AGENT BASED ROUTING URBAN DRIVING ADVISORY SYSTEM

Camelia Avram[1], Adina Astilean[2], Mihai Hulea[3], Tiberiu Letia[4], Honoriu Valean[5]

[1, 2, 3, 4, 5] Technical University of Cluj – Napoca, Automation Department,
Ghe. Baritiu st, nr. 26, Cluj – Napoca, Romania,
{camelia.avram, adina.astilean, mihai.hulea, tiberiu.letia,
honoriu.valean}@aut.utcluj.ro

Abstract: 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.

Key words: multiagent system, GSM communication, routing algorithms

1. INTRODUCTION

The urban traffic is frequently perturbed by congestions, followed by usual delays, accidents and road closures that cause supplementary delays. Even if a delay is known in advance, it is difficult to find a suitable alternate route, either because the driver is unfamiliar with the area or because those alternatives are just as clogged with traffic as the route that someone tries to avoid. The mobile telephones can be used to demand and receive information about the best route optimized not only for distance, but also for the shortest drive time the current traffic conditions allow. The architecture of Urban Traffic Advisory System is described by Astilean 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 or Multimedia Messaging Service – MMS).
We propose a system architecture base on multi – agent technology.

Agent technology can be implemented on several different layers. Communication makes it possible for agents to cooperate. For solving 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. Another source for information about traffic situation is tracking vehicles by Global Position System (GPS). Agents are expected to work within an online environment.

On-line feedback control of urban traffic networks forms a challenging problem, both due to the size of the plant and due to the complexity of the dynamics to be taken into account. This requires the use of a simulation package that is capable of evaluating the effects of several options for the setting of the control actuators (mainly traffic lights in the urban traffic control problem) faster than real time. The size of the plant imposes the use of a distributed simulation package.

The Ant Based Control algorithm is used for the route finding system. It is especially suited for the proposed distributed approach. A probability is used for all alternatives. The probability tables contain local information and no global information on the best routes. For each node, the next step in the route is determined. The route is composed of a list of nodes to be followed by vehicle and is composed of small parts from several components. When the route is complete or the maximum number of nodes in a route is reached, the response is sent back to the driver.

This paper is structured into three major parts; first we present the system architecture and briefly the road traffic model that we use, second we present the communication, between traffic agents, problem and third the ant algorithm used to find the best route between two points.

2. 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.

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.

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

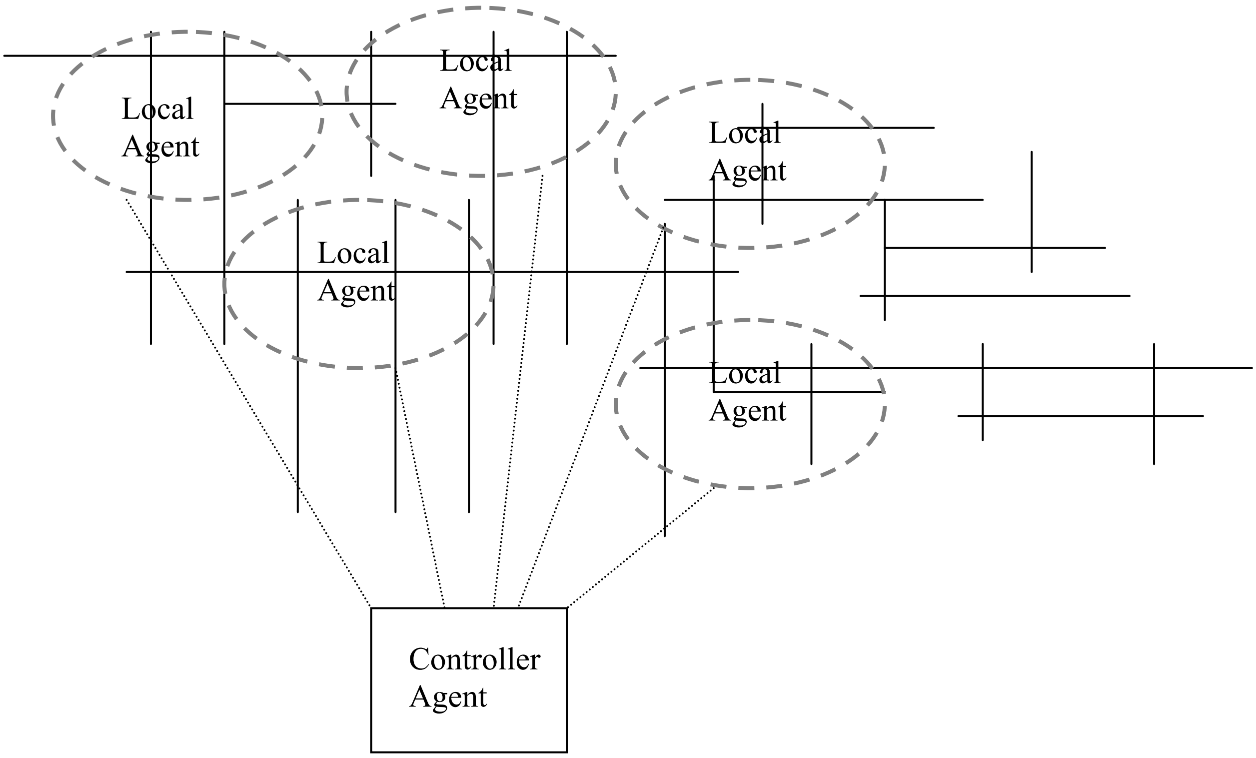

Figure 1. The system architecture

intersections, flows propagation, best routes and possible accidents. This system will give higher priority to all the requests from the police, ambulances and fire department and will provide detailed information to the public transportation companies.

3. URBAN ROAD TRAFFIC MODEL

The mathematical model that forms the basis for our distributed simulation consists of many interconnecting components, the outflow of an upstream component being the inflow in the next down-stream component. For road networks where the distance between intersections is big, one often can assume that the traffic state is roughly the same over segments of a length of several 100m. This allows the use of macroscopic models; describing the evolution of the following aggregated variables (no individual vehicles are represented):

- the flow q(x,t) of vehicles (vehicles/minute) at time t and at location x ,
- the density ρ(x,t) of vehicles at time and at location x, in (vehicles/km) and
- the average speed v(x,t) of these vehicles, in km/h.

Classical macroscopic models develop partial differential equations for the evolution of these aggregated variables, and simulate the behavior of the network by time and space discretisation See e.g. the METANET simulator, in the paper written by Papageorgiou (2002).

In the simulator for urban traffic used in this paper we take into consideration different models, for long segments (macroscopic model) and for short segments (microscopic model). Vehicles enter a short segment n through its upstream boundary at the event time "j-th vehicle enters segment n". Vehicles are then propagated via a sequence of consecutive cells, moving to the next cell as soon as they can have driven a distance equal to the length of the cell (this time is calculated using the current speed of the vehicle), and when the next cell is free (not blocked by another vehicle, nor forbidden by safety constraints imposed by downstream vehicles). More detail about the model that we use you can find in paper written by Avram and Boel (2005).

3. 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 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 2.

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 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 Astilean et. all, (2003)) and for this paper we implement the ant algorithm. 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.

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.

4. BEST ROUTE

4.1 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).

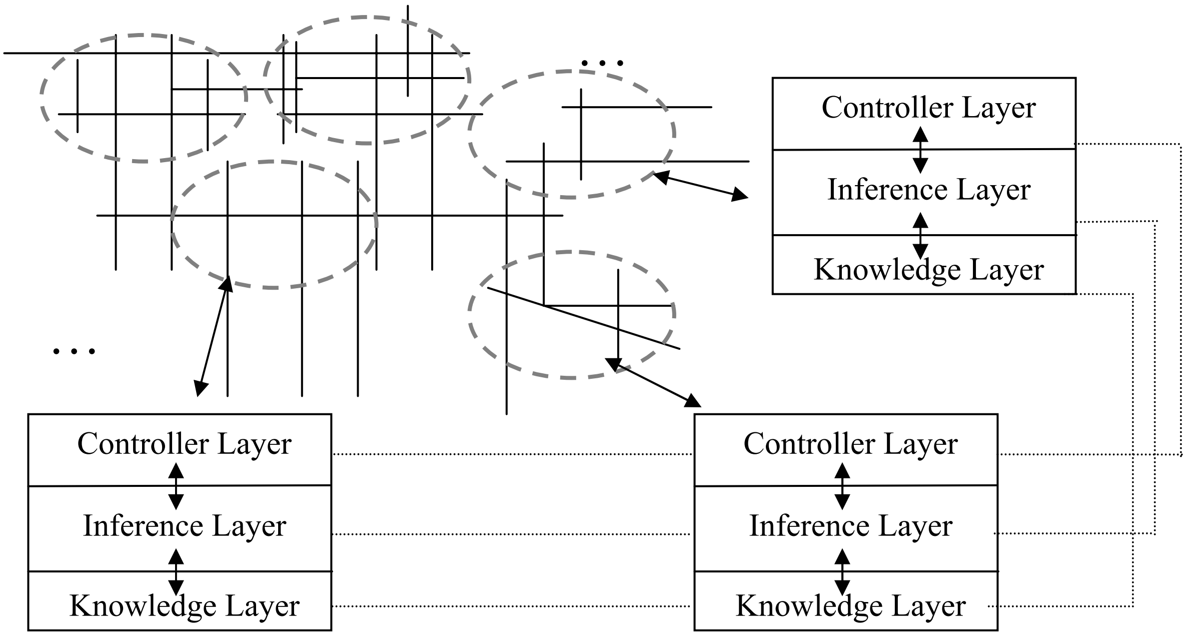

Figure 2. The UDAS multi –agents architecture

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

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

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.

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

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. To be able to distribute the computation, we have divided the algorithm in two parts and for each we have a specific time:

- The environment. It is represented by the dynamic graph. Its major role is to manage the ant population, evaporation phenomenon and simulation of weights on the edges. We store also in the vertex v_n the shortest path which comes from v_0, the minimal global cost W_{0n} of the path from v_0 to v_n. Due to the dynamic change of weights the duration of the shortest path may change when another ant covers the path crossing the same vertices and we note the instant where the ant has found the same path. For a given step, we have:

t_{env} = discrete time of the environment
BEGIN
Birth of ants on the vertex v_0
Pheromone evaporation
Weight update
IF *$t_{0n} << t_{env}$*
THEN *$W_{0n} = +\infty$*
ENDIF
$t_{env} = t_{env} + 1$
END

- The ants. Ants try to go from the vertex v_0 to another vertex v_n. Ants manage their displacements according to times and pheromones. They also drop pheromones on edges. Three states are possible for an ant looking for food, reaching the final vertex v_n and coming back to the source.

We take into consideration:
t_{ij} – the amount of pheromone trail deposited on the edge connection i and j.
w_{ij} – the weight of the edges which depends on the time of the traffic flow to connect the location i and j (this value can change in time).
p(i,j) – probability that an ant when it is located on i choose j.

4.2 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. al, (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:

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

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

CONCLUSIONS

In this paper, in order to decrease traffic congestion, we proposed the navigation mechanism with route information sharing based on mass user support. With multi – agent modeling, we construct a simple traffic flow model.
The Ant Based Control algorithm is used for the route finding system. It is best suited for the proposed distributed approach. Multi-agent systems are adapted to find emergent evolutionary solutions in dynamic problems.

REFERENCES

Aştilean, A., Avram, C., Leția, 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, Varsovia, 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.

Papageorgiou, M. et al., (2002). "Freeway Ramp Metering: an Overview", *IEEE-T- Intelligent Transport. Systems,* Vol. 3, No. 4, pp.271-280.