FIPA-compliant MAS development for road traffic management with a Knowledge-Based approach: the TRACK-R agents.

A. M. García-Serrano
Technical University of Madrid
Campus Montegancedo, s/n
Boadilla, 28660, Madrid (Spain)
Pho: +34.91.336.7441
agarcia@dia.fi.upm.es

D. Teruel Vioque
Universidad Carlos III
Ed. Sabatini, C/ Butarque,15
Leganés, 28911, Madrid (Spain)
Pho: +34.91.624.9117
dteruel@inf.uc3m.es

F. Carbone and V. D. Méndez
Technical University of Madrid
Campus Montegancedo, s/n
Boadilla, 28660, Madrid (Spain)
Pho: +34.91.336.7390
vmendez@isys.dia.fi.upm.es

ABSTRACT
In this paper, we describe our work in the design of agent-based systems using available standards and our previous experiences in the development of knowledge-based applications in the domain of road traffic management. Our main goal in this first step has been to join the development of FIPA compliant agents with the use of knowledge-based techniques during design and a logic programming language. The result of this work has been reflected on the TRACK-R agents, responsible of a geographical area that has to communicate in order to provide traffic routes recommendations for humans or other agents. Currently we have three agents running that allows a simple human request for recommendation as well as other TRACK-R agents requests. This work has been funded by the proyects DAMMAD (TIC2000-1370-C04) and Agentcities.NET (IST-2000-28384).

Categories and Subject Descriptors
I.2.11. [**Artificial Intelligence**]: Distributed Artificial Intelligence – *Coherence and coordination, Intelligent agents, Languages and structures, Multiagent systems.*

General Terms
Design, Languages, Standardization.

Keywords
Multiagent systems, traffic management, prolog, FIPA, Agentcities.

1. INTRODUCTION.
Multiagent systems (MAS) are becoming, in the last few years, a growing technology for developing distributed services. Many organizations and companies are adopting this technology to its own development and research processes [5]. One of the reasons of this increase is the creation of large-adopted standards. These are the FIPA specifications [7], whom guaranty the interoperability between agents that fit those standards, which are called FIPA-compliant agents. Moreover, many tools have been developed to facilitate the development and deployment tasks for this kind of systems.

Real time road traffic management is an appropriate domain for the multiagent systems: traffic is geographically and functionally distributed. Traffic management problems and subsystems are varied and have a high degree of autonomy, and usually traffic applications are highly dynamic. In this framework, a multiagent system including specialized agents on problems/services is a natural approach [8].

Our work concerns the design and development of the TRACK-R agents. These agents generate and sort the optimum route for a car driver from a city to another one. In order to generate this route, the TRACK-R agent infers on a knowledge base, composed by a partial instantiation of a traffic ontology [6]. Every TRACK-R agent will be responsible of a geographical area, which means a different instantiation of the ontology for each agent. Therefore, if the network involves different areas but with shared elements, the related TRACK-R agents will have to communicate in order to achieve a jointly recommendation.

In this paper the TRACK-R agents are described, and how different technologies were joined to the development of a multiagent system using a knowledge-based approach. These techniques allow to build intelligent agents that make use of complex reasoning methods in order to help the decision taking between a response generation or to start a coordination step. In particular we briefly describe the service (section 2), and the architecture of the agents involved in this project (section 3). A preliminary design of the coordination protocols is explained at section 4 according to different scenarios. Finally is included a description of the current state of the application and some conclusions and future work.

2. THE TRACK-R SYSTEM.
The **TRACK-R** (*Traffic AgentCity for Knowledge-based Recommendation*) is a multiagent application, enclosed into the Agentcities.NET project [2]. The objective of this application is to support a service to offer a road route recommendation between two given cities. The selected route is the best possible, in aspects

of travel time. That is to say, the system will return the fastest path between two cities by car.

For the design of this service, we define a multiagent architecture based on several agents, called **TRACK-R agents** (see Figure 1). These agents have the knowledge of the traffic map and state of different areas in Spain (Madrid, Barcelona and so on). More detailed information of the content of the TRACK-R agents is given on chapter 3.

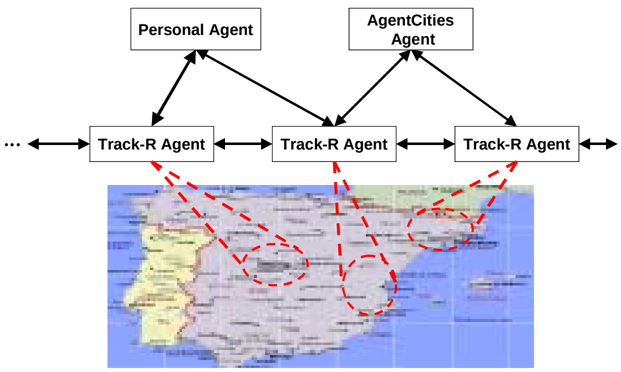

Figure 1. Scenario of the application

In addition of these agents, a user interface agent called Personal Agent is included in the architecture. The behavior, so the design, of this agent is quite simple. The personal Agent just acts sending the user requests and showing the results to the user in a windows-based GUI. The service was designed in order to include it into the *Agentcities.NET* network, so its functionality can be access not only by the personal agent, but also for any agent into the mentioned network.

3. TRACK-R AGENT ARCHITECTURE.

The TRACK-R agent is based on a three-layer deliberative architecture. This architecture comes from conventional knowledge-based applications. These systems separates the knowledge the application manages into one o several knowledge bases; and the processes that has to use this knowledge to build the functionality are called inference engines. The three layers that compose the TRACK-R agent architecture are the Knowledge Layer, the Inference Layer and the Coordination Layer.

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. DAML+OIL was used for the original description, but not for the implementation. The elements of the ontology were translated to prolog terms. This process could be automated, allowing to parse a DAML+OIL ontology to a set of prolog predicates and terms directly. The explanation of this ontology is out of the scope of this paper, but a subset of this ontology can be seen at figure 2. As can be seen, there are several concepts and relationships in this ontology. The fulfillment of the KL with data from the geographical area consists on giving the list of instances of nodes, roads, etc, from this area. For example, in the area of Madrid, the KL contains a instantiation of the concept node for Leganés, a road instantiation for the M-40, and so on. We called it partial, because not all the concepts or relations are needed, and so, are not instantiated. This process allows having the same nomenclature for the contents of the KL, what the agent knows, in all the possible TRACK-R agents. Also permit to other agents who want to access the functionalities of the system, to understand each other if they use the terminology of this traffic ontology.

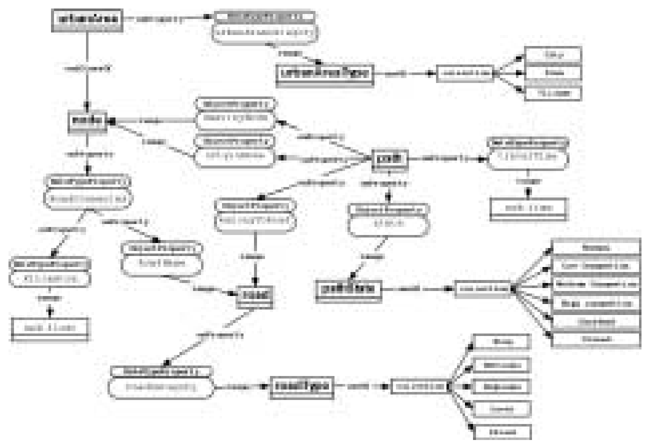

Figure 2. Partial view of the Traffic Ontology.

The **Inference Layer** (IL) is where the functionalities of the TRACK-R 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. This algorithm calculates the optimum route between two nodes, over a not-directed graph. 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, **Coordination Layer** (CL) is the one who manages the communication tasks of the TRACK-R agents. These tasks include the sending and receipt of 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 specifications and the Agentcities 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.

As it has been said, the intention was to develop a multiagent system that makes use of the knowledge based methodology. Current frameworks for multiagent development are not suited enough to the development of complex knowledge bases. Nevertheless, there are other tools, like Ciao Prolog [4], that are useful for this task, but in this case, they are not well suited for FIPA-compliant agent development. Our solution was to join both technologies, in order to get the best of them, and to solve their deficiencies when they are used sedately. The way to join them is applying the connectivity facilities between Java and Ciao prolog that the last offers. This process is detailed explained at [3].

Therefore, in the development of TRACK-R agent's architecture, both technologies were used. KL and IL was programmed with Ciao Prolog, in order to benefit from the capabilities of a logic programming language as prolog for defining and managing knowledge based structures. On the other hand, CL must fit with the FIPA specifications, so a FIPA compliant tool was used for the development and deployment of this part of the architecture. The tool that was used was JADE [9]. This integration among different technologies has allowed us to separate the behavior of the agent from the communication issues that allow our agent to fit the FIPA specifications, that its to say, to guaranties the communication with other FIPA compliant agents into the Agentcities.NET network.

4. COORDINATION MECHANISM.

We can classify the coordination mechanisms of the system into two categories: the communication with the personal agents (same for other FIPA-compliant agents accessing the service), and the coordination between the TRACK-R agents in order to get a common complex solution.

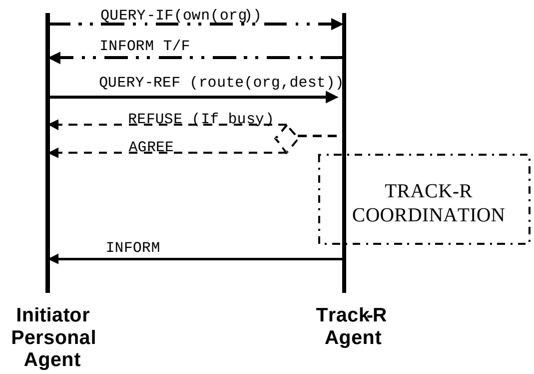

Figure 3. Personal Agent - TRACK-R Coordination

The protocol used in the first case is quite simple (see Figure 3). The first task of the Personal Agent is to localize the TRACK-R agent that owns the origin node. This is important, as we show below, because the solution process is directed by this agent. Once the Personal Agent knows this information, so much asking to the DF than using previous stored information, the Personal Agent perform a Query protocol [7] with this agent, querying for the route between the two nodes. When the solution is generated, it receives the correspondent INFORM message.

Regarding the communication between TRACK-R agents, there are four possible scenarios (see Figure 4), depending on the information the agent that owns the origin node shares and the one who owns the destiny. The first case is the simplest one. It when the agent which owns the origin (OA) and the one of the destiny (DA) overlap it influence areas, and share a node (city or crossroad). The solution then is route that involves this intermediate node. The second possibility is the (empirically) most possible one. It happens when both agents have a road in common. In this case the resultant route should include this a sub-route from the origin to the road, and other from the road to the destiny. The other two possibilities are more complex, because they involve one or more auxiliary agent in order to divide the whole problem in the two previous simpler scenarios. The third one is when there is an intermediate agent that shares some information (roads or nodes) both with origin and destiny agents. The last scenario shows when there is no possibility of finding any intermediate agent with shared knowledge, and several agents must have to linked to obtain a path.

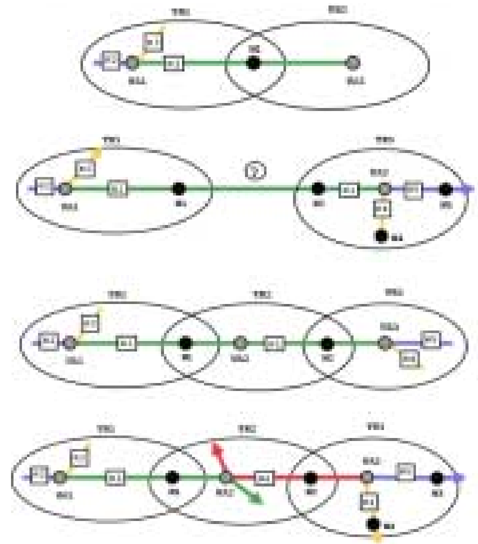

Figure 4. Coordination scenarios.

The coordination protocols between TRACK-R agents are based in the sharing of pieces of knowledge from each agent. This knowledge is represented as prolog predicates or terms, so the communication is based on passing these terms into the ACL messages, included in the content field. This process is coordinated by the OA. It directs the communication, but this role could be assumed by any TRACK-R agent, the one who own the origin node. Therefore, the protocol is not centralized in order that every agent could assume the coordinator role (centralized protocol). In the next figure we can see a representation of this protocol.

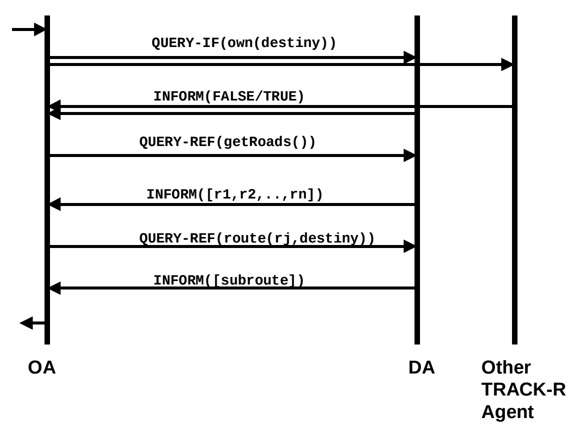

Figure 5. Communication protocol. Case 2.

5. DEPLOYED PROTOTYPE.

A prototype has been developed in order to evaluate the performance of this system. Two TRACK-R agents were implemented: one that represents the roadmap of the region of Madrid, and another for the province of Valencia, both at Spain. Currently we are increasing this prototype by adding a new

TRACK-R agent, for the city of Barcelona and its surrounding cities. For each agent, a different instantiation of the ontology was created. This instantiation reflects a graph that represents the roadmap of each agent. The next figure shows the correspondent graph for the Madrid's TRACK-R agent. The entire roadmap was simplified, because we'd not try to be exhaustive in the solution rather than to prove that our knowledge-based agents solution were correct and efficient.

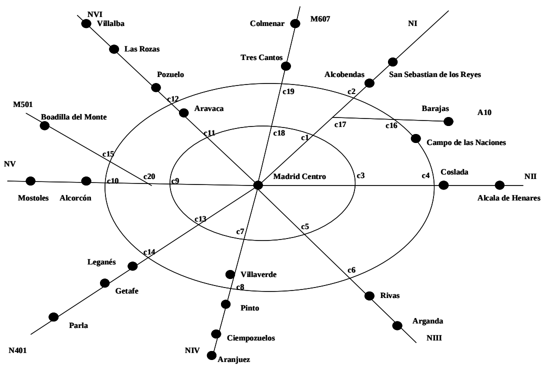

Figure 6. Road-graph for the Madrid's agent

One of the main advantages of this solution was the reduction of the development times. The time expended in the development of the first TRACK-R agent was three times longer than for the second one. Second time was mainly a work of modeling the roadmap and the identification of the kind of coordination protocols needed. Once implemented and tested the system with those two agents, the development of more agents for increasing the coverage of the system is reduced to model and codify (with prolog, that its to say, very few LOCs) the model for the new agents.

Service Type	TRaffic AgentCity for Knowledge-based Recommendation (TRACK-R) v1.0
Description	This service allows the user to get the shortest route between two places into the road map of Madrid/Valencia
Protocols	FIPA-Query Protocol
Agent Communication Languages	FIPA-ACL
Content Languages	Prolog (Ciao Prolog)
Ontologies	Traffic
Message Examples	[illegible]

Table 1. Description of the TRACK-R service

These agents, besides the personal agent for the interaction, were implemented with JADE (v 2.61) and Ciao Prolog (v 1.8p2) and were tested over MS Windows platforms (2000,NT,XP) and Linux (Debian). The previous table shows the description of the system.

6. CONCLUSION AND FUTHER WORK.

By combining traditional rule-based and ontology technologies with FIPA standards the MAS supports a first step of routes recommendation for humans (trough the Personal Agents) and TRACK-R agents. Our work showed that this kind of agent design and development allows a rapid prototype deployment as well as the following comments:

- First results show that the use of Ciao Prolog facilitates the development of mechanisms that support the knowledge-based approach given to our system. Also, the use of JADE helps during the development and deployment of the FIPA-compliant agents makes up the limitations of the Ciao Prolog limitations. The communication management is really facilitated by JADE as well as the supervision and monitoring of the agents' behaviour at run-time.
- The use of an ontology really facilitates the development of new agents for different geographical areas (not just a replication procedure). We are already working in the development of a DAML+OIL version of the ontology to communication standardisation
- The scalability is ensured with the decentralised model for co-ordination of the agents with local knowledge, as proposed in this work. To develop a new agent for the systems is a process that decreases in complexity and cost time.

The new agent developed for Barcelona local area will allow us to test the performance of the different co-ordination mechanisms identified. The HCI features of the personal agent are going to be enhanced with a simple dialogue management (in English and Spanish), and with the JSP features that JADE offers. These enhancements will offer a complex and useful web-based interface for the users, easily portable to mobile devices (phones, PDA's, etc). At this moment we are beginning with a deep analysis and evaluation of this proposal with AgentCities.ES (Spanish funded project [1]).

7. REFERENCES

[1] Agentcities.ES http://www.etse.urv.es/recerca/banzai/toni/AgentCitiesES/

[2] Agentcities.NET http://www.agentcities.org/EUNET

[3] Bueno, P. Correas, J. García-Serrano, A. Teruel, D. "An agent-based architecture for Multi-agent applications". AAMAS Challenges 02. Bologna. Italy. August 2002.

[4] Ciao Prolog. http://www.clip.dia.fi.upm.es/Software/Ciao

[5] García-Serrano, A. Martínez, P. Teruel, D. "Knowledge modeling techniques in the e-commerce scenario" IJCAI 01 Workshop on E-Business and the Intelligent Web. Seattle (USA), August 2001.

[6] Gómez-Pérez, A. Benjamins, R. "Overview of Knowledge Sharing and Reuse Components: Ontologies and Problem-Solving methods" IJCAI Workshop on ontologies and PSM. Stockholm (Sweden) August 1999

[7] FIPA http://www.fipa.org

[8] Hernández, J.Z. Ossowski, S. García-Serrano, A. Journal: Transportation Research, Part C, Pergamon Press, V 10 Issue 5-6, pp:473-506, 2002

[9] JADE http://sharon.cselt.it/projects/jade