2).

This is a very easy approach to adopt. If the software has been analysed and designed to meet the requirements then executing the test cases will fully test the software. Watson and McCabe identify that the requirements are usually at a higher level than the code so a lot of the code will not be tested.

Therefore, a lower level approach must be adopted. The code is inspected and a set of test cases is derived that test each and every statement, line, branch, variable, or path, through the code. This is a manual method, which is very time consuming and error prone.

Watson outlines the different testing criteria including; statement testing, code coverage, branch testing, data flow testing, and structured testing.

Watson states that “Structured testing, also known as basic path testing, is a methodology for software module testing based on the cyclomatic complexity measure of McCabe.” (Watson 1996, piii and p1).

The cyclomatic complexity measures the logical complexity of a module.

Watson and McCabe state that “it gives the number of recommended tests for software.” (Watson and McCabe 1996, p7). A module’s cyclomatic complexity is the minimum number of tests

required to fully test the module. It is a theoretical value, which may not be achievable in practice.

Watson and Watson and McCabe both define and characterise cyclomatic complexity.

Watson and McCabe cite the work of McCabe stating that “Structured testing is more theoretically rigorous and more effective at detecting errors in practice than other common test coverage criteria such as statement and branch coverage.” (Watson and McCabe 1996, p31).

Watson identifies an automated approach to structured testing in which the source code is instrumented and writes a trace file of its execution. The McCabe INTEGRATED QUALITY™ toolset is a commercial tool that automatically instruments the source code and analyses the resultant trace file. The tool reports code, branch, and complexity coverage.

There are very few papers on structured testing and the McCabe toolset. There are many papers on metrics, some of which question the usefulness and theoretical foundations of the cyclomatic complexity metric. In their defence, Watson and McCabe present many case studies that report successes with cyclomatic complexity.

Finally, there is general consensus that 100% testing is feasible using an automated tool to record test coverage. However, it is still a high risk strategy to perform 100% testing in preference to code inspections especially as Laitenberger and DeBaud reports that “available quantitative evidence [between 19-93% of all defects were detected by inspections] ... indicates that inspections have had significant positive impact on the quality of the developed software and that inspections are more cost-effective than other defect detection activities, such as testing.” (Laitenberger and DeBaud 1998, p21).

3. TESTING STRATEGY

The case study adopted a two stage testing strategy. A black-box testing approach was carried out in the first stage with a white-box testing approach in the second stage. These two approaches are not alternatives; they are complimentary. They are normally applied at different stages in the development of the code. This particular testing strategy was adopted to see how much test coverage was achieved by each approach.

Black-box testing tests the functionality of the software at its interfaces. The tests are usually performed at the end of the coding stage. They check that the software meets its requirements. White-box testing (which includes basic path and control structure testing techniques) tests the code from a design perspective.

With knowledge of the code and its data structures, a set of test cases can be derived that test the individual paths, controls, and data within the code. These tests can be performed whilst the code is being developed. Experience has shown that most errors are logical ones. They occur through incorrect equality tests (e.g. $a < b$, $a \leq b$, etc.) and incorrect boundary conditions on loops. These errors are easily detected by white-box testing.

With white-box testing, 100% test coverage should be achieved but it takes a long time. Whereas, with black-box testing, all of the functionality can be tested in a reasonable time frame but 100% test coverage may not be achieved. The correct testing strategy will use the best combination of both approaches to find the maximum number of errors in the minimum time whilst ensuring maximum test coverage.

The individual test cases are described in Appendix A.

MCCABE DYNAMIC ANALYSIS OF PMAMOPROC APPLICATION

A commercial tool, McCabe INTEGRATED QUALITY™ Test tool was used to measure and assess the effectiveness of the testing strategy. The tool automatically instruments the source code so that it can identify what has been covered. It supports a number of different module testing methodologies; structured testing, design path coverage, branch coverage, slice coverage, and Boolean coverage (McCabe 2001, p116).

Path, code, and branch coverage are all used to assess how well the PmAmoProc application is tested using the test cases defined in appendix A.

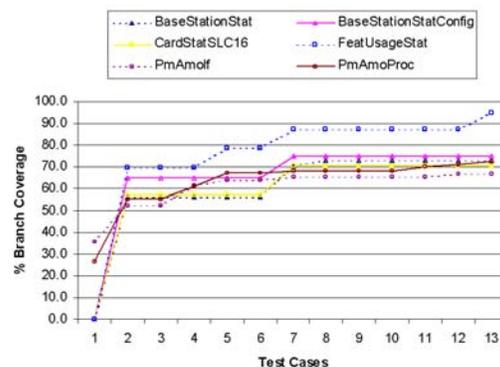


Figure 2 - Class Coverage Metrics

Figure 2 test cases 1 and 2 show the branch coverage for the six classes in PmAmoProc. It clearly shows that only 52.4% to 69.6% branch coverage has been achieved by black-box testing. This confirms the Watson and McCabe (1996)

finding that test cases based only on requirements do not test most of the code.

The McCabe INTEGRATED QUALITY™ Test tool was used to identify untested paths, determine why they were not tested, and derive suitable test cases.

The reasons for the untested paths were as follows:

1. **High Level Requirements** - Watson and McCabe identified that requirements may be at higher level than the code so code inspection may be required to generate test cases. This approach was used to generate the latter test cases listed in appendix A.

2. **Boundary Checks** – All parameters received from the command line or external modules should be range checked/validated. With embedded software and no debugging environment, it may be difficult to generate out of range values from external modules.

3. **Test Code** – There are many different approaches to testing. One approach is to write all the code then test it whilst an alternative approach is to write some of the code, test it, then write some more. To perform incremental testing, test stubs or test options are added but rarely removed. The additional code, tested or not, affects the test coverage metrics.

4. **Trace Code** – It can be very difficult to debug embedded software on target hardware so trace statements are often added. As they always effect performance, they must be enabled or disabled through compiler options or control files on the target hardware. A control file is the preferred option because it allows tracing to be turned on and off in the field whilst the compiler option requires a new version to be built, distributed and installed. The software is often tested with tracing enabled. It must also be tested with tracing disabled to ensure maximum test coverage.

5. **Unreachable Code** – Sometimes modules (or code) are written that can never be reached. For example, a class must always have a default constructor. If the class also has a non-default constructor, the default constructor may never be called. The additional code effects the test coverage metrics.

6. **Error/Exception Handling** – Embedded software must detect and handle all error conditions (or exceptions) to ensure that the software continues to run and does not cause any problems. For example, if the application writes to a file, it must always check that the file has been successfully opened before it writes to it. If it fails to open the file, it should log an error rather than write to it. It is very difficult to test system errors and exceptions.

7. **Semaphore/Control Files** – With multi-threaded and multi-process applications, there is controlled access to shared resources. With embedded software and no debugging environment, this may be difficult to test.

8. **Redundant Code** – Sometimes modules are written but never called. The additional code effects the test coverage metrics. They should always be removed.

9. **Coding Error** – Sometimes coding errors occur that results in unreachable code, which is not detected by the compiler. For example, in PmAmoProc, a class instance is explicitly created but never deleted. This coding error should have been picked up at code review. It effects the test coverage metrics. It may cause memory leaks at run time.

The derived test cases and their class coverage metrics are listed in appendix A. Figure 2 shows a graph of branch coverage versus the test cases for the six classes in PmAmoProc. Branch coverage was chosen because it gave the highest coverage value. However, Figure 2 clearly shows that only 66.9% to 94.9% branch coverage has been achieved with the derived test cases. The discrepancy is due to boundary checks, unreachable code, and error/exception handling.

Table 2 lists the path, code, and branch coverage for the least tested modules after completing all of the test cases listed in appendix A.

Module Name	% Coverage			
	v(G)	iv(G)	Lines	Branches
BaseStationStatConfig--BaseStationStatConfig	0	0	0	0
PmAmoMf.checkAmoSuccessfullyCompleted	25	28.6	34.6	46.7
PmAmoMf.convertAndStripSpaces	25	33.3	100	57.1
PmAmoProc.refreshSLC16details	18.2	18.2	100	61.9
PmAmoMf.setupMpsidInterfaceDirect	33.3	33.3	29.7	63.6
PmAmoMf.setupFamosSession	22.2	25	66.7	64.7
PmAmoMf.getBaseStationStatFromRegZcdOutput	0	0	83.9	69.6
PmAmoMf.getSLC16CardStatFromRegZcdOutput	0	0	89.5	72.7
PmAmoMf.getDateFeatStatsFromDisZausdOutput	0	0	85.7	75
PmAmoMf.getSLC16CardPositionsFromDisBcsuOutput	0	0	90.9	90.9
PmAmoMf.getStationNumbersFromDisSdsuOutput	0	0	93.6	91.3

Table 2 - McCabe Dynamic Analysis – Least Tested Modules

A module’s cyclomatic complexity (v(G)) is the minimum number of tests required to fully test the module. It is a theoretical value, which may not be achievable in practice (see McCabe 2001, p51-53).

A module’s design complexity (iv(G)) is the number of paths with calls to other modules (see McCabe 2001, p55-57). If there are no calls to other modules, the design complexity is zero and can not be tested.

Line coverage shows how many source code statements were executed. This does not include any comments or blank lines, as they can never be tested.

Branch coverage shows how many of the exits from branches were executed. For example, an 'if ... then' has two exits, one if the condition is true and one if the condition is false. A 'if ... then ... else' also has the same two exits.

An analysis of the modules listed in Table 2 revealed:

- **1 Coding Error** – For example, an instance of BaseStationStatConfig was created by PmAmoProc and never deleted.
- **4 Boundary Checks** – For example, checkAmoSuccessfullyCompleted() checked the AMO output for NOT COMPLETED, NOT EXECUTED, and EXECUTED. These conditions were not generated during the tests.
- **31 Error/Exception Handling** – For example, convertAndStripSpaces() checked for null strings and strings with no spaces. These conditions were not generated during the tests. This analysis clearly shows that it is extremely difficult to achieve 100% test coverage because of the difficulties testing exception handling code.

As this situation is not unique to these particular modules, McCabe have developed an additional utility (McCabe Exception Coverage Utility Version 1.8 20020228) that can mark the exception branches. The branch report shows the percentage of branches tested and the percentage of non-exceptional branches tested (i.e. the number of actual branches minus those marked as exceptional). This effectively removes them from the metrics thereby achieving higher overall test coverage.

Finally, Table 2 shows 5 modules with low cyclomatic and design complexity coverage but high code and branch coverage. The untested graph listings in the McCabe INTEGRATED QUALITY™ Test tool did not match the cyclomatic complexity coverage. This matter was raised with McCabe. The calculation of the cyclomatic and design complexity coverage in the test tool is designed for modules with only one entry and exit point. If the code allows early exits from the module by exiting straight out of a loop, the coverage for that module may not be recorded. There is some debate over whether it is good or bad practice to exit straight out of loops. The author believes it is sometimes permissible if it makes the overall code simpler.

4. RESULTS AND DISCUSSION

The main intention of this case study was to subject a developed application to 100% testing and not do code reviews. Section 0 identified that this is a high-risk strategy, which may not find all of the code errors. One way of determining that the strategy has worked is to carry out a fault analysis at the end of the field trial. The fault analysis determines; how many errors were detected, where they were found, their cause, and whether they should have been found earlier. The fault analysis was extended to cover the whole development cycle (i.e. development phase, module testing, integration testing, and field trials).

If the strategy has worked, none of the errors raised during the field trial will have a cause of coding error.

Table 3 shows a summary of the fault analysis for PmAmoProc.

Class	Found	Number of Occurrences	Source	Should have been found
Error	Module Testing	7	Coding Error	Yes - Code Review
Error	Module Testing	1	Coding Error	No
Error	Module Testing	1	Unclear Requirement	No
Error	Module Testing	1	Missing Requirement	Yes - Requirement Review
Error	Integration Testing	4	Missing Requirement	No
Error	Integration Testing	1	Missing Requirement	Yes - Requirement Review
Error	Integration Testing	3	Additional Requirement	Yes - Requirement Review
Error	Integration Testing	2	Requirement Error	Yes - Requirement Review
Error	Integration Testing	11	Coding Error	Yes - Code Review
Change Request	Integration Testing	9	Requirement Change	No
Change Request	Integration Testing	2	Switch Error	No
Error	Integration Testing	1	Library Error	No
Error	Integration Testing	1	Compiler Error	No
Error	Field Trial	6	Coding Error	Yes - Code Review
Error	Field Trial	1	Coding Error and Insufficient Testing	Yes - Code Review
Error	Field Trial	1	Requirement Change	No

Table 3 - Summary Fault Analysis for PmAmoProc

The fault analysis showed that 41 faults and 11 change requests were found on the PmAmoProc application.

The faults were categorised as:

- Unclear, Missing, Changed, and Additional Requirements – 22
- Coding Errors and Insufficient Testing – 26
- External Errors (i.e. Switch, Library, and Compiler) - 4

This clearly demonstrates one of the problems of developing software is the nature of the requirements. They are often unclear and change during the development.

The coding errors were examined to determine the cause of the high fault rate (21 faults/KLOC).

There were three classes, which were very similar in functionality. The code from one class was copied to the other classes and amended as appropriate. There were coding errors in the first class, which were also copied into the other classes; resulting in a higher fault rate.

The remaining coding errors were due to over zealous error reporting.

Embedded software must detect and handle all error conditions (or exceptions) to ensure that the software continues to run and does not cause any problems.

The PmAmoProc application is a good example. It is expected to automatically run on a periodic basis, collect the information, and produce the reports. There is no user intervention. If it fails to run, collect the information, or generate the reports, the historical information for that period will be lost forever.

There are two ways of developing the error or exception handling code; pre-emption or responding to crashes.

The first approach looks at the overall system, tries to determine what might cause errors, and adds exception code to handle these situations. The exception code usually reports an error and recovers from the situation. There are two potential problems with this approach; it may identify errors that are not errors, and it will never find every possible error.

The second approach waits for the system to crash during testing, identifies the cause, and adds exception code to handle the crash. Again, the exception code usually reports an error and recovers from the situation. There are also two potential problems with this approach; crashes may not occur until field trials, and it can be very difficult to identify the cause of a crash from the crash dump and any trace logs.

Hence, the first approach is recommended but there is sometimes a fine line between what is believed to be an error and what is actually an error.

Finally, the fault analysis clearly shows that the strategy of subjecting the developed product to 100% testing and not doing any code reviews did not work because seven coding errors were found during field trial that would have been found (in the author's opinion) at a code review. Finding coding errors at the later stages could delay the product as they have to be fixed, retested, and retrialled.

5. CONCLUSIONS

On reflection, it was unwise to adopt the high-risk strategy of 100% testing and not do code reviews. The fault analysis showed too many coding errors found during the latter stages of test and trial, which should have been found earlier.

Therefore, code reviews and testing should both be carried out. They both should be used for their strengths. Code reviews can check for typical coding errors and understand what the code is trying to do. Whereas, testing can check that a product meets its requirements and does what it should do.

The metrics can be used to identify the risks and take appropriate action. For example, any complex code or code that has not been tested, should be reviewed. Likewise, any code that has not been reviewed should be 100% tested.

To be effective, code reviews should involve diligent software engineers with detailed knowledge of the product, the requirements, and available libraries, and general background knowledge of software, and the general subject area.

The testing clearly showed how extremely difficult it was to obtain 100% test coverage across the whole PmAmoProc application. Some modules were 100% tested but the average test coverage for lines and branches were 79.4% and 75.4, respectively. The discrepancy was mainly due to exception handling code. The additional utility to mask the execution code and report a higher coverage may satisfy 100% testing contracts but it does not solve the underlying problem shown by the fault analysis that even exception code has coding errors.

There was some concern that different metrics for the same module showed different % coverage (e.g. 100% line coverage but only 57% branch coverage, 64% branch coverage but only 30% line coverage). This shows the importance of understanding the metrics and how they are generated rather than taking them at face value. It also shows that more than one coverage metric should be used to demonstrate 100% test coverage.

There was a slight incompatibility problem between the HiPath 4000 Manager development environment and the McCabe INTEGRATED QUALITY™ Test tool environment. However, once these were overcome, the tool provided very good support for instrumenting the C++ code, exporting the instrumented code, importing the resultant output from running the instrumented code, and generating the reports.

Finally, product quality can be improved by using the metrics to focus resources on those areas that

need reviewing. The project team must decide which metrics are appropriate for their project and what level they should be limited to.

AUTHOR



John Simner is a senior software engineer in Siemens' Design Services at Nottingham, U.K.. He graduated from the University of Birmingham in 1978 with a BSc with Honours Class I in Electronic and Electrical Engineering. He has worked in the Telecommunication Industry for over 25 years, working on real-time embedded and application software in C, C++, and Java. Currently, he is part of a team enhancing a web-based administration and service (A&S) product developed by Siemens Information and Communication Networks. He was part of the first cohort on a MSc course set up between NTU and Roger Andrews Siemens' Head of Engineering. This paper is taken from his MSc. project which developed an application that remotely collected Cordless Telecommunication statistics from HiPath 4000 telecommunication equipment.

ACKNOWLEDGMENTS

I would like to thank Siemens ICN and especially my immediate management; Mr. Roger Andrews, Mr. Paul Ereckens and Mr. Graham Underwood for their support. Finally, I owe a lot of gratitude and thanks to my wife Mrs. Janet Simner and children David and Andrew for their love, support, and encouragement whilst doing this paper.

REFERENCES

The following were used as reference material for this paper:

McCABE, 2001. **Using McCabe Test, Version 7.1, Manual.** Columbia, MD: McCabe Associates.

LAITENBERGER, O. & DEBUAD, J-M, 1998. **An Encompassing Life-Cycle Centric Survey of Software Inspection (ISERN-98-32).** Fraunhofer Institute for Experimental Software Engineering, Kaiserslautern, Germany & Lucent Technologies, Naperville, IL

RUSHBY, J., 1991. **Measures and Techniques for Software Quality Assurance.** SRI International, Menlo Park, CA.

WATSON, A.H., 1996. **FastCGI Structured Testing: Analysis and Extensions.** A Dissertation presented to the Faculty of Princeton University in Candidacy for the Degree of Doctor of Philosophy.

WATSON, A.H. & McCABE, T.J., 1996.

Structured Testing: A Testing Methodology Using the Cyclomatic Complexity Metric.

National Institute of Standards and Technology, Gaithersburg, MD. NIST Special Publication 500-235.

APPENDIX A – McCabe Dynamic Analysis – Class Coverage Metrics

This appendix contains the class results of the McCabe Dynamic Analysis of the PmAmoProc application using McCabe INTEGRATED QUALITY™ toolset Version 7.1.

Table 4 to Table 9 show the class coverage metrics for each class in the PmAmoProc application for the following test cases (T1 to T13):

1. With Unity A&S Trace Tool tracing enabled and PmAmoProc tracing disabled, update the list of cordless equipment on the HiPath 4000 switch using the HiPath 4000 Manager Client GUI.
2. With Unity A&S Trace Tool tracing, cordless and feature usage collection all enabled, and PmAmoProc tracing disabled, collect the cordless and feature usage statistics from the HiPath 4000 switch.
3. With Unity A&S Trace Tool tracing enabled and PmAmoProc tracing disabled, update the list of cordless equipment on the HiPath 4000 switch using the now option on the local command line interface.
4. With PmAmoProc tracing disabled and Unity A&S Trace Tool tracing both enabled and disabled, update the list of cordless equipment on the HiPath 4000 switch using the midnight option on the local command line interface.
5. With Unity A&S Trace Tool tracing, PmAmoProc tracing, cordless and feature usage collection all disabled, collect the cordless and feature usage statistics from the HiPath 4000 switch.
6. With Unity A&S Trace Tool tracing and PmAmoProc tracing both disabled, update the list of cordless equipment on the HiPath 4000 switch using the HiPath 4000 Manager Client GUI.
7. With Unity A&S Trace Tool tracing and PmAmoProc tracing both disabled, and cordless and feature usage collection both enabled, collect the cordless and feature usage statistics from the HiPath 4000 switch.
8. With PmAmoProc tracing enabled, update the list of cordless equipment on the HiPath 4000 switch using the HiPath 4000 Manager Client GUI and collect the cordless and feature usage statistics from the HiPath 4000 switch.
9. Using the local command line interface:
 - Invoke PmAmoProc with more than two arguments,
 - Invoke PmAmoProc with an invalid argument,
 - Invoke PmAmoProc with no argument,
 - Invoke PmAmoProc after removing the unlock file,
 - Invoke PmAmoProc after creating installation control file,
 - Invoke PmAmoProc after creating startup control file,

- Invoke PmAmoProc then remove the lock file.

10. With cordless and feature usage collection both enabled, create pm_temp_base_stat.txt and ziel_intern files, and collect the cordless and feature usage statistics from the HiPath 4000 switch.

11. With cordless and feature usage collection both enabled, delete all previous statistic files and station number files the list of cordless equipment on the HiPath 4000 switch using the HiPath 4000 Manager Client GUI, and collect the cordless and feature usage statistics from the HiPath 4000 switch.

12. With cordless and feature usage collection both enabled, delete the zausl control file (PmAmoProc.zausl), and collect the cordless and feature usage statistics from the HiPath 4000 switch.

13. With cordless and feature usage collection both enabled, create the collection stop files in the file transfer directory, and collect the cordless and feature usage

statistics from the HiPath 4000 switch. After a few minutes, delete the collection stop files.

The Unity A&S Server Trace Tool is a proprietary trace tool used in the HiPath 4000 Manager. PmAmoProc tracing is a local tracing utility within the PmAmoProc application. The cordless collection and feature usage collection are controlled by local control files.

The tables include the percentage coverage for three metrics; Cyclomatic Complexity, Module Design Complexity, and branches.

A module's Cyclomatic Complexity (v(G)) is the minimum number of linearly independent paths through the module (see McCabe 2001, p51-53).

A module's Design Complexity (iv(G)) is the number of paths with calls to other modules (see McCabe 2001, p55-57).

Branch coverage shows how many of the exits from branches were executed.

Table 4 - Class Coverage Metrics for BaseStationStat													
% Coverage	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13
v(G)	0.0	36.7	36.7	36.7	36.7	36.7	59.0	62.8	62.8	62.8	62.8	62.8	62.8
iv(G)	0.0	38.0	38.0	38.0	38.0	38.0	62.8	66.7	66.7	66.7	66.7	66.7	66.7
Branches	0.0	56.2	56.2	56.2	56.2	56.2	70.4	73.0	73.0	73.0	73.0	73.0	73.0
Table 5 - Class Coverage Metrics for BaseStationStatConfig													
% Coverage	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13
v(G)	0.0	44.3	44.3	44.3	44.3	44.3	60.7	60.7	60.7	60.7	60.7	60.7	60.7
iv(G)	0.0	46.7	46.7	46.7	46.7	46.7	63.9	63.9	63.9	63.9	63.9	63.9	63.9
Branches	0.0	64.7	64.7	64.7	64.7	64.7	74.9	74.9	74.9	74.9	74.9	74.9	74.9
Table 6 - Class Coverage Metrics for CardStatSLC16													
% Coverage	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13
v(G)	0.0	34.2	34.2	34.2	34.2	34.2	55.8	55.8	55.8	55.8	55.8	55.8	55.8
iv(G)	0.0	36.5	36.5	36.5	36.5	36.5	61.4	61.4	61.4	61.4	61.4	61.4	61.4
Branches	0.0	57.2	57.2	57.2	57.2	57.2	70.7	70.7	70.7	70.7	70.7	70.7	70.7
Table 7 - Class Coverage Metrics for FeatUsageStat													
% Coverage	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13
v(G)	0.0	44.1	44.1	44.1	58.0	58.0	71.7	71.7	71.7	71.7	71.7	71.7	80.1
iv(G)	0.0	45.8	45.8	45.8	61.3	61.3	76.2	76.2	76.2	76.2	76.2	76.2	85.1
Branches	0.0	69.6	69.6	69.6	78.6	78.6	87.1	87.1	87.1	87.1	87.1	87.1	94.9
Table 8 - Class Coverage Metrics for PmAmolf													
% Coverage	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13
v(G)	11.3	14.6	14.6	24.6	26.5	26.5	27.8	27.8	27.8	27.8	27.8	28.7	28.7
iv(G)	12.4	15.6	15.6	26.2	28.2	28.2	29.5	29.5	29.5	29.5	29.5	30.4	30.4
Branches	35.6	52.4	52.4	61.6	63.8	63.8	65.4	65.4	65.4	65.4	65.4	66.9	66.9
Table 9 - Class Coverage Metrics for PmAmoProc													
% Coverage	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13
v(G)	12.6	19.5	19.5	28.8	33.9	33.9	40.8	40.8	40.8	40.8	43.4	44.4	45.2
iv(G)	12.6	19.5	19.5	28.9	33.9	33.9	40.8	40.8	40.8	40.8	43.4	44.4	45.2
Branches	26.9	55.3	55.3	60.9	67.2	67.2	68.1	68.1	68.1	68.1	70.0	71.0	72.2

TCP/IP CONNECTION MANAGEMENT USING A REAL-TIME DEVELOPMENT TOOL

ANN GRAY*, R. WHITELOCK*, E. PEYTCHEV** and D. AL-DABASS**

* Siemens Communications
Technology Drive, Beeston,
Nottingham, NG9 1LA.
ann.gray@siemens.com

** School of Computing & Mathematics
The Nottingham Trent University
Nottingham, NG1 4BU.
evtim.peytchev@ntu.ac.uk

Abstract: The Central Integration Unit (CIU) is a major component of a system being developed to provide voice and data communications between mobile radio users and fixed terminal users. The CIU has several different TCP/IP interfaces, both external and internal to the CIU, with varying characteristics. Some use permanent connections whilst others use transaction-based connections; some are clients or servers, others incorporate both client and server operation. This paper looks at the issues behind the design of the connection management aspects of the CIU and then describes the implementation of the connection management software within the CIU.

Keywords: TCP/IP, real time tools

1. INTRODUCTION

The aim of this project is to demonstrate a method of managing several different types of TCP/IP interface within a single application.

The Voice Radio System provides a means of intelligently routing calls between mobile users and fixed terminals based upon current location of the mobile user. This system uses a GSM network to provide the mobile user switching capabilities, in conjunction with a Central Integration Unit (CIU) to provide the voice and data routing capabilities. Figure 1.1 illustrates the basic functionality of the Voice Radio System.

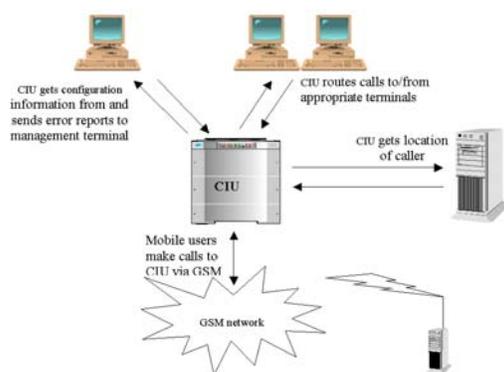


Figure 1-1 Overview of Voice Radio System

The CIU is a combined hardware and software development consisting of many Commercial off the Shelf (COTS) sub-components, including a Realitis PABX, along with bespoke CIU application software. The application's primary function is to control the routing of calls to and

from the fixed terminals, which it does using the CTI (computer telephony integration) capabilities of the PABX. In order to perform this role it uses application level communications to the other external components of the system. These communications use a proprietary system messaging protocol, which has been specified by the customer, via TCP/IP socket connections.

All of these interfaces (five in total) use uni-directional, transaction-based connections. This means that the connection is established whenever a single message is required to be sent on that interface and closed again immediately after the transmission. Three of the links are one-to-one, the others having multiple remote end-points.

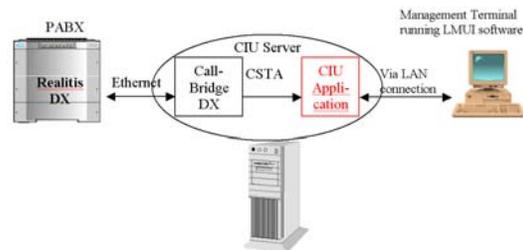


Figure 1-2 CIU Internal Interfaces

In addition to the external interfaces, the CIU application also has two further internal TCP/IP interfaces to manage, as illustrated in Figure 1.2. The first of these concerns the CTI interface to the PABX (via CallBridge DX software, an existing Siemens Communications product). This utilises CSTA, the European standard CTI API, as defined by the standards ECMA-179 and ECMA-180 (ECMA, 1992). The underlying TCP/IP connection

is permanent, one-to-one and bi-directional, with the CIU application acting as the client.

The final interface allows the CIU Local Management User Interface (LMUI), another piece of software specifically designed for the CIU development, to connect to the application for administration and configuration purposes. This is also a permanent, one-to-one, bi-directional TCP/IP connection, but in this case the CIU application acts as the server.

In order to provide the call routing capabilities required for the Voice Radio System, the CIU must be developed in such a way as to manage its multiple TCP/IP connections in the most effective manner in order to support the level of performance required by the customer. The aim of this project is to demonstrate a way of managing these different types of TCP/IP interface.

2. THE CURENT SYSTEM

For a prototype CIU development, some of the TCP/IP interfaces were developed, namely the two internal interfaces plus one of the external interfaces. The design of all three interface components was based upon the CSTA interface component (known as the CSTAIC) from the CallBridge DX software, with the following adaptations:-

1. The CIU CSTA interface component was modified to be a client, rather than a server.
2. The CIU LMUI interface component needed to be a server, therefore no modification was required.
3. The external interface component was considerably altered to handle transaction-based operations for both client (for outgoing messages) and server (for incoming messages).

Therefore the prototype software contained three different interface components, all with the same basic structure. This increased the amount of testing required, and also led to maintainability issues. Any fault that was found in one interface component had to be checked and potentially fixed in two other places. In the final CIU development this situation would be further compounded with the addition of four further external interfaces. It is not practical to maintain seven different interface components. Therefore it is desirable to have as few interface components as possible, with the ideal being a single, generic connection component.

The second concern regarding the design of the connection interfaces is whether to re-use any part

of the prototype software as the basis for the phase 5 interface components. The prototype software, including the interface components, utilises a proprietary framework, which was developed by a third party, for which Siemens Communications owned the libraries, but not the original source code. This framework provides the following capabilities: - threads, task scheduling, inter-process communication and diagnostic logging, as well as socket handling classes.

The framework was found to have several limitations. Firstly, link errors were encountered when building the CIU application using Microsoft Visual C++ 6.0. It is thought that the framework library was built using an earlier version of Visual C++ and includes versions of the Microsoft C++ libraries that are incompatible with those linked in using version 6.0. During the prototype development, reverting back to Visual C++ 5.0 solved this problem, but this is not an ideal long-term solution, since support for earlier releases of Microsoft products is not guaranteed indefinitely. Although it may be possible to acquire a version of the framework library that is compatible with Microsoft Visual C++ 6.0, this is also limiting since there is no guarantee that it would be compatible with future versions. Furthermore this may take too long to acquire and would generate an unwanted dependency on external developers.

Secondly, there were doubts within the engineering department regarding the performance of this framework under heavy traffic, although this has not been proven one way or the other. Although the prototype CIU met its required load, the traffic requirements for the final product would be much greater. The project time constraints mean that the risk is too high to invest time in using this framework only to find later that it is not effective.

The final limitation is that, since Siemens Communications do not own the source code, there is no control over its operation. Any modifications to the framework behaviour have to be made by adapting the custom software in the application to achieve the right results. This is very restrictive and, in some cases, impossible.

For all these reasons, it was decided that the framework used for the prototype would not be re-used for the main CIU development.

3. THE NEW SYSTEM

In order to determine a strategy for the design of the connection management software, several options were considered. These options had to be examined in light of a customer requirement mandating the use of Rational Rose RealTime (RRRT) as a development tool.

Rational Rose RealTime is a modelling tool which uses an extended form of UML to enable the modelling, implementation, building and debugging of complex real-time systems. The RRRT modelling language includes support for concurrent objects, allowing communication between them via ports using user-defined protocols. The dynamic behaviour of these capsules can also be modelled using state diagrams. Further information about these concepts can be found in the Rational Rose RealTime Modeling Language Guide (Rational 2002).

3.1 Pre-Existing CallBridge Framework

The majority of the CallBridge software utilises a different framework to that used for the CSTA interface. It was originally developed for CallBridge's predecessor and has been enhanced over many years into a robust and reliable platform on which to build an application. This framework has also been used in other in-house developments and is an obvious first-choice candidate for use in the CIU. It provides a round-robin task-based scheduler, including inter-task communication, plus a message transport layer that allows non-blocking TCP/IP socket management within a single-threaded application.

Although, at first, the CallBridge framework appears to be a good option, there are some disadvantages to weigh up.

1. The message transport layer currently only supports permanent connections, and would therefore require substantial modifications to allow its use for transaction-based connections.
2. Since RRRT has its own built-in mechanisms for task-scheduling and inter-task communication amongst other things, it will provide the framework for the CIU application. The only part of the CallBridge framework that is required is the message transport layer. However, the CallBridge framework is not structured in such a way as to easily extract the message transport software on its own. Effort would be required to repackage it for use within RRRT.
3. Finally, the CallBridge framework is written in 'C' and is likely to be difficult to maintain by a team whose main skills are in C++ and Java. This would not have been a major issue if the framework was suitable in its present form, but since points 1 and 2 imply a substantial amount of rework, this would add a greater risk to the project.

3.2 Single Generic Connection Manager

This option involves the development, from scratch, of a generic connection manager component, using the RRRT framework. This component would be able to handle both permanent and transaction-based connections; client or server operation, or both; and either single or multiple remote endpoints.

The advantage of this approach is that there is common code for all interfaces, so modifications only need to be made in one place. This would also reduce the time required for testing the connection management software and it carries less risk, both to the project timescales and the quality of the resulting software.

Preliminary investigations into a possible design, however, indicate that this is not straightforward, and that the resulting code could be very complex and difficult to follow.

3.3. Common External Connection Manager Template

The final option is to have a single connection manager template for all external interfaces, i.e. those requiring transaction-based connections. The two internal interfaces would then have their own customised connection managers. The customer had already used Rational Rose RealTime during the prototype development of one of the external components and made available the connection package (known as the TCP/IP Key Mechanism) for possible re-use by Siemens Communications.

As can be seen in Figure 3.1, the top-level TCP/IPHandler object is responsible for initialising the Winsock library ready for use as well as managing the incoming and outgoing connections. Outgoing messages are passed to the TCPClient, which creates the client connection, sends the message, then closes the connection down.

In order to handle incoming messages to a particular port, the TCP/IPHandler initialises the TCP/ServerController, informing it of the server port number. The TCP/ServerController then creates the listening socket and kicks off the first server thread to listen for a connection. The listening socket is a global variable so that it can be accessed by each server thread. As soon as a server thread receives a connection, the controller is informed and the next thread is set listening whilst the first receives the message.

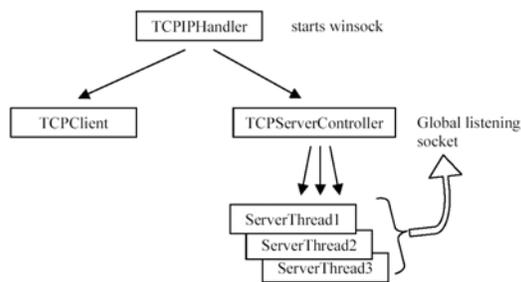


Figure 3-1 Overview of TCP/IP Key Mechanism

Whilst this connection package contains the basic classes for transmission and receipt of messages using the transaction-based sockets method, it was designed for use in an application that only requires a single TCP/IP interface whereas the CIU requires multiple such interfaces. It would therefore require some small modification in order to re-use the TCP/IP Key Mechanism in the CIU.

It was proposed that the two internal connection managers should be designed from scratch in RRRT using non-blocking sockets. The reason for use of non-blocking sockets is that for permanent bi-directional connections, the socket code needs to be able to deal with incoming and outgoing stimuli asynchronously. Using a blocking socket to wait for input would delay outgoing message transmission.

Some of the low-level socket handling code from the prototype CIU connection classes could be incorporated in the transitions and operations of the new capsules where appropriate in order to reduce the implementation time and reduce the risk.

3.4 Decision Analysis

In order to determine which of the three options is most appropriate, a Kepner Tregoe Decision Analysis was performed. This is a technique for arriving at a decision based on an analysis of the alternatives against the key objectives of the decision. The full analysis is shown in Appendix A.

There are two mandatory objectives – the solution must be capable of supporting both permanent and transaction-based connections, and it must be compatible with the RRRT and C++ development environment.

The CallBridge Framework option fails both of these criteria, and was therefore rejected outright.

The remaining objectives were weighted according to importance and the remaining two alternatives (having met the mandatory objectives) were judged on their performance against these criteria.

The Common External Connection Manager Template alternative performed better on most of the criteria and finished top overall. Therefore this option was chosen.

Despite the design goal of minimising the number of different connection managers, the decision analysis process showed that this objective was of relatively low importance compared to the issues of timescale, maintainability and risk.

4. SOFTWARE IMPLEMENTATION

In this section, the design and implementation of the software for the management of the internal connections will be described.

Rational Rose RealTime was used throughout the development for the design and implementation of the application, as well as the building and unit testing. The design is described in the following subsections, using terminology and diagrams from RRRT. The reader is referred to Rational (2002) for a detailed explanation of these concepts.

4.1 External Connection Manager Template

The design of the connection managers for the external interfaces is based upon the TCPIP Key Mechanism with only a few modifications to enable its use in the CIU.

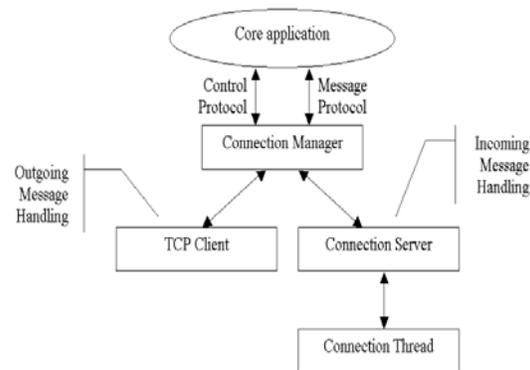


Figure 4.1 External Connection Manager Architecture

Since the modifications were minor the design will not be described in detail here, however the basic architecture is shown in Figure 4.1 in order to provide a reference for the discussion on the CSTA interface design.

4.2 CSTA Connection Management

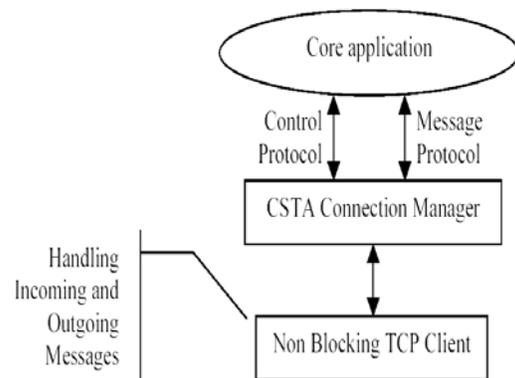
4.2.1 Architecture

When looking at the design for the management of the CSTA connection, it is pertinent to investigate the possibility of re-using any parts of the TCPIP

Key Mechanism. To recap, the CSTA interface uses a permanent bi-directional connection acting as a client.

The overall structure of the External Connection Manager Template, whereby the core application communicates with the manager using one protocol for connection control and another protocol for messaging, and the manager controls the capsules that deal with the low-level connection processing, could be re-used. Indeed, using the same protocols would help to create an integrated connection layer with a common interface to the core application.

The *TCPClient* capsule from the Key Mechanism cannot be used here since it only handles outgoing messages. Similarly, the *ConnectionThread* capsule only handles incoming messages. Moreover, it uses a blocking socket to wait for input. This means that all other operations on that thread are blocked until a message is received. Whilst this is viable in the case of the external interfaces (since that thread is only performing a single operation, either transmitting one message or receiving one message) it is not appropriate for permanent connections where the relative ordering of incoming and outgoing messages on the single thread is indeterminate. In the latter case, it is possible that an outgoing message may need to be sent before the blocking call has returned. A non-blocking socket must therefore be used, so that stimuli from either the socket or the core application can be acted upon immediately.



4.2 CSTA Connection Manager Architecture

NonBlockingTCPClient capsule is created that handles both incoming and outgoing messages asynchronously. In order to make the *NonBlockingTCPClient* capsule re-useable, a new *ConnProfile* data class is created to hold interface specific information such as knowledge about the format of the message header and position of the message length field. This class is initialised by the *CSTAConnectionManager* and passed to the

NonBlockingTCPClient on connection startup. Thus the operation of the connection is tailored to the CSTA interface.

The CSTA Connection Manager has responsibility for the startup and shutdown of the CSTA connection, as well as the configuration of the connection profile. The complexity lies in the *NonBlockingTCPClient*.

4.2.2 NonBlockingTCPClient Design

The *NonBlockingTCPClient* capsule has the following responsibilities:-

- Maintenance of the link to CallBridge DX, if the connection is in a started state.
- Reading and writing of messages at the socket level.
- The ability to handle asynchronous bi-directional communication with the *CSTAConnectionManager*.

The implementation of the first two of these was already understood since this type of interface was implemented in the prototype. The implementation of the last one, however, is specific to the way in which Rational Rose RealTime handles its internal protocols and requires some investigation.

In RRRT, when one capsule transmits a message to another capsule, the message is placed onto the receiving capsule's queue and the capsule is signalled to wake up and process the message. This processing is all done within the RRRT capsule framework and normally the developer does not need to be aware of the mechanism by which this is achieved.

In the *NonBlockingTCPClient*, however, the capsule needs to be woken up either by a message from the *CSTAConnectionManager* (i.e. the normal wakeup mechanism) or by activity on the socket. In order to do this it is necessary to customise the capsule's behaviour to allow it to check for both types of event. RRRT permits customisation of a capsule running on its own thread by enabling the developer to change the type of thread controller from *RTPeerController* (the default setting) to *RTCustomeController*. The developer can then override the default *waitForEvents()* operation with a customised version.

To enable the capsule to detect events on both the local interface (to the *CSTAConnectionManager*) and the remote one (to CallBridge DX) use is made of the TCP/IP "select" function through which the

software can specify a number of sockets to be monitored for activity. RRRT implements this functionality through its *RTIOMonitor* class. For this to work it is necessary for a socket to be used for the local interface as well as the remote one, however a UDP socket will suffice in this case since reliability of the link will not be an issue.

On connection startup the following actions are performed by the *NonBlockingTCPClient* :-

1. create a UDP socket (attribute *internalFd*) and connect to the local port
2. register the UDP socket with the *RTIOMonitor* (attribute *ioMonitor*)
3. register the remote socket (attribute *c_socket*) with the *ioMonitor*, so that its status can be monitored
4. attempt to connect to CallBridge DX
5. if connection succeeds, the client is now ready to process messages
6. if connection fails, a retry timer is started, upon expiry of which connection will be attempted again

The *waitForEvents()* operation monitors activity on *internalFd* and *c_socket* as follows:-

1. it checks the status of the sockets by calling *ioMonitor.wait()* with a parameter of 0 to indicate 'no blocking'.
2. if there is something to read on *internalFd*, i.e. *ioMonitor.status(internalFd)* returns a non-zero value, then it wakes up the state machine by calling *recv()* on the *internalFd* socket. The internal message will then be processed through the state machine.
3. if the remote connection is in the established state and there is some data to read, i.e. *c_socket.hasData()* returns a non-zero value, the *readSocket()* operation is called to read the data in from the socket and process it.

4.3 LMUI Connection Manager

4.3.1 Architecture

Since the LMUI connection is, like the CSTA connection, a permanent bi-directional link, it also cannot re-use any of the classes from the TCP/IP Key Mechanism. It should use the same control and messaging protocols to communicate with the core application, and will also have the same basic structure consisting of a connection manager (*LMUIConnectionManager*) to handle the startup and shutdown and to initialise the connection profile, and a low-level capsule to handle the socket processing.

However, as the LMUI connection acts as a server rather than a client, it cannot re-use the

NonBlockingTCPClient capsule. Instead, a *NonBlockingTCPServer* capsule will be created.

```

static const RTTmespec awhile( 1, 0 );
static char anything;

waitForEvents()

if ioMonitor.wait( &awhile ) <= 0 )
    return;

// Check the listening socket
if listener.hasData() )
{
    checkForNewClientConnections();
}

// Check for data to read
if ( c_socket.state() == RTTsocket::Established ) && c_socket.hasData() )
{
    readSocket();
}

// Check for internal signals
if ( ioMonitor.status(internalFd) & RTIOMonitor::CanRead != 0 )
{
    :recv( internalFd, &anything, 1, 0 );
    return;
}

```

Figure 4.3 *waitForEvents()* operation in *NonBlockingTCPServer*

4.3.2 NonBlockingTCPServer Design

The design for the *NonBlockingTCPServer* is similar to that for the *NonBlockingTCPClient* in that it needs to create a UDP connection to the connection manager and use an *RTIOMonitor* to check for activity on both this and the client connection. The key differences are that on startup, a listening socket is created and the server then waits for the client to connect; the listening socket has to be registered with the *ioMonitor*, so that any new connection activity can be detected by *waitForEvents()* (see Figure 4-8); if a new client connection is received when the *c_socket* is already active, then the new connection is refused by closing it down; and if an active connection goes away, the server just waits for a subsequent reconnection – no retry state is required.

5. RESULTS AND DISCUSSION

5.1 Test Objectives

Since the connection management software is very low level it is crucial to the operation of all communication between the application and the other system components. As such it is important to verify the connection software thoroughly before testing the higher level functionality.

The testing must prove that the appropriate connections can be established, that data can be transmitted in either direction, and that the software can recover from a connection failure.

5.2 Test Strategy

The initial unit testing of each connection manager in isolation was to be performed using Rational Quality Architect (RQA-RT), a desktop test tool that is integrated into RRRT. RQA-RT provides the

developer with the capability to specify, and automatically verify, a test sequence. The aim is to test as near to 100% of the code as possible. In order to demonstrate the RQA-RT test procedure that was used the testing of only one of the connection managers (the CSTA Connection Manager) will be described in detail, in section 0. The same procedure was followed for all connection components.

Once unit testing of all components has been completed, the application would undergo pre-integration testing on a test system in the laboratory followed by a period of thorough functional and performance testing.

5.3 Unit Testing the CSTA Connection Manager

In order to test any of the connection managers it is essential first to have an application to simulate the remote end of the connection. In the case of the CSTA Connection Manager, a server that can send and receive CSTA type messages is required. Note that the full CSTA encoding and decoding is not required for the testing of the low-level connection software since the connection management code is only concerned with the transport of the messages, not their content. Thus, the phrase “CSTA type message” means that the header part of the message has the correct length and format for CSTA messages.

A simple TestServer application was created in RRRT with the help of the *NonBlockingTCPServer* capsule. The parent *TestServer* capsule creates the

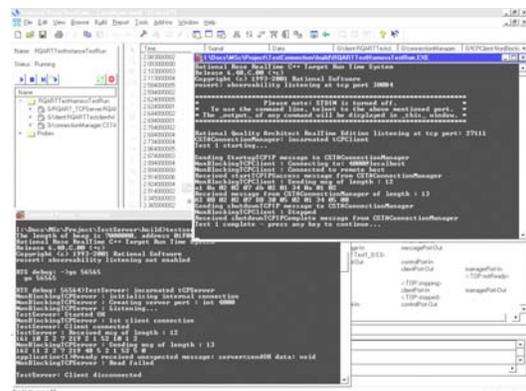


Fig. 5.1 Screenshot of RQA-RT Test Run

NonBlockingTCPServer, and then just prints out connection information and received messages, sending a response message in reply. The TestServer application was built in RRRT and the executable was run from the DOS command line when required.

The next stage of test preparation is to create a test capsule in RRRT, this was named *ConnectionTest*. This will contain the capsule under test (CUT), in this case *CSTAConnectionManager*, as well as a dummy client capsule, *ConnClient*, that is connected to the test capsule to simulate the core application.

Each test is now specified by creating a sequence diagram showing the expected sequence of events for the particular scenario. Input data must be specified for signals sent into the CUT and, similarly, the content of output signals must be checked. Once the test specification is complete and a build component has been created for the test capsule, RQA-RT is then invoked to verify the sequence. RQA-RT automatically generates the test code for the dummy *ConnClient* capsule, then builds the test capsule and runs the sequence. Finally, it verifies the test output against the expected sequence and reports any differences.

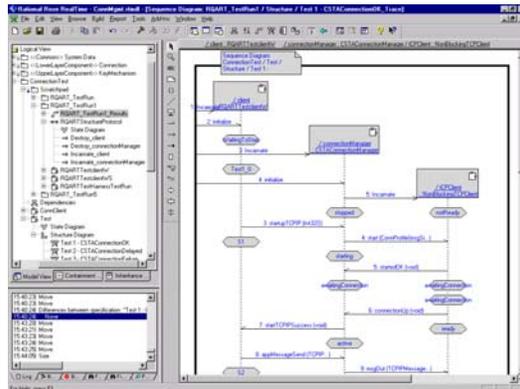


Figure 5.2 Screenshot of RQA-RT Test Trace

Figure 5.1 contains a screen shot of a running RQA-RT test. It shows the *TestServer* output window as well as the console window for the test. In the background is the RRRT window that is controlling the test run.

Figure 5.2 shows a screen shot of the test trace that is generated at the end of the test run. The Log window in the bottom left of the screen details any differences between the test specification and the trace. In this case there were no differences.

6. FUTURE DEVELOPMENT

The report shows that it is possible to design a consistent approach to managing to diverse types of connection within a single application and that this can be done within the Rational Rose RealTime framework by use of the *RTCcustomController* designation.

Furthermore, the use of RQA-RT as a testing tool enabled rapid development of a test environment, which would have been very difficult to achieve using traditional methods.

The combination of a robust framework with the already proven interface specific code from the prototype development meant that very few problems were found relating to the connection management software, and only one fault was raised in this area of software during formal system testing. This fault, in the external connection manager software meant that a failure to connect to the remote server in order to transmit a message caused the application to block for around 20 seconds.

Further work has been done more recently to enhance the performance and maintainability of the external connection managers. Use of the dynamic thread creation and deletion capabilities of RRRT has enabled the production of a single common connection manager which dynamically incarnates TCPClient capsules on separate threads. This has solved the bug that caused the application to block.

REFERENCES

RATIONAL SOFTWARE CORPORATION 2002. **Rational Rose RealTime Modeling Language Guide**

KEPNER, C.H. & TREGOE, B.B. 1997. **The New Rational Manager**. Princeton, New Jersey: Princeton Research Press

EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION 1992. **Standard ECMA-179, Services for Computer-Supported Telecommunications Applications (CSTA)**
Available at: <http://www.ecma.ch/>

EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION 1992. **Standard ECMA-180, Protocol for Computer-Supported Telecommunications Applications (CSTA)**
Available at: <http://www.ecma.ch/>

BIBLIOGRAPHY

COMER, DOUGLAS & STEVENS, DAVID 1991. **Internetworking with TCP/IP**
Englewood Cliffs, New Jersey: Prentice-Hall International, Inc.

ACCELERATING JOINT DESIGN: SIMULATION BUILDING BLOCKS AND PROCESS SUPPORT

EDWIN C. VALENTIN, JACO H. APPELMAN and
MARIELLE DEN HENGST-BRUGGELING

*Delft University of Technology, the Netherlands
faculty of Technology, Policy and Management
{edwinv ; jacoa ; mariellb}@tbn.tudelft.nl*

Abstract: Simulation is often seen as a powerful, but time consuming research instrument. We think that by employing simulation building blocks it is possible to use simulation in joint design meetings. We reflect on field experiments where this approach was tested and discuss how these findings inform the design of the joint design meeting. We aim to offer the first contours of an approach that could deliver a jointly designed simulation model in one day.

Keywords: Collaboration, meeting activities, process design, simulation, joint design, building blocks, collaborative engineering.

1 INTRODUCTION

Strategic decisions that involve the use of technology cannot be undertaken without taking into account their impact on relevant stakeholders. Collaboration is therefore an important activity during the development of new policies or preparation and execution of technically complex projects. This communication intensive process tends to consume a lot of time because it usually has to provide solutions for a mix of interrelated strategic or technically complex problems that impinge on different disciplines [Ackermann & Eden, 1996; Vennix, 1996]. A solution often used to shorten decision-making processes is by employing Group Decision Support [Vreede, 1995]. However, the actual act of decision-making is not the most intricate part in the process, evaluation of different alternatives or convergence towards one best or most shared solution is much more difficult. A popular research instrument that can support these meeting activities is simulation because it provides insight in the behaviour of a system and visualizes outcomes. Actors involved in the decision-making process can qualify and quantify the different solutions.

However, the modelling of a system is a cognitively complex task in which the time needed to deliver a specified output cannot be accurately predicted and tend to consume a lot of time [Keller et al, 1991]. It is not uncommon for a simulation study to take over a year, which is often too long for decision makers. In previous research we found that simulation building blocks can reduce the time of a simulation study and still provide the support required by decision makers

[Verbraeck et al, 2002]. Support is improved because decision makers better understand the simulation models and much more alternatives can be evaluated in a shorter time-frame, thanks to the fact that model adjustment is significantly easier.

An interesting new research field is the combination of the design of collaborative meetings and simulation because it could amplify the advantages simulation delivers. We expect that simulation studies will be concluded much faster and we aim to reduce the lead-time of simulation studies from months to at most a couple of meetings. The way we hope to do that is:

1. through a reduction of the cognitive load involved in constructing a model by using simulation building blocks;
2. and by structuring and designing GSS-supported meeting processes explicitly.

Formulated in this way we built on earlier research in the research-tradition labelled Collaborative Business Engineering [Maghnouji et.al., 2001]. We describe in this paper a background on simulation and building blocks (section 2). Then we introduce the concepts of meeting activities and ThinkLets(section 3). We elaborate on our experiences with two experiments that meant to speed up joint design of models (section 4). In section 5 we provide an overview of our observations and we describe the sequences of meeting activities that should be performed during different phases of model construction. We conclude in section 6 with the main lessons learned concerning joint design processes that use simulation blocks and offer suggestions for further research.

2 BACKGROUND ON SIMULATION, BUILDING BLOCKS AND JOINT DESIGN

Joint design or collaborative engineering can be used amongst others, to develop solutions for problems in multi-organizational settings. Different actors with different opinions want to make sure their points of view are represented in the design and ensure that the selected solution satisfies stakeholders they represent. As a result even simple convergent problems are affected by politic and social contexts, which leads to messy problems. [Ackermann and Eden 2001, Appelman 2002, Vennix, 1996]. In such circumstances it is of utmost importance that all stakeholders involved have a common frame of reference, a shared group memory. Modelling is a way to condense information to such an extent that all participants can make sense of the impacts that the modelled system might have.

Visualization of the potential impacts of choices or policies through building blocks contributes to more speedy but robust decision-making. [Verbraeck et al, 2002]. Note that visualization is broadly defined, the use of charts, graphs or other means that condense information is also implied. Visualization is thus not only cartoon-like animation of trucks going from A to B. The visual is the dominant sense that allows us to grasp and formulate, in retrospect, knowledge that can be generalized and objectified. Something that is much harder for other senses like hearing and taste [Urry, 2001]. We hypothesize that visualization of outcomes or behaviour of a system boosts the alignment of perceptions. All participants SEE the same information and, they do not have to develop their own individual mental pictures as much as they would have if they would have listened to an oral explanation. Cognitive distance between group members be more easily achieved when visualization of the process and outcomes is possible. [Nooteboom, 2001] Perfect alignment of perceptions would mean that every member had the exact same image of all the objects and outcomes involved. Alignment therefore ensures that the participants involved in a process of collaboration come to more robust models in a shorter time-frame. Alignment of perceptions also contributes to the creation of consensus that supports decision-making and implementation. The more visualization and joint design practices contribute to a shared focus the more likely it will be that the project will also be successful in the implementation phase.

We conclude that there seems to be a great potential for collaborative simulation but, in practice, it is used sparingly. Keller et al [1991] propose that the following reasons mainly account for this fact:

- that the simulation experts do not understand the decision makers, and thus deliver simulation models that do not provide the right answers and;
- that the simulation process takes too much time, and thus provides the desired support after the ideal moment of decision making.

Simulation building blocks support the rapid construction of simulation models because visualization ensures a high recognizability of the concepts for all stakeholders involved [Valentin and Verbraeck, 2002]. Building blocks hide the complexity of the underlying code and the complex functional behaviour and they provide an easy to understand user-interface including visualization. At the same time, visualization could improve the quality of decision-making because all actors share the same set of visualized system building blocks and these enable cognitive distance to be bridged.

Building blocks are in first instance a technical feature, used for the execution of current simulation studies but we think that this concept constitutes a key innovation for joint simulation in the future [Pater and Teunisse, 1997; Valentin and Verbraeck, 2002]. Whether it concerns a new container port, the expansion of an airport, the organizational structure of local governments or the closing of a factory, in each of these cases different actors with different learning-curves, opinions and ideas talk and discuss a visualized lay-out that is the same for every actor involved.

Zeigler et al [2000] reflect on the different modes of collaboration necessary at different stages of model building. They base themselves on the concept of DEVS, which provides chunks of re-usable simulation models similar to simulation building blocks. The DEVS-approach incorporates all stages of a model-building trajectory [Banks, 1999]. Each stage consists of a number of phases. To each phase, within a stage, a collaboration mode is assigned and a drawn-out process is implied. Zeigler et al do not, however, devote much text on what the different collaboration modes entail in terms of cognitive (group-) activities that have to be performed. This “simple” version of collaboration and simulation is visualized in figure 1.

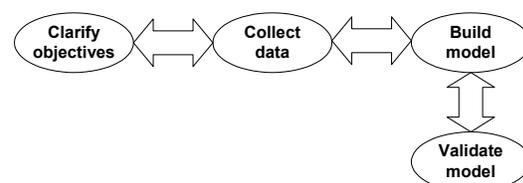


Figure 1: Phases in Model Construction (adopted from Zeigler et al, 2000)

They do not expand on what happens in the different phases and do not explain what the different actors should do during the joint design process. We intend to remedy this situation by formulating a process design based on meeting activities and simulation building blocks. For the sake of the argument, brevity and clarity we assume that all activities take place in one conventional meeting.

In collaborative settings the validation of the model will directly lead to suggestions for adjustments to the model, which means a new model. Although the arrows in the “simple” model represent the iterative character of collaborative processes it is not visualized clearly enough and it does not show at any point in the process the modeling exercise could be abandoned.

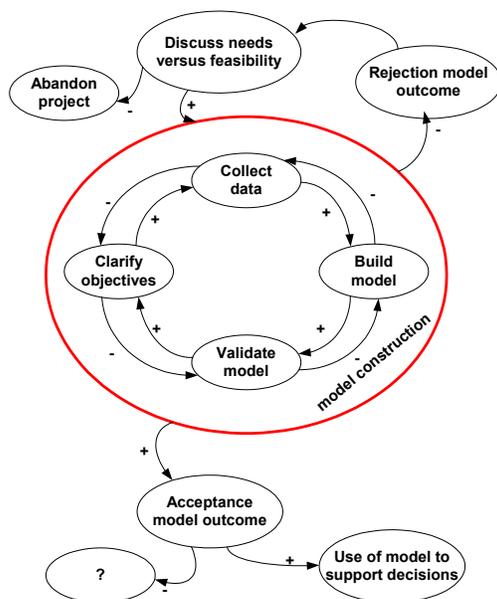


Figure 2: Dynamic Model of Model Construction

Figure 1 displays only a small part of a more complex problem solving project. In figure 2 we contextualize the generic core-model, we show the relation of the 4 phases within the whole project. The actors involved have the power to stop the cycle at any moment and accept or reject whatever outcomes have been reached thus far. The process facilitator should constantly be aware of the possibility that such a contingency occurs. One of his main tasks is to ensure that people stay committed and participate as long as is needed.

3 PROCESS STRUCTURING USING MEETING ACTIVITIES

We argued that Zeigler’s work provided a good foundation to model/describe meetings but that it could be ameliorated by the provision of more detail on what needs to be done during the different phases. In order to be able to do this in section 5 we introduce here the concept of meeting activity and ThinkLet.

At an abstract level the following 5 meeting activities can describe any meeting: Diverging, Converging, Organizing, Evaluating and BuildingConsensus. Divergence is popularly known as brainstorming and denotes going from a few or no concepts to a larger number. Convergence works the other way around, here we reduce the amount of concepts, ideas, etc. Organizing serves to structure in information in such a way that the participants and the constituencies they represent recognize their input and to ensure that all the participants share the same group memory. Put differently it is a move to a better understanding of the relationships between concepts or categories. Evaluate involves specifying criteria to value concepts. It is a move to achieve more understanding concerning the values a group attaches to concepts. Building Consensus is the last activity and it denotes the process of getting more agreement among stakeholders with the outcomes of the meeting. (although it can also be an objective of a meeting in itself). Building Consensus inevitably entails a degree of negotiation and could be considered a ‘cross-over’ phase or an activity that connects collaboration with negotiation. Since negotiation is a separate field of research we do not go into these activities involved.

In addition to the meeting activities researchers at TU-Delft and Arizona University have formulated a smaller unit of analysis, that represent processes and (the settings) of technologies used in meeting activities. In short, structures the design of GSS-supported meetings, this unit has coined a ThinkLet and is defined as: “the smallest unit of intellectual capital required to create one repeatable, predictable pattern of thinking among people working toward a goal.” [Briggs and de Vreede 2001: p.1] ThinkLets have as an additional benefit that they focus on the process of thinking and reasoning a group must go through in order to achieve a goal. Which was one of the elements we found lacking in the description of Zeigler and we need such insights to integrate joint design of simulation models with building blocks and meeting design. In other words, we aim to integrate the processes of thinking and reasoning that are part of simulation and modelling exercises with processes of reasoning and thinking that are commonplace in (electronically supported) meetings. A ThinkLet is build up of 3 parts:

- the script,
- the tool and,
- configuration of the tool.

The script explains how the ThinkLet should be introduced to the group by the facilitator or chairman as part of the meeting activity. The script contains instructions/explanations about the tool, and the activity to be performed. A tool is defined as a description of the version of the software and hardware used to create a pattern of thinking. The configuration details the specifics of how the hardware and software are configured to create a particular pattern of interaction. We disregard the hardware requirement in this paper that is necessary for research purposes because otherwise verification of results cannot be undertaken. Here we do not compare GSS-research outcomes, so the need is not there.

ThinkLets are categorized according to the kind of meeting activity they support but there are many variations to a theme. One can for instance, diverge in multiple ways. We will return to this observation in section 5.

Now that we have introduced the concepts of ThinkLets and meeting activities we continue, in section 4, with two experiments that elucidate the intricacies and barriers that groups of users experience when using simulation blocks. We draw lessons from these cases that, at a later stage, might lead to 'recipes' or 'prescriptions' of sequences of ThinkLets for joint design meetings. In section 5 we limit ourselves to a formulation of process guidelines that would support successful joint design meetings. We will do so, after we have described the activities to be undertaken in each phase of model construction.

4 OUR OWN EXPERIENCES WITH JOINT DESIGN AND SIMULATION

As argued before, process improvements occur in joint design meetings when simulation building blocks, that visualize output or represent it graphically, are used to develop simulation models. Improvements in efficiency (time-compression) and effectivity (more robust models) are anticipated. In this section we sketch our first experiences and conclusions on joint design based on one field (airport-security) and a laboratory (container-terminal design) experiments. The first experiment we researched the phases of model building and validation in the last and more controlled experiment we made the participants go through all 4 phase involved in model construction.

4.1 Security at Amsterdam Airport Schiphol

The European Committee requires that everyone in the lounges of international airports is screened for security reasons. This means that airports have to integrate security with passport control. The effectuation of this requirement involved participation

of four important stakeholders at the Amsterdam Airport Schiphol:

- Securop, the security company of the airport;
- KMAR, the Royal Dutch Police the organization that checks passports;
- The passengers' representatives;
- Schiphol passenger management.

These four different stakeholders entertained different ideas and preferred different solutions. For example, Securop liked to provide a large number of staff; the passengers do not want to wait and Schiphol Management does not want to pay.

In three joint design meetings the actors were brought together to discuss different alternatives using simulation. The design options were the number of passport checks and security X-ray machines, the sequence (first passport check followed by security check or other way around) and the planning and allocation of personnel over different locations at the airport. Detailed descriptions of the simulation building blocks used in these experiments can be found in Verbraeck and Valentin [2002].

During the sessions, designs were configured and transformed to simulation models. Transformation took a few minutes, but the run of the simulation and the evaluation of the outcome was a cumbersome task. The process facilitator needed to keep the participants occupied with anecdotes, but participants responded that this was misuse of their valuable time. They were able to quickly specify what they wanted because of the ease of use of the building blocks and expected that the results could be displayed almost instantly. However, the transformation slowed down because the level of detail inside the simulation models resulted in a lot of data that complicated analysis. The level of detail also affected the ability to rapidly adjust the model and affected speed of the simulation runs. The main effect of slow transformation was that participants became dissatisfied with process and therewith with outcomes.

4.2 Design of container terminals

A management game served as a pre-text for a laboratory experiment. Ten different kinds of (simulated) actors (like municipality, bank, logistics company, environmentalists and residents) were part of the experiment. All actors were obliged to jointly design a container terminal. The management game showed how participants should interact and specified how many different performance indicators were needed to design a container terminal from hundreds of design options. Like in any multi-actor system, each actor had a different set of priorities and desired outcomes for different performance indicators. For example, residents did not want the container terminal to make a lot of noise, a bank wants a stable financial plan and an operator expects a healthy profit margin.

Most of the performance indicators of a container terminal could only be evaluated with a simulation model. We developed a set of simulation building blocks that represented the behaviour of a container terminal. We simplified the design process by the provision of a link to the drawing environment VISIO and outcomes of the simulation model were produced in the form of an Excel sheet.

The participants (70 students of our full-time education program) filled a questionnaire after the game. Main conclusions from the survey are that the support tool consisting of VISIO-drawings, a pre-defined database, easy to understand Arena models and representation of outcomes in Excel, helped them with:

- Understanding each other's roles and preferences (94%)
- Evaluating the performance of the designed container terminal (85%)
- Speeding up the design process (81%)
- Making container terminal designs of high quality (68%)

More detailed information concerning the outcomes of the survey can be found in Bockstael-Blok et al [2003].

We tentatively conclude that the use of simulation in this management-game was a success, however, observations by experts on collaboration made some other things clear as well. For example, the participants could and did easily hide some input parameters, which made their results look much better than they actually were. The participants also limited themselves to a discussion of topics defined by the support tool. They did not produce alternative solutions like: noise-shields, deepening the canal or replacing the marina because these topics did not have corresponding input parameters in the support tool.

5 THE INTEGRATION OF PROCESS SUPPORT AND SIMULATION BUILDING BLOCKS

We combine the lessons from sections 3 and 4 and formulate a number of suggestions on how to model the process of GSS supported meetings using simulation building blocks. By linking the domains of group systems support and simulation it is possible to deliver better simulation models in a shorter time span.

5.1 Pre-meeting

A meeting can only be effective when it is thoroughly prepared. A meeting that aims to design in the context of a multi-actor environment should even be better prepared because of the need to satisfy diverging or opposing interests and to cope with complexity that is a result of the need to accommodate interests or to integrate different technologies to produce a solution. Another reason for the rejection of model outcomes

surfaced from the experiment. This became especially clear in the airport field experiment. The delays that occurred between the input given by the participants and the time needed to produce visualized results did not match their expectations. The net result was dissatisfaction with both process and outcomes. The outcomes the validated models delivered were not trusted. The outcomes were rejected and a discussion on the need and feasibility of a new round of joint design meetings ensued. It was decided that the model construction phase would be outsourced so decision makers would only have to bother themselves with selection of the 'right' model.

It is therefore important to try to mitigate goal-divergence, manage expectations, accommodate interests and to avoid delays between input delivered by the participants and output generated by the model/GSS. Ideally, the 'object-clarification' phase and data-collection phases are completed before the meeting. Practically it would mean that:

- A set of simulation building blocks has been developed or bought that fit the design questions of the actors involved.
- The time needed to transform input into visualized output is tested and improved if deemed necessary. Put differently, a number of simulation models should be developed beforehand.
- An initial simulation model, ideally drawn from information provided by the organizations and their representatives that will participate in the design meeting(-s), has been developed.

Zeigler et al comment on this phase: "... consider that it is very hard to get people to agree on common objectives in building a model. Perhaps, the only way to do that is to bring them together in one room and try to hammer out agreement through seemingly-endless discussions. Several meetings of this sort might be required." (Zeigler et al, 2001:534) However, we said that we would aim to reduce the cycle of model construction to one meeting. Which is what we will do in the next subsections.

5.2 Meeting activities to support the model construction cycle

5.2.1 Phase 1: Clarify objectives

The meeting should start with explanations by the facilitator or problem-owner(-s). It should be made clear why the meeting is held, why the group of persons is brought together, why simulation will be used and what the possible outcomes of the different simulation runs will be. This exercise is important because it is a mean to manage the expectations of the group and we have seen that unmanaged expectations concerning the time needed to produce results in the Schiphol case led to rejection of the model and its

outcomes. The facilitator should make sure the process is of interest to all participants. They will feel more committed when they perceive the advantages of participating.

When the meeting is supported by a GSS, participants will start to diverge, brainstorm. They produce a list of objectives that then needs to be organized to allow for an evaluation. The evaluation will deliver a ranking and reduce the number of objectives as far as possible. Through evaluation the group converges toward the most important objectives. Then a new divergence activity starts where participants can comment on the remaining objectives. This activity ensures that the participants align their individual perceptions they bridge cognitive distance in this phase. The last step is the building of consensus. This can be a lengthy process but if these meeting activities are converted to a design with ThinkLets we estimate that this phase can be done in 2 to 3 hours. Provided that all decision makers are present and not one of them has an agenda of frustrating the meeting. Secondly, divergence can also be done in a 'relay-form'. This opens up the possibility to let decision makers define objectives, that are then checked by technical experts concerning the feasibility of the objectives. Once closure has been reached on an objective a facilitator can instruct the GSS to forward this objective to domain experts that in their turn provide the data needed. So, you get cycles within cycles, within cycles and this explains why you can really speed up the model construction phase. Small groups are simultaneously working on different phases but always follow the right order that lead to the construction of a model. When done in such a way it becomes possible to construct a model in one meeting.

Reality is however never so clear-cut as we would like to have it. We do not deem it likely that everybody will start to do 'one-day modelling meetings'. We therefore continue to comment on how meeting activities can best be supported during the different phases.

5.2.2 Phase 2: Data collection

Data collection can be a synchronous non-distributed group activity, but it is unlikely because it would mean that a facilitator would be able to predict, before the phase of the clarification of objectives has commenced, what data need to be gathered and which actors can deliver that particular data. It is perhaps much wiser to do this a-synchronous and let simulation expert evaluate the data for usability. However, if done in a meeting the emphasis in this phase should be on divergence and organization activities. As much data as possible should be elicited from the participants and be stored in a group memory to prepare for the next phases.

5.2.3 Phase 3 and 4: Build model and validation

We combine phase 3 and 4 to emphasize the iteration involved in these phases. During these phases a technical simulation model builder should be available for support. Simulation building blocks support the easy and rapid construction of models. Different models and/or different outcomes can be produced. As a result the model building and validation phases will be characterized by convergence and evaluation meeting activities. Within the small group different designs can be generated and evaluated using the simulation model. The order of input in the simulation model determines the order in which topics are discussed. When a simulation model is ready, the simulation model can be executed, followed by an evaluation of the performance indicators. If the outcomes are not satisfactory the simulation model can be adjusted and re-executed. This process continues until the actors in the group can converge to an agreement. The time it takes to build consensus can vary enormously and is influenced by a host of variables that cannot be influenced during a meeting. Consistent with the cases we assume that the goal of the meeting is to produce a model of which the input parameters can be changed. The process of consensus building that should eventually lead to select one best solution or model is not considered.

Taking a number of constraints into account we do think it will be practically possible to design and construct a model jointly in one day. But only if the political activity surrounding the project is low, otherwise people will not feel free to divulge all the information needed to design and construct a model.[Appelman et al, 2002]

6 CONCLUSIONS AND FURTHER RESEARCH

In this paper we describe the possibilities to support the design and use of simulation building blocks to enable joint design. We identified the need for a methodology that supports participatory design processes and the evaluation processes of the designs using simulation. Visual representation makes learning easier and speeds up different meeting activities that need to be performed in a sequence to come to a model everybody can agree to. Our own experiences with joint design were illustrated with two different case-descriptions of experiments with joint design in two different domains.

The most important lesson we learned was to keep expectations of the participants realistic and invest much preparation time in the phases 1 and 2. Investment is not just the development of the right simulation building blocks, execution of test-sessions it is also an investment in the enthusiasm of the participating actors. The experiment with container

terminals clearly showed that it speeds up phases 3 and 4 because participants had clear instructions, had to adhere to their role and were provided with the right simulation building blocks. We learned from the Schiphol experiment that time between input and visualization should be as short as possible.

6.1 Further Research

In the beginning of this paper we explained that we aim to develop a first methodology for joint design using simulation. The second case of the container terminal gave us very motivating results, but we know we still have to do a lot of things in the research of joint design. Firstly, we aim to replicate the experiments in order to optimise the design of the process and the simultaneous use of different GSS's. Secondly, we think two other areas of research could offer knowledge to better support joint design. On the one hand it would be interesting to know the extent to which it is possible to perform meeting activities such as joint design in a distributed setting. On the other hand research that delivers commonly performed sequences of meeting activities can inform the design of meeting processes supported by a GSS.

REFERENCES

Ackermann, F. and C.Eden "Contrasting Single User and Networked Group Decision Support Systems for Strategy Making," *Group Decision and Negotiation*, volume 10, issue 1, p47-66. 2001

Ackermann, F. and C.Eden "Contrasting GDSSs and GSSs in the Context of Strategic Change; Implications for Facilitation," *Journal of Decision Systems*, volume 6, issue 3, p221-250. 1996

Appelman, J.; Rouwette, E.; Qureshi, S. "The Dynamics of Negotiation in a Global InterOrganizational Network: Findings from the Air Transport and Travel Industry," *Group Decision and Negotiation*, volume 11, issue 1, p145-163. 2002

Appelman, J. "Acceptance and transition Of Group Support Systems in Interorganizational Networks.," *Proceedings of the Group Decision and Negotiation Conference Perth, Elsevier*. 2002

Banks, J. "Introduction to simulation", In: P.A. Farrington, H.B. Nembhard, D.T. Sturrock and G.W. Evans (Eds.) *Proceedings of the 1999 Winter Simulation Conference*, p7-13, 1999

Briggs, R.O.; G.J.de Vreede; J.F.Nunamaker; D.Tobey. "ThinkLets: Achieving Predictable, Repeatable Patterns of Group Interaction with Group Support Systems (GSS)" In: R.H. Sprague and J.Nunamaker (Eds). *Proceedings of 34th Annual Hawaii International Conference on System Sciences* (CD-ROM), 10 pages, 2001

Bockstael-Blok,W.; I.S.Mayer, E.C.Valentin "Supporting the design of an inland container terminal through visualization and gaming-simulation" In: R.H. Sprague and J.Nunamaker (Eds). *Proceedings of 36th Annual Hawaii International Conference on System Sciences* (CD-ROM), 10 pages, 2003

Keller, L. ; C.Harrell ; J.Leavy. "The three reasons why simulation fails". In: *Industrial Engineering*, Volume 23, Issue 4, p27-31, 1991

Maghnoouji, R.; G.J. de Vreede; A. Verbraeck; H.G. Sol; "Collaborative Simulation Modeling". In: R. Sprague, J.F. Nunamaker (Eds.) *Proceedings of the 34th annual Hawaii International Conference on System Sciences* (cd-rom). p. 1-10. 2001

Nooteboom, B., Learning and innovation in organizations and economies, Oxford University Press, 2001.

Pater, A.J.G.and M.Teunisse "The Use of a Template-Based Methodology in the Simulation of a New Cargo Track from Rotterdam Harbor to Germany" In: S.Andradottir, K.J.Healy, D.H.Withers and B.L.Nelson (eds.)*Proceedings of the 1997 Winter Simulation Conference*, San Diego, 1997

Valentin, E.C.; A. Verbraeck. "Simulation using building blocks " In: F.J.Barros, N.Giambiasi (Eds.) *Proceedings conference on AI, Simulation and Planning*, p65-71, 2002

Vennix, J.A.M. *Group model-building: facilitating teamlearning using system dynamics*. Chichester: John Wiley. 1996

Verbraeck A.; Y. Saanen; Z. Stojanovic; B. Shishkov; A. Meijer; E. Valentin; K.van der Meer. *Chapter 2:What are building blocks?.* IN: Verbraeck and Dahanayake (eds.) *Building blocks for Effective Telematics Application Development and Evaluation*, TU Delft press, 2002

Vreede, G.J.de. *Facilitating Organizational Change*, Dissertation TU Delft, 1995

Vreede, G.J. de and A. Verbraeck. "Animating Organizational Processes - Insight Eases Change". In: *Simulation Practice and Theory* 4 Elsevier Science B.V., Amsterdam. pp. 245-263. 1996.

Urry J., 'Mobile Cultures' (draft), Department of Sociology, Lancaster University, http://www.comp.lancs.ac.uk/sociology/soc030ju.html

Zeigler B.P.; H. Praehofer, T.G. Kim. *Theory of Modeling and Simulation : Integrating Discrete Event and Continuous Complex Dynamic Systems*. Academic Press, 2000.

AUTHOR BIOGRAPHIES

EDWIN C. VALENTIN is a researcher in the Systems Engineering Group of the Faculty of Technology, Policy and Management of Delft University of Technology. His specialty is the development of domain-dependent generic discrete-event simulation libraries. Edwin participates in the BETADE research program on developing new concepts for designing and using building blocks in software engineering, simulation, and organizational modeling. His web addresses is www.tbm.tudelft.nl/webstaf/edwin.

JACO H. APPELMAN is a senior researcher in the Systems Engineering Group of the Faculty of Technology, Policy and Management of Delft University of Technology. His specialty is design and support group decision making processes. Appelman has facilitated many group meetings in both public and private environments. His research focuses at the design of group processes to support (distributed) decision making.



MARIELLE DEN HENGST is a senior researcher in the Systems Engineering Group of the Faculty of Technology, Policy and Management of Delft University of Technology. She is a specialist in supporting methodologies for decision making processes. Using a way of thinking of models and group processes she is developing a methodology called Collaborative Business Engineering. Den Hengst has been actively involved in this field with projects for the Port of Rotterdam, the policesquads Amsterdam/Amstelland and the supply chain of Methanol.

AUTHORS INDEX

A'Apice, C.	263	Delamarche, P.	318
Aissani, A.	174	Do, T. V.	290
Al-Akaidi, M. M.	199, 543	Dobson, C.	329
Al-Begain, K.	302	Dugan, J. B.	217
Al-Dabass, D.	618, 628, 636	Dvornik, J.	520
Alfonseca, M.	59	Economou, A.	193
Alnsour, D.	199	Essafi, L.	25
Alsalami, M. A. T.	543	Fagan, M.	329
Amelkin, A.A.	311	Farah, A.	205
Andrews, S. G.	142	Ferrucci, L.	371
Antonic, R.	574	Fleming, P. J.	83, 568
Aplin, P.	124	Fortin-Parisi, S.	284
Argibay-Losada, P.	442	Franek, F.	48
Awan, I.	302	Frenkel, S.	278, 462
Azgomi, A.	169	Furfaro, A.	526
Bai, L.	71	Galluccio, L.	229
Baiardi, F.	371	Gelenbe, E.	5
Ball, G. R.	131	Grabovac, M. F.	538
Balsamo, S.	562	Gray, A.	636
Bandrivskyy, A.	180	Habbi, A.	43
Barbosa, G. J.	437	Hartley, J. K.	423
Barbot, N.	239	Heitzinger, C.	496
Beck, S.	618	Hennig, A.	509
Belavkin, R.	105	Hillston, J.	296
Benazzouz, D.	205	Ho, T. K.	378
Benzekri, A.	257	Houhamdi, Z.	154
Beri, S.	180	Huet, T.	110
Berry, S.	585, 589	Iwu, F.	391
Bocharov, P.	263	Jagnjić, Z.	349
Bolch, G.	25	Jović, F.	349
Bonotti, A.	371	Karatza, H.	385
Brát, M.	557	Karlsson, M.	532
Brenner, M.	211	Kerckhoffs, E. J. H.	514
Bruha, I.	48	Khalil, M.	595
Bruzzo, A.	491	Khalilidelshad, M.	222
Çakmak, H.K.	355	Kim, S.-Y.	404
Canessa, E.	136	Krcum, M.	520
Cant, R. J.	148	Krus, P.	532
Carter, J. M.	589	Kühnapfe, U.	355
Chakka, R.	290	Ladbrook, J.	431, 474
Chliveros, G.	338	Lancashire, L. J.	131
Clement, R.	344	Langensiepen, C. S.	148
Colobert, B.	318	Langton, C.	329
Conway, J.	628	Lara de, J.	59, 65
Cooper, D.	338	Lee, J. Y. B.	378
Craven, D. A.	538	Lee, S. W.	118
Cretual, A.	318	Lees, M.	77
D'Auria, B.	263	Leinemann, K.	411

Leonardi, A.	229	Schefczik, P.	186
Levytskyy, A.	514	Selberherr, S.	496
Limoeiro, C.	437	Sericola, B.	234, 239, 284
Logan, B.	77	Serrano, A. F. N.	169
Lopez-Ardao, J.	442	Sethson, M.	532
Lopez-Garcia, C.	442	Shaaban, Y. A.	296
Low, M. Y. H.	397	Sheikholeslami, A.	496
Lowndes, V.	585, 589	Shen, L.	71
Luchinsky, D.	180	Simner, J. C.	618, 628
Mannella, R.	180	Sisias, G.	329
Marzolla, M.	562	Sklenar, J.	417
Mather, P.	124	Smari, W. W.	404
Mattila, V.	456	Šnorek, M.	557
McClintock, P.	180	Solomon, L.	607
Meehan, J. W.	538	Stepashko, V.	603
Mian, S.	131	Stewart, P.	83, 568
Morabito, G.	229	Stocker, E.	217
Mori, P.	371	Stone, D.	83
Mosca, R.	491	Suarez-Gonzalez, A.	442
Multon, F.	318	Sviridov, A. P.	485
Munitic, A.	520, 574	Swift, C.	411
Nigro, L.	526	Tadić, Z.	349
Nikolaidis, G.	186	Tait, R. J.	355
Nolle, L.	53	Taylor, S. J. E.	474
Orsoni, A.	491	Tepper, J.	118
Orsulic, O.	520	Teujeiro-Ruiz, D.	442
Osman, T.	618, 628	Theodoropoulos, G.	77
Osman, T.	110	Thomas, N.	245, 251
Palmer-Brown, D.	118	Thornley, D.	251
Pand, Z.	290	Tolwinski, R.	469
Peytchev, E.	595, 636	Touzene, A.	163
Phillips, R.	329	Tropper, C.	607
Ponitsch, M.	509	Tutsch, D.	211
Pupo, F.	526	Vieira, C. A. O.	124
Rabehasaina, L.	234	Virtanen, K.	456
Raivio, T.	456	Vukic, Z.	574
Ray, C.	110	Whitelock, R.	636
Rees, R. C.	131	Wiedemann, A.	186
Revill, D.	509	Winnell, A.	431
Ricci, L.	371	Wu, M. H.	589
Riolo, R.	136	Wuwer, M.	618
Roadknight, C.	118	Yan-Da, L.	5
Robinson, S.	448, 474	Zatschler, H.	251
Rodrigues, M. A.	338	Zelinka, I.	53
Rodriguez-Rubio, R.	442	Zelmat, M.	43
Saghafi, F.	222	Zerfiridis, K.	385
Salem, O.	257	Zhi-Hong, M.	5
Sankaranarayana, R.	363	Zigman, J.	363
Santana, R.	91, 98, 272	Zobel, R.	34, 391
Schaefer, G.	355, 411	Zvorygina, T.	603