MULTI AGENT IMPLEMENTATION OF AN URBAN ROAD TRAFFIC ADVISOR

Camelia Avram
Technical University of Cluj-Napoca
Ge. Bariţiu, no 26–28, Cluj – Napoca, 400027, Romania
E-mail: camelia.avram@aut.utcluj.ro

Adina Aştilean
Technical University of Cluj-Napoca
Ge. Bariţiu, no 26–28, Cluj – Napoca, 400027, Romania
E-mail: adina.astilean@aut.utcluj.ro

Tiberiu Leţia
Technical University of Cluj-Napoca
Ge. Bariţiu, no 26–28, Cluj – Napoca, 400027, Romania
E-mail: tiberiu.letia@aut.utcluj.ro

KEY WORDS
Multi-agent system, traffic management, routing algorithms.

ABSTRACT

This paper describes generic multi-agent systems architecture for an urban driving advisory system. This architecture is described as composed of basic role-oriented processes communicating though primitive interaction protocols. The architecture is aimed to provide an enabling communications framework upon which multi-agent system models can be organized and built to be used for an simulation of an urban map and to estimate the traffic behavior (to provide information about the best routes). The paper presents a hierarchical routing strategy for an Urban Driving Advisory System (UDAS), based on agent technology. UDAS assists the drivers to get the desired destination taking into account the current situation of traffic characteristics. It gives the estimated arrival time and the corresponding distance between a start and an arrival point. The necessary information is obtained from a real-time traffic control system (RTCS). The drivers could consult the advisory system using a variety of devices like mobile phones. The information given by the advisory system has the form of predefined short messages.

INTRODUCTION

The urban traffic is frequently perturbed by congestions, followed by usual delays, accidents and road closures that cause supplementary delays. The architecture of Urban Traffic Advisory System is described by Aştilean et al. (2002), according to this paper different clients can access the system to receive data about the road traffic. The communication between possible clients and UDAS can be done using a mobile phone (through Short Message Service – SMS, General Packet Radio Service – GPRS, Wireless Application Protocol – WAP).
We propose a system architecture base on multi – agent technology. The agents will work in an online environment. The architecture that we propose for this system is implemented on several different layers.

Communication makes it possible for agents to cooperate. To solve the communication problem, we use intelligent agent. When we describe agents as intelligent, we refer to their ability to:

- communicate with each other using an expressive communication language;
- work together cooperatively to accomplish complex goals;
- act on their own initiative and;
- Use local information and knowledge to manage local resources and handle requests from peer agents.

We implement agents with different function like: agents for urban map (for intersections, streets) and route agents. The needed data is collected from different sensors (installed on the road surface and counting the number of vehicles, cameras installed along the roads which provide visual information about the traffic, ultrasonic sensors etc.) locally stored and transmitted to other agents. The paper is structured into four sections: the first section introduces the early work in the field, in the second section the proposed multi-agent system architecture is presented, the third section the communication system used by agents is presented and in the fourth section the system used to provide the best route for the mobile users is presented shortly. The ant algorithm is used for finding the best route. The simulated map is divided into several zones. The routing algorithm is applied on two layers, first to get the zones needed to be pas and second on each zone to find the route between entry point and exit point (for the situation that we just pas the zone) or the entry point and destination (for the situation that we are in the final zone).

SYSTEM ARCHITECTURE

Multi – agent systems provide possible solutions to the traffic problem, while meeting all the criteria. Agents are expected to work within a real-time environment. For managing an urban traffic system, a hierarchical multi – agent system that consists of several locally operating agents each representing an intersection of a traffic system or an area (depend on the configuration and the complexity of the area) is proposed. Improvements to urban traffic congestion must focus on reducing internal bottlenecks to the network, rather than replacing the network itself.

Proceedings 20th European Conference on Modelling and Simulation
Wolfgang Borutzky, Alessandra Orsoni, Richard Zobel © ECMS, 2006
ISBN 0-9553018-0-7 / ISBN 0-9553018-1-5 (CD)

Traffic lights possess sensors to provide basic information relating to their immediate environment. This includes road and clock sensors, measuring the presence and density of traffic and providing the time of day to the traffic light.
A solution to the urban traffic problem using agents is to simply replace all decision-making objects within the system by a corresponding agent.
The proposed architecture is presented in figure 1.

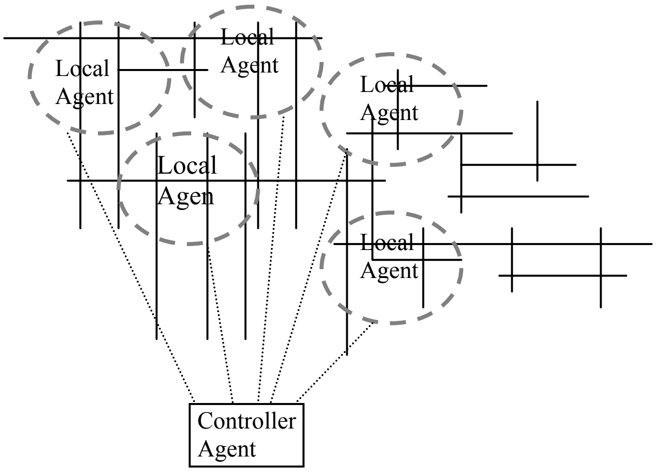

Figure 1. The system architecture

The map is divided into several areas and these areas are supervised by a local agent. All agents communicate with the controller agent. On this paper we propose a multi – agent architecture of the UDAS.

The UDAS will provide information about the traffic like: congestions, queues at the priority to all the requests from the police, ambulances and fire department and will provide detailed information to the public transportation companies.

In figure 2 are presented the goals of the multi agents system to achieve the imposed request on the system.

The controller agent must manage the local zones, manage the requests from users (about the urban traffic behavior, the best route between two points – provided by the user, manage the request of modification of an previous request about an best route, display the result for user – send an SMS, or how the user specified to receive the response, save the request and the response) and to manage the communication between different simulated zones. For zones simulation (urban traffic and different situations – like any traffic problems that occur and we simulate the map to find the best solution, ex: car accident and we have two or more possible solution for managing the traffic with minimum cost, time and money) we use the simulator presented by Avram & all, 2005. When the controller agent produces the simulation zones must take into consideration several request presented in the paper of Avram & all, 2005.

AGENT COMMUNICATION

The UDAS agent is based on a three-layer deliberative architecture. This architecture comes from conventional knowledge – based applications. This system separates the application into three knowledge

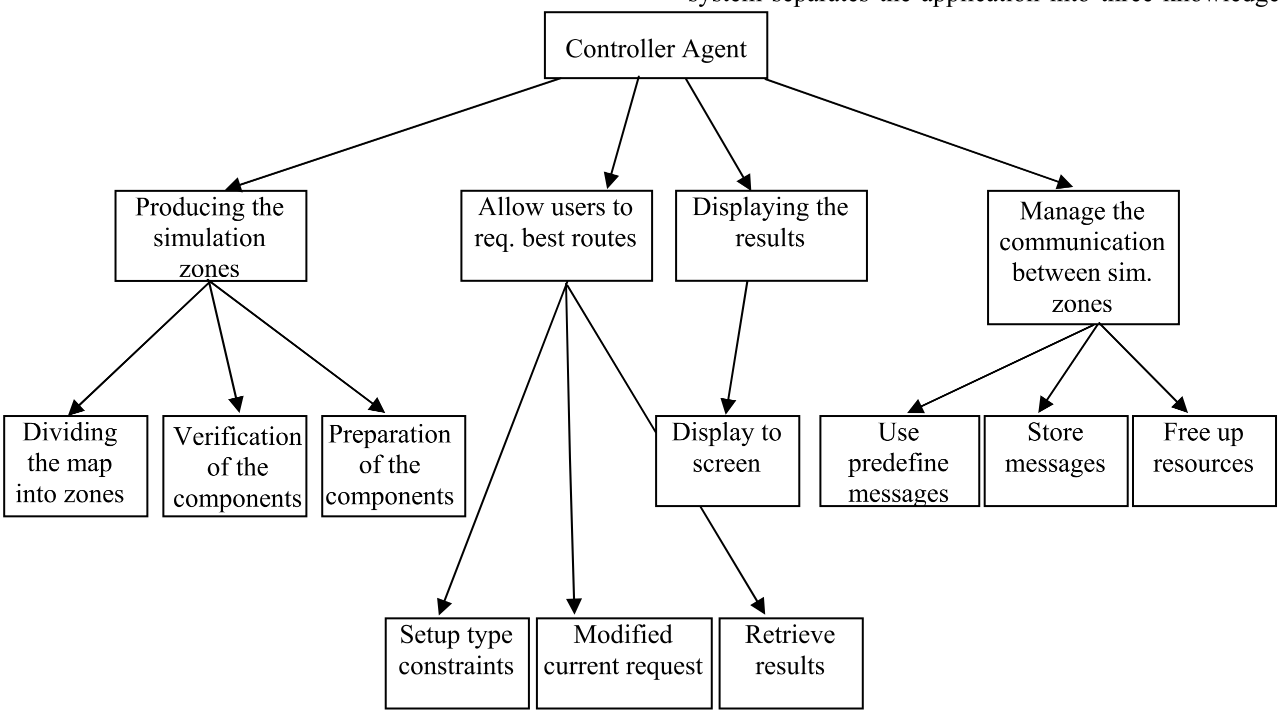

Figure 2. The goals of the controller agent

bases. The three layers that compose the UDAS agent architecture are the Knowledge Layer, the Inference Layer and the Controller Layer. The UDAS multi – agent architecture is presented in figure 3.

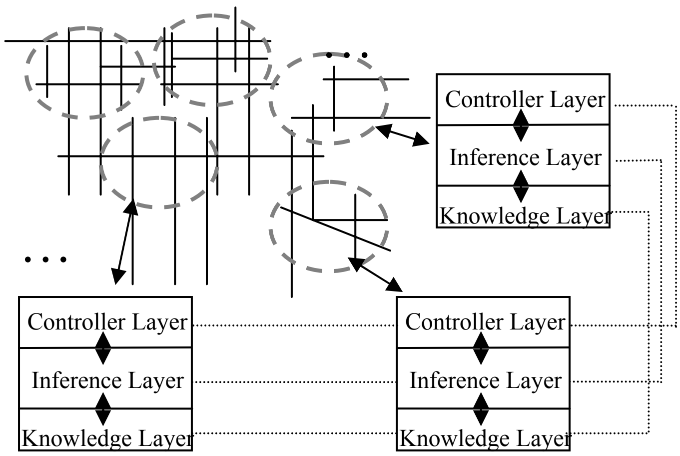

Figure 3. The UDAS multi –agents architecture

data from the geographical area consists on giving the list of instances of nodes, roads, etc.

The **Inference Layer** (IL) is where the functionalities of the UDAS agent are implemented. This layer uses the knowledge of the KL, and certain algorithms to calculate its results. More concretely, in order to calculate the proper route for a given petition, the IL implements a modified Dijkstra algorithm that operates with the graph contained in the KL (more information you can find in the paper written by Aştilean et. al, 2003) and for this paper we implement the ant algorithm. Dijkstra algorithm gives good result in case of a static search. The ant algorithm calculates the optimum route between two nodes. The selection method selected is the travel time from a node to its neighbors. The travel time is calculated dividing the distance between two nodes and the maximum velocity of the road that connect them. This velocity can be modified at run time in order to reflect the road state. That it's to say, to reflect the congestion level of a road: fluid, with intermittent stops, road closed, etc. This reduces the maximum velocity value used in the selection.

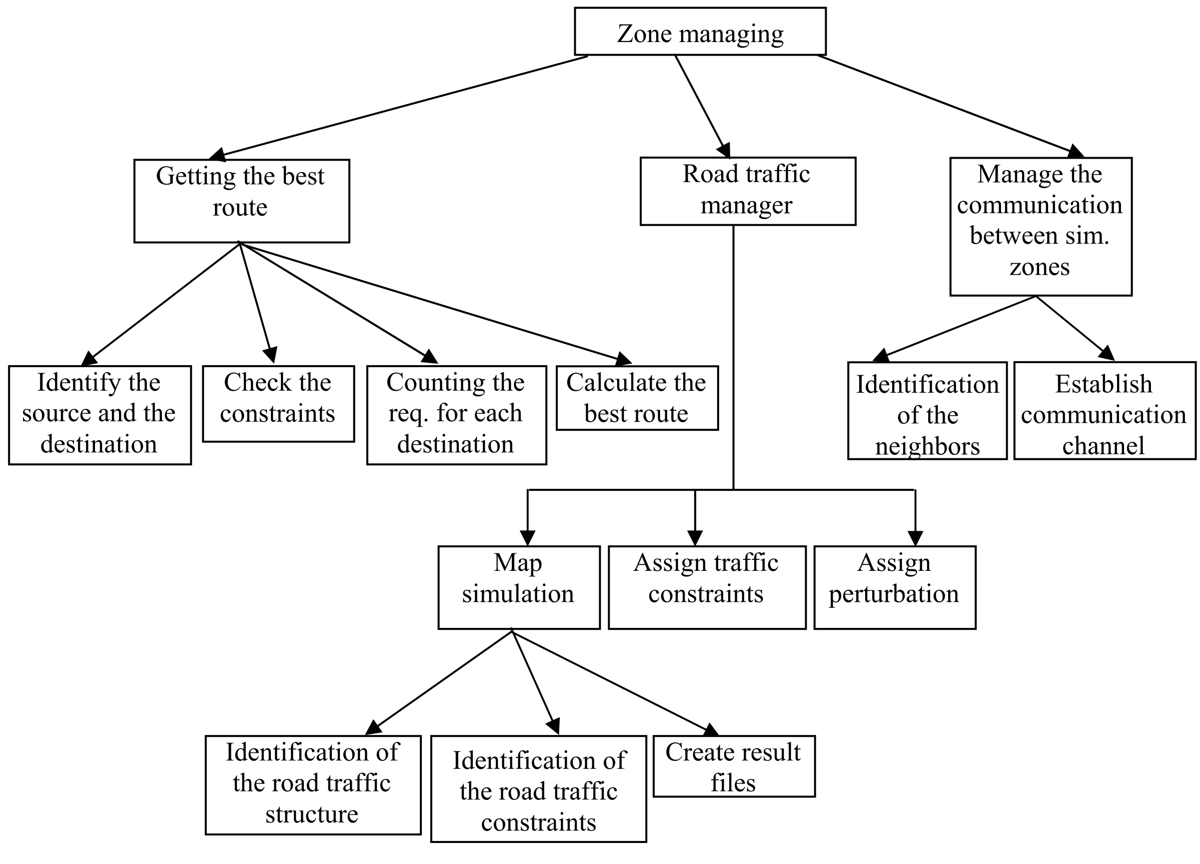

Figure 4. The goals for zone managing

The **Knowledge Layer** (KL) is the one where the knowledge about the traffic of the respective area is stored. Above other information, this layer contains a formal representation of the road map of certain area. This representation consists on a partial instantiation of the traffic ontology. The fulfillment of the KL with

Finally, **Controller Layer** (CL) is the one who manages the communication tasks of the UDAS agents. These tasks include the sending and receipt of Agent Communication Language (ACL) messages and the follow up of the coordination protocols that have been defined. This layer is able to interpret the incoming messages, to extract the query and send it to the lower layer. Also, it has to construct the proper message in order to reply the petition with the data received from the IL. On the other hand, the CL has to perform some other tasks, related to the FIPA (Foundation for Intelligent Physical Agents) specifications and the Agent cities network. It has to register the agent in the DF, search for other agent name in the DF, connect and disconnect the agent from the network, etc.

The goal is to develop a multi – agent system that makes use of the knowledge based methodology.

In figure 4 the goals of the agent that manage the simulation zone is presented.

The agent must simulate the road traffic for his zone and for simulation is using the simulator described by Avram & all, 2005.

This simulator is a hybrid simulator; use macroscopic model in case of the streets where we have information about the medium values of the speed and no. of cars (on and off ramps, or long street where we are not interested about the individual cars) and microscopic model in case of the streets where we want to track individual cars (intersections and the adjacent streets). Streets can be divided into several virtual segments and these segments can be modeled using microscopic or macroscopic model based on the amount of information from traffic. The simulator gives information about the no. of cars / time interval and the speed needed to pass that street (in case of the macroscopic model we get average values and in case of the microscopic model we get the speed of all cars that pass the street). This simulator can simulate an average map (with 3000 cars) and 2-3 hours of traffic in less than 2 minutes having different traffic scenarios (the map have several inputs streets, we can modify the time interval between the cars that enter on the simulated map).

The agent must determine the best route between two points from his simulated map; this goal can be achieve using the algorithm that will be presented later.

The UML Class diagram for finding the best route is presented in figure 5.

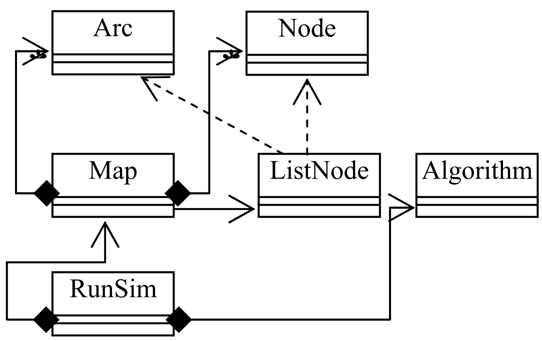

Figure 5. The UML Class Diagram of the package used for finding the best route.

For a scenario the agent can setup some constraints for traffic and also for the best route algorithm. The agent must estimate the traffic behavior for each request for a best route and set up some costs for each route. In case that we have several request for the same interrogation or part of the rote we set up some costs and these costs are use when we search for best route do not give the same result, other wise we will have some traffic jams started by our system (for ex.: In case of a music concert if several hundred request for the same destination are solved statically (giving the same result for all the request) will be a problem so we will memorize all the request, setup some costs and estimate the traffic behavior when the user will be in that area to get the best result from the system).

In figure 6 is presented the UML Package Diagram for the packages used for solving the estimation problem.

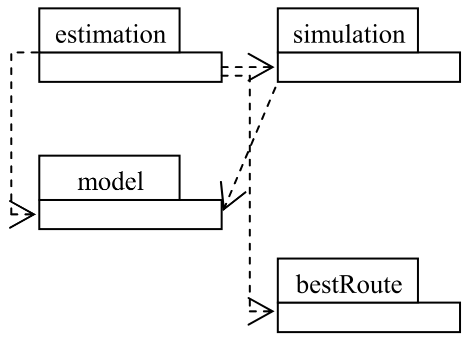

Figure 6. The UML Packages Diagram – used for estimation problem

BEST ROUTE

Ant algorithm

Ant algorithm is a class of meta-heuristics that can yield near-optimal solutions to hard optimization problems. They maintain a population of agents that exhibit a cooperative behavior. For example, ants deposit *pheromones* in the environment that influence others which tend to follow it. Such an approach is robust and well supports parameter changes in the problem.

Ant algorithms has been applied successfully to various combinatorial optimization problems like the Traveling Salesman Problem by Dorigo (1997), routing in networks by Caro (1997) and by White (1997), for distributed simulation by Bertelle et al., (2002) and graph partitioning by Kuntz et al., 1997).

Based on the model described on a previous chapter we describe the urban map as a weighted digraph:

$$G = (N, A), \qquad (1)$$

Where:

N is a set of nodes, representing the short segments and the long segments (the boundary between two segments) and;

A is a set of directed edges representing the length of the segments.

$$A = NxN; \quad a = (n_i, n_j) \qquad (2)$$

In case of the intersections from one node we can have multiple choices depending on the number of the possible next segments. Two directed edges, one in either direction, are used if the street is two – way, and a single directed edge is used if it is a one – way segment.

The edge weight w_{ij} between the vertices v_i and v_j is a dynamic factor which represents the time to cross the edge (v_i,v_j). We search in the graph some paths between two vertices v_0 and v_n. The resolution method is distributed and based on auto – organization mechanisms. We continually release numerical ants on the dynamic graph, and allow them to find routes between pairs of vertices. The ants deposit numerical pheromones on edges. The amount of pheromone deposited is a function of the length and congestion of paths. Ants are attracted by weights of edges and pheromones. The evaporation allows forgetting bad paths. The ants tend to converge on paths which are the fastest.

This algorithm is best suited for dynamic search of the map; in case that the search situation is changed the algorithm is applied from that point and if we use the Dijkstra algorithm we have to search the entire map again.

Best route finding system

In the process of finding the best route we divide the problem into two parts:

- First we apply the ant algorithm to a graph that has as nodes the zones (the map is divided into several zones) and the weight of vertices is a coefficient named P_{ij} (i, j are nodes). The P_{ij} coefficient is variable and his value is influence by degree of the occupancy of the specific urban zone. Using different values for the same P_{ij} in time the system can avoid the situation of sending all the clients' on the same route and blocking the streets.
- After we decide the optimum way zones we start to use the ant algorithm on each zone. Each zone has a number of input and output streets and for each of them we setup a coefficient. The coefficient is different for each search.

This system uses the earlier mentioned ant-based control algorithm (ABC-algorithm) by Kramer et. all, (1999). This algorithm makes use of forward and backward agents. The forward agents collect the data and the backward agents update the corresponding probability tables in the associated direction. The algorithm consists of the following steps:

- At regular time intervals from every network node s, a forward agent is launched with a random destination d: F_{sd}. This agent has a memory that is updated with new information at every node k that it visits. The identifier k of the visited node and the time it took the agent to get from the previous node to this node (according to the timetable) is added to the memory. This results in a list of (k, t_k) – pairs in the memory of the agent. Note that the agent can move faster than the time in the timetable.
- Each traveling agent selects the link to the next node using the probabilities in the probability table. The probabilities for the nodes that have already been visited by this agent are filtered out for this agent. Then a copy of the remaining probabilities is made for this agent and these probabilities are normalized to 1. Only this agent uses this temporary probability distribution to choose a next node. So the probability table is not updated yet.
- If an agent has no other option than going back to a previously visited node, the arising cycle is deleted from the memory of the agent.
- When the destination node d is reached, the agent F_{sd} generates a backward B_{ds}. The forward agent transfers all its memory to the backward agent and then destroys itself.
- The backward agent travels from destination node d to the source node s along the same path as the forward agent, but in the opposite direction. It uses its memory instead of the probability tables to find its way.

The backward agent with previous node f updates the probability table in the current node k. The probability p_{df} associated with node f and destination node d is incremented. The other probabilities, associated with the same destination node d but another neighboring node, are decremented.

The final result will be a sum of local optimal values:

$$R_{opt} = \sum_{i=1}^{n} R_{l\_opt}(i); \qquad (3)$$

R_{opt} – is optimal result;
n – Number of zones;
R_{l_opt} – is local optimal result.

CONCLUSIONS

In this paper the authors propose and present a multi-agent architecture for managing the urban road traffic and providing information about the traffic and best route between two points. The architecture presented is multilayered and the problem of agent communication is solved. The agents are intelligent; they can communicate each other and also can communicate with the controller agent. For solving the best route problem the authors transform the map into a graph and the ant algorithm is used for optimization. The costs for arcs of the graph can be time or distance between two intersections. The ant algorithm is best suited for a dynamic search.

REFERENCES

Aştilean, A., Avram, C., Leţia, T. S., Hulea, M. and Vălean, H., (2002). "Using Mobile Data Services for Urban Driving Advisory Systems", *Mobile Open Society through Wireless Telecommunications - MOST*

Conference, Varsow, Poland, pag. 4, ISBN 83 - 87091 - 32 - 4.
Aştilean, A., Leția, T., Avram, C. (2003). "Information and Communication Strategies for Urban Driving Advisory System", *CSCS – 14th International conference on control systems and computer science*, Bucharest, Romania.
Avram, C., Boel, R. (2005). "Distributed implementation of a heterogeneous simulation of urban road traffic", *High Performance Computing & Simulation Conference (HPC&S'05),* Riga, Latvia.
Bertelle, C., Dutot, A., Guinand, F., and Olivier, D., (2002). "Distribution of agent based simulation with colored ant algorithm". In *ESS2002*, pages 39–43, Dresden, Germany.
Caro, G. D. and Dorigo, M. (1997). "Ant net: A mobile agents approach to adaptive routing". *Technical report, IRIDIA*, Université libre de Bruxelles, Belgium.
Dorigo, M. and Gambardella, L.. (1997). "Ant colony system: A cooperative learning approach to the traveling saleman problem". *IEEE Transactions on Evolutionary Computation*, 1(1):53–66.
Kuntz, P., Layzell, P., and Snyers, D. (1997). "A colony of antlike agents for partitioning in vlsi technology". In *Fourth European Conference on Artificial Life*, pages 417–424, Cambridge, MA:MIT Press.
Kramer, K. H., Minar, N., and Maes, P., (1999). "Tutorial: Mobile Software Agents for Dynamic Routing", MIT Media Lab Room E15-305, 20 Ames Street, Cambridge, MA 02139 USA, *Published in Mobile Computing and Communications Review* vol.3 no.2
White, T. (1997). "Routing with swarm intelligence". *Technical Report SCE-97-15*, SUA.

AUTHOR BIOGRAPHIES

CAMELIA AVRAM went to the Technical University of Cluj–Napoca, where she studied automation and obtained her degree in 2001. She is a PhD student at the same university. Her e-mail address is: camelia.avram@aut.utcluj.ro and her web-page can be found at: http://users.utcluj.ro/~avramc.