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.

Proceedings 17th European Simulation Multiconference
(c) SCS Europe BVBA, 2003 ISBN 3-936150-25-7

Figure 1 shows a Statechart representing the vant behaviour.

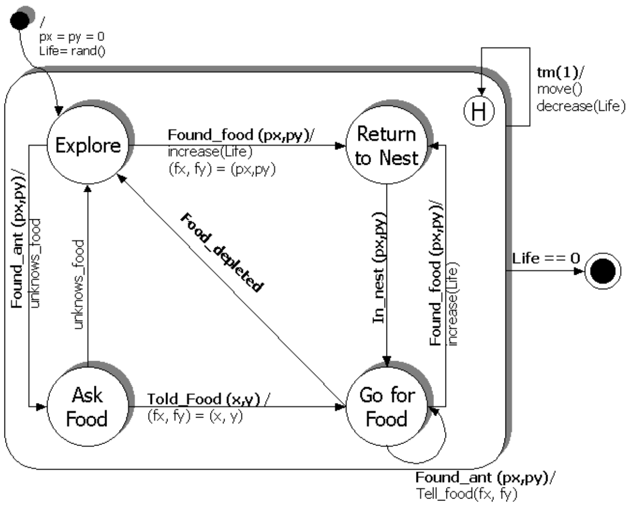

Figure 1: Behaviour of a vant.

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 Δ=Δx=Δy=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:

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

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

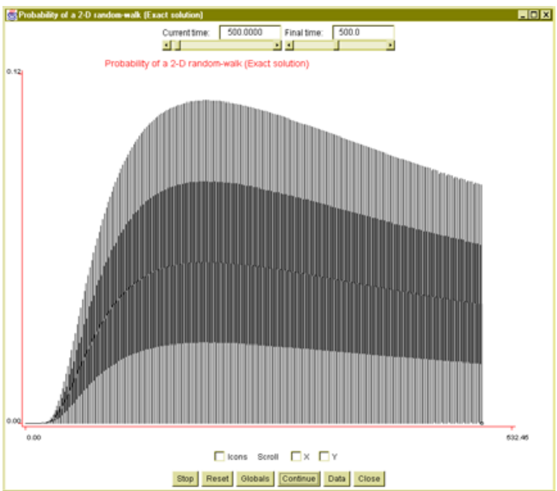

Figure 2: Probability that cell (10,10) is occupied in a random-walk of 100 walkers starting at (0,0)

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.

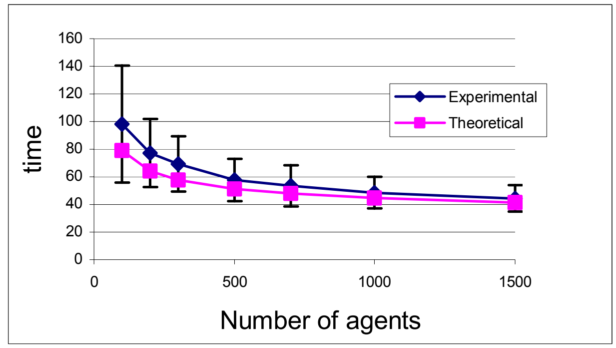

Figure 3: Average time to reach the food at (10,10) (theoretical and experimental) as the number of vants increases.

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.

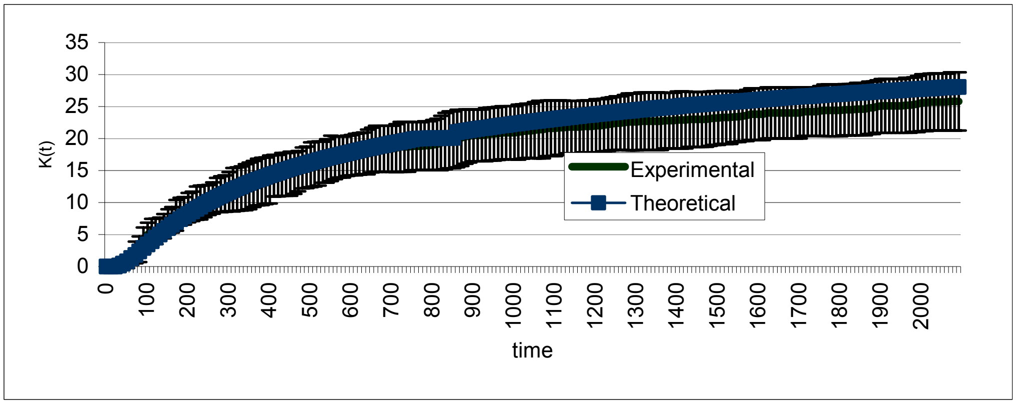

Figure 4: Vants that know the food position, without information exchange in collisions.

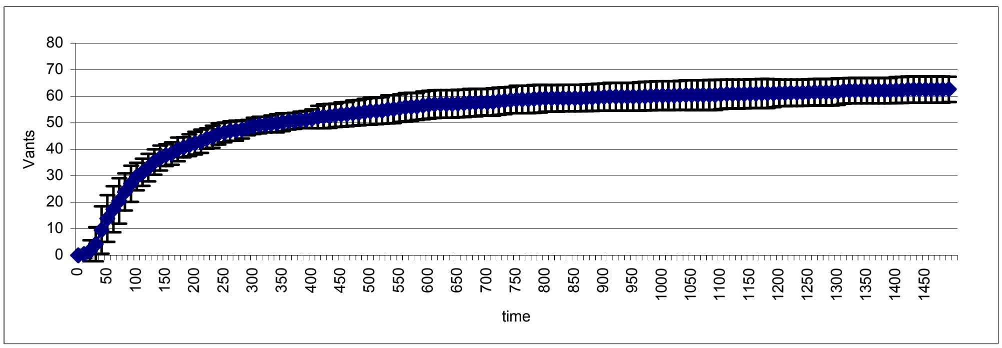

Figure 5: Vants that know the food position, with information exchange in collisions.

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 OOCSMP [Alfonseca and de Lara 2002b], each vant was represented as an OOCSMP object. The territory, with information about the food sources is represented as another OOCSMP 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/ecomm/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.

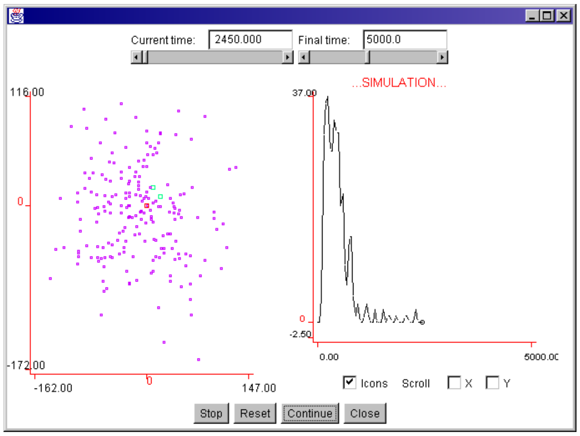

Figure 6: 200 vants with memory of 3 time steps

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.

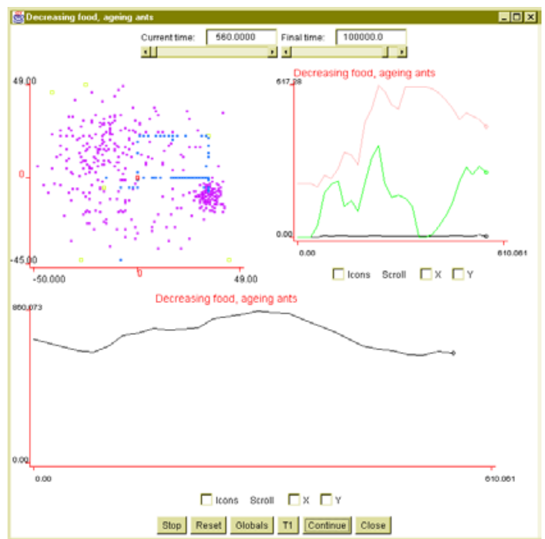

Figure 7: Simulating finite food places and dying vants

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/ecomm/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 OOCSMP*". 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 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.