ISSN:0975-9646

V.Divya et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 3 (1) , 2012, 3006 - 3011

Pondering of agent technologies in road transportation system and their critical issues

V.Divya[#1], M.Hema Madhuri[#2], Movva N.V Kiran Babu[#3]

[#1]*Assistant professor, Department of CSE, K L University, Andhra Pradesh.*

[#2]*M .Tech student, Department of CSE, K University, Andhrapradesh.*

[#3]*Associate Professor, Department of CSE, Mother Theresa Institute of Science &Technology,Sathupally,AP.*

***Abstract*— Traditional IT systems are not efficient to interact with dynamic environment. They are error prone. In contrast intelligent agents has proven to be well suited for applications requiring, complex interactions with dynamic environment. But there are some critical issues in developing agent based traffic control and management system such as interoperability, flexibility, extendability.In this paper we discuss these issues and several agent technologies developed by various organizations.**

***Key words*— Intelligent Agent, agent computing, mobile agent systems**

I. INTRODUCTION

Agent can be defined to be independent, problem solving computational entities capable of effective operation in dynamic and open environment[1] .agents are often deployed in environment in which they interact, and work together with other agent including people and software they have possibly contradictory aims such environments are known as the multi agent systems .agents can operate without the direct involvement of humans and others.[2] Agent can be used as a design symbol for designers and developers in the way of structuring an application around autonomous, communicative elements, and elements, and lead to the constructing of software tools and infrastructure to support the design symbol. In this sense, they offer a new and often more appropriate route to the development of complex systems, especially in open and changing environments. Agent technologies span a range of specific techniques and algorithm for dealing with interactions with others in changing and open environment. These include issues such as balancing reaction. Agent-based systems are one of the most effective and important areas in research and development to have emerged in information technology in the 1990s.[3]an agent is a computer system that is capable of flexible autonomous action in dynamic, unpredictable, typically multiage domains the characteristics of dynamic and open environments in which, for example, heterogeneous systems must interact, span organizational boundaries, and operate effectively within rapidly changing circumstances and with dramatically increasing quantities of available information, suggest that improvements on traditional computing models and paradigms are required .thus the need for some degree of autonomy, to allow components to respond dynamically to changing circumstances while trying to achieve over-arching objectives, is seen by many as fundamental. Many observers therefore believe that an agent represents the most important new paradigm for software development since object orientation. The concept of an agent has found wide range of sub-disciplines of information technology, including computer networks, software engineering, artificial intelligence, human-computer interaction, distributed and concurrent systems, mobile systems, telematics, computer-supported cooperative work, control systems, decision support, information retrieval and management, and electronic commerce. In practical developments, web services, for example, now offer fundamentally new ways of doing business through a set of standardized tools, and support a service-oriented view of different and independent software components interacting to provide valuable functionality. In the context of such developments, agent technologies have gradually come to the foreground. Because of its horizontal nature, it is likely that the successful adoption of agent technology will have a profound, long-term impact both on the competiveness and capability of IT industries, and on the way in which future computer systems will be conceptualized and implemented. Many researchers and programmers see agents as programs roaming a network to collect business-related data in order help users to buy goods, or implement platform independent code-on-demand, for example this need for mobile agents is acknowledged, and builds on

European strengths, but mobility brings added security problems. The research effort concentrates on how to guarantee termination, security or exactly-once protocols. To protect against malicious hosts, agents should contain time limit validity, and electronic money with an expiration date. A key issue that needs to be addressed here is administrability of mobile agent systems, e.g., authorization policies; this has been a major reason why mobile agents have not yet been taken up by the mainstream. Note also that hosts need to be protected as well as agents End users already encounter the situation that, while ample bandwidth is available on the backbones of network service providers, their experience is limited by the constraints of the infamous last mile. Mobile agents may improve the end user experience by offloading application-specific filtering, media adaptation, and other pre-processing to a node with high bandwidth connectivity. This is particularly interesting for mobile phones and portable devices. One of the commercial application areas in which the added value of mobile agents is very high, is large-scale distributed or decentralized system integration with highly adaptive and dynamic business logic. Existing solutions are generally centralized, pulling everything onto one platform, limiting the complexity and changes that can be handled. A decentralized agent approach divides and conquers complexity by pushing a large part of the business logic out onto source systems so that much monitoring and aggregation can be done on each. This distributes workload and increases robustness because the local processing can be performed independently of other systems, resulting in fewer and more relevant interactions with these systems, at a higher level of abstraction. In turn, mobility, mainly single hop, is the answer to the increasing need for flexibility and adaptability in business logic. Agents can easily be deployed to source systems, carrying new database drivers, code to interact with new application or file types, or new data processing rules. Software is updated at the component-level, at runtime, proving a level of dynamism and flexibility that goes far beyond current release policies. Agent communication and behavior capabilities complete the picture, being very well suited to high-level service-based Interactions, the decentralized implementation of business logic, and for adapting and handling change in their environment. A nice property of the dynamic, component-level approach is that it naturally fits step-by-step system integration, with each step resulting in added value for the business. This is a particularly significant advantage in the current economic climate, in which many companies have seen mega-projects fail. For example, Global IDs Inc in the US offers a next-generation product suite for data integration based on the Tryllian mobile agent platform. Their data integration products are capable of simultaneously monitoring many hundreds of enterprise systems for relevant changes in data or metadata, by deploying mobile agents onto those systems. The agents tap into local databases or applications, keep track of changes, can pre-process data and only forward relevant events or structured derived data to centralized collectors – in real time if required. The mobility of the agents allows highly customized functionality, which can be dynamically updated. Thus, the business user can change the business rules that are being executed at any point in time, while only relevant drivers and adapters are transferred to a source system. Agents can assess the impact of changes in the business rules and handle that impact throughout the integration process.

II. CRITICAL ISSUES IN DEVELOPING AGENT BASED TRAFFIC CONTROL AND MANAGEMENT SYSTEM

In this paper, we address three issues: ability to handle uncertainty, interoperability, and extensibility of agent-based distributed traffic management systems.[4]

A. Extensibility

First, the reported agent-based applications in roadway traffic management focus on developing multiagent systems (MASs) that consist of multiple distributed stationary agents. The use of multi-agent systems provides a clear added value of high degree of autonomy and co-operability to conventional systems. However, MASs have a limited ability to deal with uncertainty in dynamic environments. To overcome this weakness, we propose to integrate mobile agent technology with multi-agent systems to enhance the flexibility and adaptability of large scale traffic management systems. Different from stationary agents, mobile agents are able to migrate from one node in a network to other nodes and to be executed on any nodes in the network. Mobile agents can be created dynamically at runtime and dispatched to destination systems to perform tasks with most updated code and algorithms. Mobility provides great opportunities to address challenges in traffic control and management systems, such as quick incident diagnosis, dynamic system configuration, deploying new algorithms or operations dynamically, taking unanticipated actions, and reducing data transmission over a network.

B. Interoperability

Second, most existing agent-based traffic management systems are lack of taking consideration of the interoperability between agent systems at the agent platform level. The interoperability is critically needed in making decisions based on information across systems, organizational and jurisdictional boundaries. To tackle interoperability issue, IEEE FIPA (Foundation for Intelligent Physical Agents), a consortium of companies, government agencies, and schools, has been working on producing software standards for heterogeneous and interacting agents and agent-based systems. The goal of FIPA standards is to guarantee the interoperability between agents by coordinating different aspects of systems, including system architecture, agent communication, agent management, and agent message transportation.

C. Scalability

Third, less attention has been paid on the openness and scalability design of systems, which is important for system extension.

III. EXISTING AGENT TECHNOLOGIES

Several agent technologies developed by various organizations which we will discuss in this section. The various agent technologies are TRACK-R, Mobile-C, In TRYS and TRYSA2, aDAPTS

A.TRACK-R

Multiagent systems (MAS) are becoming, in the last few years, a growing technology for developing distributed services [5]. Many Organizations and companies are adopting this technology to its own development and research processes one of the reasons of this increase is the creation of large-adopted standards. These are the FIPA specifications [6]. 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 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 traffic ontology [7]. 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. The **TRACK-R** (*Traffic Agent City for Knowledge-based Recommendation*) is a multiagent application, enclosed into the Agentcities.NET project 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) 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.

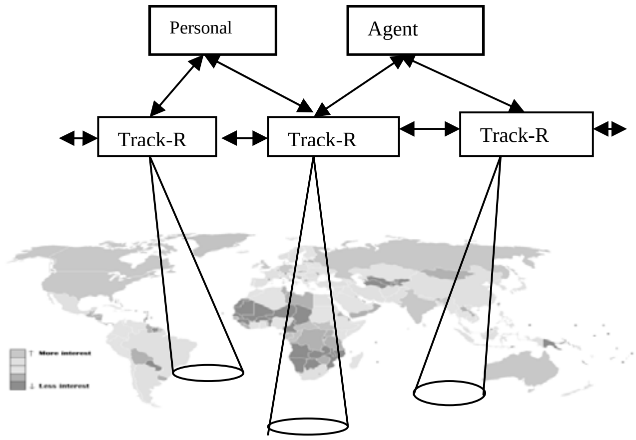

Fig 1: scenario of TRACK-R application

B .Mobile-C

Mobile-C an IEEE the Foundation for Intelligent Physical Agents (FIPA) compliant agent platform, and its application in real-time traffic detection and management systems [8]. FIPA is one of two international agent standards, FIPA and OMGs Mobile Agent System Interoperability Facility (MASIF) .The strength of FIPA standards is that it promotes the interoperation of agents and agent systems across heterogeneous agent platforms. To support this, FIPA has been working on specifications that range from agent platform architecture to support interagent Communication, communication languages and content languages for expressing exchanging messages, ontologies to define semantic contents of messages and interaction protocols that direct agent communication through an admissible sequence of messages between agents. FIPA has been increasingly accepted in the agent community, and the compliance with FIPA standards has been recognized as a crucial property for the interoperability of agents. The development of Mobile-C is primarily motivated by applications that contain low-level hardware, such as detection systems and control systems. Since most of these types of systems are written in C/C++, Mobile-C chooses C/C++ as the mobile agent language for easy interfacing with control programs and underlying hardware. Mobile-C extends FIPA standards and integrates an embeddable C/C++ interpreter into the agent platform as a mobile agent execution engine. An agent mobility protocol has been designed to regulate the entire migration operation. Mobile agent migration in Mobile-C is achieved through FIPA agent communication language (ACL) messages. Using FIPA ACL messages for agent migration in FIPA compliant agent systems simplifies agent platform, reduces development Effort and easily achieves inter-platform migration through well-designed communication mechanisms provided in the agent platform. Messages for agent communication and migration are expressed in FIPA ACL and encoded in XML. Mobile-C has been used to simulate highway traffic detection and management for Intelligent Transportation Systems (ITSs). Agent-based real-time traffic detection and management system is a decentralized, autonomous, interoperable and scalable system. The system includes both stationary agents and mobile agents. The stationary agents are usually located at distributed detection stations and transportation management center (TMC). The communication between agents allows detection stations to cooperate with each other to perform distributed real-time traffic information fusion, which will dramatically reduce data transmission and the response time to incidents. The system also used mobile agents to perform unanticipated actions by dynamically deploying new algorithms and code.

1) The System Architecture of Mobile-C: The system architecture of Mobile-C in the fig2. Agencies are the major building blocks of the system and are installed in each node of Mobile-C. They are the actual runtime environment for stationary agents (SA) and mobile agents (MA). They also serve as 'home bases' for locating and messaging mobile and detached agents, collecting knowledge about a group of agents and providing an environment in which a mobile agent executes. The core of an agency is the agent platform, which provides local services for agents and proxies to access remote agencies. An agent platform represents the minimal functionality required by an agency in order to support the execution of agents. The main functionalities of an agent platform can be summarized as follows.

• Agent Management System (AMS): AMS manages the life cycle of agents. It controls the creation, registration, retirement, migration and persistence of agents. AMSmaintains a directory of Agent Identifiers (AID), which contains transport addresses (amongst other things) for registered agents. Each agent must register with an AMS in order to get a valid AID.

• Agent Communication Channel (ACC): ACC routes messages between local and remote entities, realizing messages using FIPA ACL. This service is responsible for all of the remote interactions that take place between the distributed components, such as inter-agent communication and interplatform agent migration. All of the interactions can be performed via ACL message exchange.

• Agent Security Manager (ASM): ASM is responsible for maintaining security policies for the Platform and infrastructure, such as communication and transport-level security.

• Directory Facilitator (DF): DF serves yellow page services. Agents in the system can register their services with DF for providing to the community. They can also look up required services with DF.

• Agent Execution Engine (AEE): AEE serves as the execution environment for the mobile agents. Mobile agents must reside inside an engine to execute. AEE has to be platform independent in order to support a mobile agent executing in a heterogeneous network.

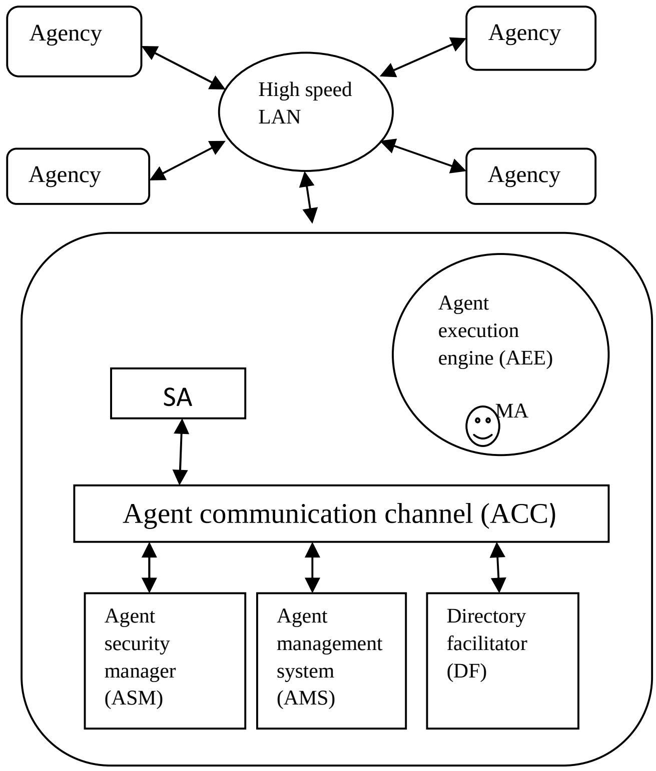

Fig 2: the system architecture of Mobile-C.

C. *TRYS and TRYSA2*

TRYS and TRYS are two autonomous agents, two multiagent systems that perform decision support for real-time traffic management in the urban motorway network around Barcelona [9]. Both systems draw upon traffic management agents that use similar knowledge-based reasoning techniques in order to deal with local traffic problems. In this section the applications of multiagent architectures to knowledge-based ITMS. In the first place, the TRYS traffic management generic architecture is outlined. Subsequently, the knowledge-based reasoning model of TRYS traffic management agents is outlined.

TRYS architecture overview

TRYS is an agent-based environment for building ITMS applications for motorway networks. It provides a generic and modular knowledge model supporting an intelligent reasoning layer that can complement conventional traffic management application capabilities. The TRYS approach has been applied to develop several ITMSs, all of them installed and tested on-line in TCCs, giving rise to the so-called TRYS family of systems. Two relevant Members of this family are the integrated TRYS (InTRYS) and TRYS autonomous agents (TRYSA2) applications.

InTRYS is the result of the technical work oriented to integrate the TRYS environment as part of the existing traffic management infrastructure in the TCC in Barcelona. Conceptually, the main difference between the original TRYS approach, applied for instance in Madrid, and the InTRYS one lies on the bigger complexity of the Barcelona network with sideways parallel to the main road along the whole ring road supervised and including ramp metering devices. TRYSA2 is a decentralized multiagent system for traffic management that targets the same problem domain as InTRYS. Contrasting InTRYS, TRYSA2 has been developed as an experimental prototype only for laboratory purposes, in order to gain experiences with decentralized multiagent architectures for their future industrial application. the InTRYS system relies on a set of 18 knowledge-based traffic control agents, each responsible for traffic management in one such area. This number of agents was obtained considering both senses of traffic for every major adjacent road, and a partition of the Barcelona ring road in four parts the network areas assigned to the agents are control dependent, as some agents have to share control devices. The InTRYS system faces this problem by means a special coordinator agent endowed with knowledge on how to integrate local control proposals into a coherent global signal plan for the whole traffic network. It receives local control proposals from the traffic control agents, resolves conflicts between them, and sends the resulting globally consistent local signal plans back to the traffic agents. In TRYSA2, 11 spatial problem areas are controlled by autonomous (self-interested) traffic agents that coordinate laterally, based on a mechanism called structural cooperation Control devices ‘‘belong’’ to certain agents, and the corresponding mutual dependence (agents may exchange favors respecting the use of ‘‘their’’ devices) provides a potential for cooperation. Normative prescriptions permit or forbid the use of some devices for certain agents, thus biasing agent interactions and influencing self-interested agent behavior, so as to make
It functional with respect to the desired traffic management functionality The goal of a traffic management agent is to provide two types of information: (i) diagnosis of the traffic problems present in a local area together with an explanation justifying such a diagnosis, and (ii) proposed control actions for the available signal devices to improve traffic conditions using the diagnosis information. In order to achieve this goal agents are endowed with four main types of knowledge: possible traffic problem scenarios, potential traffic control proposals, and regarding traffic behavior in the network, the network structure and historic traffic demand

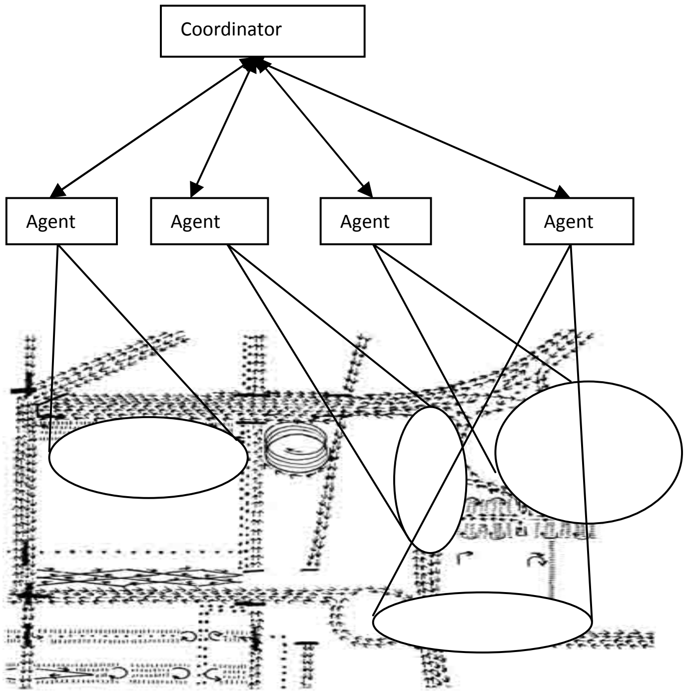

Fig 3.1 Centralized (InTRYS)

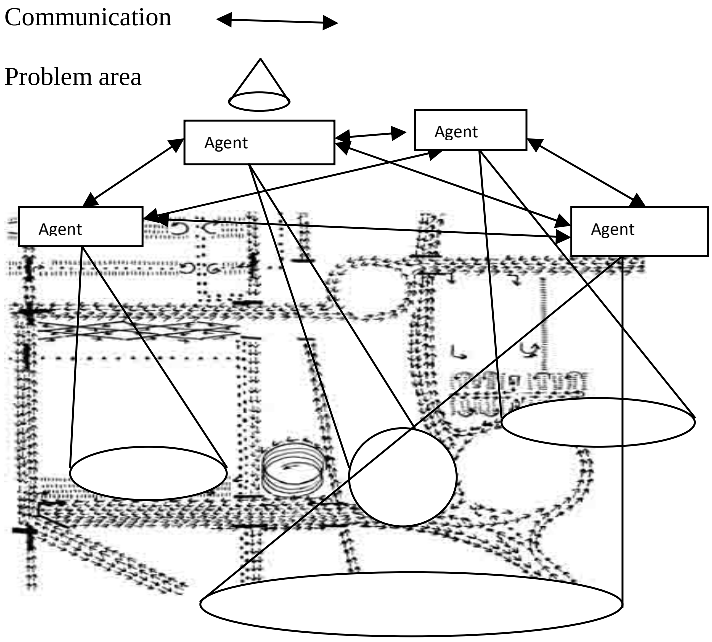

Figure 3.2 decentralized coordination (TRYSA2)

aDAPTS

In the control and management mode, the Adapts (Agent-Based Distributed and Adaptive Platform for Transportation Systems) system provides environments for designing, constructing, managing, and maintaining autonomous-agent programs for\ traffic tasks and functions. Those agents are delivered to traffic control centers, roadside controllers, sensing devices, and information systems via communication networks to make correct decisions and collect the correct information at the correct times [10]. An agent based Networked traffic-management system. The agent-based control decomposes a sophisticated control algorithm into simple Task-oriented agents that are distributed over a network. The ability of dynamically deploying and replacing control Agents as needed allows the network to operate in a "control on demand" mode to adapt to various control scenarios. The System architecture employs a three-level hierarchical architecture. The highest level performs reasoning and planning of task Sequences for control agents; the middle level dispatches and coordinates control agents; and the lowest level hosts and runs Control agents. The control agents are represented by mobile agents that could migrate from remote traffic control centers to Field traffic devices or from one field device to another. A testbed to allow designers of MASs to experiment with different strategies and examine the applicability of developed systems [11]. The testbed consists of intelligent models for modeling intelligence of agents, a world model for representing traffic process, and an interaction model for modeling the interactions between agents. The communication in the testbed conforms to the FIPA standards.

IV. CONCLUSIONS

Agent technologies play a vital role for developing traffic control and management systems in road transportation. This paper provides a review on several existing agent technologies with the prominence of critical issues like, interoperability, flexibility, extendability.Finally we conclude that, by understanding these issues one can modify the existing techniques or can design the new agent technology.

ACKNOWLEDGEMENTS

We like to express our gratitude to all those who gave us the possibility to carry out the paper. We would like to thank Mr.K.Satyanarayana, chancellor of K.L.University, Dr.K.Raja Sekhara Rao, Dean, and K.L.University for stimulating suggestions and encouragement. We have further more to thank Prof.S.Venkateswarlu, Dr.K.Subrahmanyam, who encouraged us to go ahead with this paper.

REFERENCES

[1] Agent Technology: Enabling Next Generation Computing: A Roadmap for Agent-Based Computing by Michael Luck, Peter McBurney and Chris Priest January 2003, Version 1.0, ISBN 0854 327886

[2] Bo Chen, *Member, IEEE*, and Harry H. Cheng, *Senior Member, IEEE,"* A Review of the Applications of Agent Technology in Traffic and Transportation Systems" IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, VOL. 11, NO. 2, JUNE 2010.

[3] Agent Technology: Computing as Interaction: A Roadmap for Agent Based Computing by Michael Luck, Peter McBurney, Onn Shehory, Steve Willmott and the AgentLink Community, © September 2005, AgentLink III ISBN 085432 845 9

[4] B. Chen, H. H. Cheng, and J. Palen, "Integrating mobile agent technology with multi-agent systems for distributed traffic detection and management systems," Transp. Res. Part C: Emerging Technol., vol. 17, no. 1, pp. 1–10, Feb. 2009.

[5] A. M. Garcia-Serrano, D. Teruel Vioque, F. Carbone, and V. D. Mendez, "FIPA-compliant MAS development for road traffic management with a Knowledge-based approach: The TRACK-R agents," in Proc. Challenges Open Agent Syst. Workshop, Melbourne, Australia, 2003.

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

[7] 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.

[8] B. Chen, H. H. Cheng, and J. Palen, "Mobile-C: A mobile agent platform for mobile C/C++ agents," Softw.—Pract. Exp., vol. 36, no. 15, pp. 1711 -1733, Dec. 2006.

[9] J. Z. Hernandez, S. Ossowski, and A. Garcia-Serrano, "Multiagent architectures for intelligent traffic management systems," Transp. Res. Part C: Emerging Technol., vol. 10, no. 5/6, pp. 473–506, Oct.–Dec. 2002.

[10] F. Y. Wang, "Toward a revolution in transportation operations: AI For complex systems," IEEE Intell. Syst., vol. 23, no. 6, pp. 8–13, Nov. /Dec. 2008

[11] R. T. Van Katwijk, P. Van Koningsbruggen, B. De Schutter, and J. Hellendoorn, "Test bed for multiagent control systems in road traffic management," Transp. Res. Rec., vol. 1910, pp. 108–115, 2005.