A multiagent architecture for bus fleet management

Alberto Fernández[1], Eduardo Alonso[2], Sascha Ossowski[1]

[1]AI Group, University Rey Juan Carlos, Campus de Móstoles s/n, E-28933 Madrid, Spain
{al.fernandez, s.ossowski}@escet.urjc.es
[2]agents@city Group, City University, Northampton Square, London, EC1V 0HB, UK
eduardo@soi.city.ac.uk

Abstract. Nowadays, telematic infrastructures allow operators in a control centre to monitor the state of the public transport network and to maintain a good service quality. But the huge quantity of data arriving at a control centre makes difficult for operators to process it in real time. In this paper we propose a multiagent based decision support system to assist operators in bus fleet management. The system will be applied to a real-world scenario in Malaga (Spain).

1. Introduction

Advances in telematic infrastructures are currently influencing in almost all modalities of transport. Typically, information about the state of a transport network is provided to a control centre. There, operators are responsible for maintaining a reasonable quality of service. The ever increasing amount of data to be processed in a short period of time adds to the complexity of the decision problems that operators face, and suggests the use of complex intelligent systems to assist them in taking "good" management actions in-time.

In past years, multiagent systems [12, 18] have been successfully applied in building distributed intelligent systems [10]. The complexity of the system, its efficiency, geographical distribution and scalability are some of the reasons that lead to construct a system as distributed.

Intelligent systems have been applied to the field of individual transport management for many years [4, 11, 16, 17]. More recent systems often take advantage of a distributed architecture [9]. In this paper we focus on the problem of public transport. Some works have been done in this field recently [8]. We propose a multiagent architecture for real-world bus fleet management systems.

This paper is organised as follows. In section 2 we briefly describe the domain of bus fleet management. The architecture of the system is presented in section 3. Several examples of interaction among agents are shown in section 4. We finish with some conclusions and further work.

2. Bus Fleet Management Domain

Vehicles in modern buses fleet are equipped with devices (GPS, radio…) providing information to control operators in a Bus Fleet Management Centre (BFMC) that enables them to determine the current location of buses, estimate time of arrival at bus stops, etc., so as to take the necessary actions to maintain an appropriate quality of this public transport service. Currently, the management is performed by means of an Exploitation Support System (ESS), which provides basic information concerning the status of the buses with regard to the scheduled service.

A typical task of BFMC operators is to detect problems by comparing the timetable of all lines and buses with current data. The most frequent incidents that can be detected in this way fall into the following categories: *individual delay* (one of the buses in a line is delayed), *generalised delay* (several buses are delayed), *advance* (a bus arrives at a stop before the expected time), *individual saturation* (some people cannot take the bus because it is full), *generalised saturation* (several buses on a line are full) or a *breakdown* of a bus. When such incidents occur, operators need to devise a management plan, containing a set of control actions for the drivers, with the aim of minimising the impact of these problems on the overall quality of service. Examples of control actions include *increase/reduce speed* (of an individual bus), change *timetable regulation* (each bus must arrive at the stop at a fixed time) to *frequency regulation* (a bus must arrive a stop every x minutes) or vice versa, *change frequency of regulation*, *timetable rotation* in a line (each bus in a line adopts the scheduled timetable of its successor), *reinforce* a line with a reserve bus (or a bus from another line), etc. It is obvious that the less time the operator wastes in taking such management decisions the less will be the impact the incident on the transportation service from the point of view of the passengers.

Due to the quantity of data that operators receive and the complexity of their reasoning, a Decision Support System (DSS) [3] is necessary to ensure a real-time response of sufficient quality. Modern agent-based DSS can be conceived as Intelligent Decision-making Assistants (IDEAS) [15] that render support to their human operators during the various stages of their decision-making process by means of flexible dialogues. It is essential to notice in this respect that a DSS is not a substitute of the operator, but a tool that helps her to better understand the meaning of the data, and to explore potential consequences of her control actions. The final decision (and the responsibility for it) stays with the operator. As a consequence, it is important that the DSS be able of explaining its reasoning in the course of a decision support conversation, so as to increase the operators' confidence in the control proposals.

In the next section a multiagent architecture for bus fleet management is presented.

3. Multiagent architecture

Fig 1 depicts the proposed multiagent architecture for bus fleet management. Its design was driven by the aim to take advantage of the proper distribution of the domain, organisational, spatial as well as functional, while being compliant with the FIPA agent standard[1] [6]. Four categories of agents have been identified: bus fleet management, control centre, bus fleet connection and external agents. *Bus fleet management* agents are the agents in charge of the management. The *control centre group* facilitates communication between the operators and the system. *Bus fleet connection* includes agents that directly communicate with the infrastructure (vehicles, stops...). Finally, by *external agents* we refer to those agents that assist other agents in the system in executing their tasks. These external agents can be re-used in other systems.

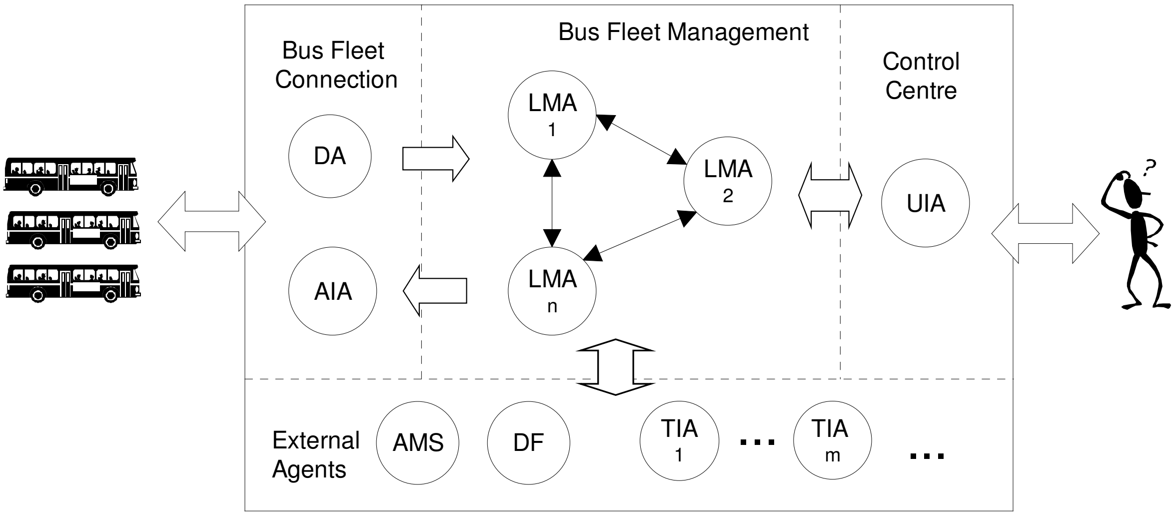


Fig. 1. Bus fleet management agent architecture

3.1. Bus fleet management agents

This category is composed of several *line management agents (LMA)*. Each *LMA* is in charge of the supervision of a line (or a set of lines depending on their complexity). It carries out the four basic tasks of a Decision Support System [14]: problem identification, diagnosis, action planning and prediction.

This agent knows the timetable of the line(s) that it manages. It also receives information each time a bus arrives at a stop, so that it is always monitoring the line. When a problem is detected, a warning message is *proactively* sent to the *user interface agent (UIA)*. Alternatively, it can detect a problem in advance, for instance when traffic information is provided by the *traffic information agent (TIA)*. A particularly interesting problem arises when passengers can miss a connection due to a delay. This kind of problem can be detected by communicating with other *LMAs*. Problem identification is performed by matching the current data against a knowledge base of rules. In Fig. 2, a partial knowledge base for problem identification is shown. The knowledge base is written in CLIPS [2]. In the figure two rules are shown. The first one is used to abstract different levels of severity of individual delays; in particular to identify a medium degree of severity

[1] We follow the FIPA standard. For the development of the proposed system the FIPA compliant platform JADE [1] is being used.

when the delay is between 5 and 10 minutes. The second one is used to detect a generalised delay in a line (two or more buses are delayed).

```
...

(defrule Individual_Delay_Medium
   (bus (id ?b) (delay ?m) (line ?l))
   (test (>= ?m 5))
   (test (< ?m 10))
=>
   (assert (individual_delay (bus ?b) (severity medium)))
)
...
(defrule Generalised_Delay
   (bus (id ?b1) (delay ?m1) (line ?l))
   (bus (id ?b2) (delay ?m2) (line ?l))
   (test (neq ?b1 ?b2))
   (test (> ?m1 0))
   (test (> ?m2 0))
=>
   (assert (generalised_delay ?l))
)
...
```

Fig. 2. Partial knowledge base for problem identification

When the operator perceives a potential problem, she can access the agent also *reactively*, by requesting a diagnosis of such a problem through the *UIA*. The *LMA* can also assist the operator in planning control actions in order to minimise the impact of the detected problems on the overall quality of the service. A knowledge base of rules can be applied here to relate problems and control actions. The operator can request a prediction of the effects of the proposed actions on the network.

Some control actions may require co-ordination among agents [13], for instance, when an extra bus must be introduced in a line to reduce the impact of a problem. This is the case, for instance, in our trial scenario where reserve buses do not exist[2]. Instead, a bus is moved from one line to another. Thus, *LMAs* should negotiate and decide which one lends a bus to the line in trouble.

This agent not only provides a bus line management service but also information about the line that it manages (buses, stops, timetables...) that might be useful to other agents in the system.

3.2. Control centre agents

The *UIA* allows the operators to communicate with the rest of the system. It shows the location of each bus. In order to do this, it needs to be subscribed to an information service about bus arrivals. Moreover, the *UIA* warns users when a problem is detected via a *LMA*. Through the interface, the operator can request diagnosis, planning and estimation. All these services are provided by the *LMA*.

As the system acts as an assistant, *UIA* must provide a flexible and adaptive communication channel with the operators. For example, suppose that in order to solve a problem in a line a bus must be moved from another line. In this case, operators responsible for the lines should negotiate to decide which one lends the bus. The system might assist at different levels: simply recommending negotiation, assisting during the negotiation or even carrying out the negotiation automatically. This type of communication may be based on human-computer dialogues. Recent results in the field of intelligent user interfaces applied to transport management can be applied [5,8].

3.3. Bus fleet connection agents

Modern bus fleet infrastructures involve a set of devices that provide information to the control centre. This information is received from different kinds of heterogeneous information sources (radio, SMS, GPS...). This amount of data is introduced in the system by the *data agent (DA)*. This agent encapsulates the different information sources and distributes them in a coherent way. Different agents can subscribe to the information provided by the *DA*. For instance, when a bus gets at stop, this is notified to the *LMA* of the corresponding

[2] The bus fleet management system is to be applied to part of the bus network of the *Empresa Malagueña de Transportes* (EMT)

line and to the *UIA*. So, agents subscribe to those *DA* information service in which they are interested (e. g. arrivals/departures, broken-down, single delays...).

In addition to *DA*, the *action implementation agent (AIA)* communicates with the bus fleet so that the actions planned to reduce the impact of the identified problems are carried out; for instance, sending orders to buses by SMS or radio messages.

3.4. External agents

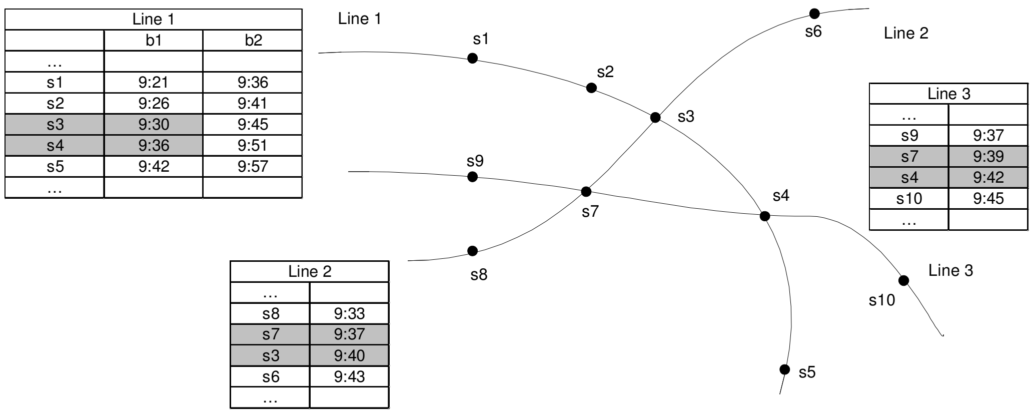
These agents are not responsible for bus fleet management but can participate in it. In our proposal we use FIPA's *agent management system (AMS)* and *directory facilitator (DF)* [6], and *traffic information agents (TIA)* as external agents. *AMS* and *DF* are mandatory in any FIPA compliant platform. They are white and yellow pages services respectively. Each agent in the system registers with an *AMS*. At the same time, each agent will advertise the services it provides via the *DF*.

TIAs, on the other hand, are external agents, which provide information about an area of the city. The way these agents work can range from a person looking at the traffic through a camera to a more sophisticated real time management system [4, 9]. These agents, in turn, can get information from the bus fleet management system by analysing how long it takes for the buses to cover the distance between several stops.

As we have described, agents need to communicate with each other to carry out their tasks successfully. In the next section we show several examples of interaction.

4. Examples of interaction among agents

In this section we will use the example depicted in Fig. 3 to illustrate some typical interactions among agents in our system. In this figure three bus lines are shown. Black dots represent stops. Furthermore, a partial timetable for each line is shown. b_i stands for bus i and s_i for stop i. For *line 1*, two consecutive services have been shown.



Line 1		
	b1	b2
…		
s1	9:21	9:36
s2	9:26	9:41
s3	9:30	9:45
s4	9:36	9:51
s5	9:42	9:57
…		

Line 2	
…	
s8	9:33
s7	9:37
s3	9:40
s6	9:43
…	

Line 3	
…	
s9	9:37
s7	9:39
s4	9:42
s10	9:45
…	

Fig. 3. Example of three interrelated bus lines and their bus stops

Table 1 explains some of the messages[3] used in the figures of this section.

Table 1. Messages used in figures

arrival(b,s)	bus *b* arrives stop *s*
delay(b,t)	bus *b* is *t* min delayed
generalised delay(l)	there is a generalised delay problem line *l*
recommendation?	Can you recommend any control action?
timetable rotation	A *timetable rotation* action is recommended
connection problem(l,s)	There is a connection problem in stop *s* of line *l*
dep(s,i)?	Is there any departure from stop *s* in the interval *i?*

[3] We will use FIPA ACL as the agent communication language. For the shake of clarity in the figures syntactic sugars will be used though

Example 1.

We will illustrate what happen when a bus arrives at a stop following the *FIPA-subscribe-protocol* [7], shown in Fig. 5. Following this protocol, in a first phase, LMA_1, UIA and TIA_2 (playing the role of *initiator*) subscribe to the *arrival information service* for bus b_1. This service, provided by *DA,* as the *participant*, *informs* them every time bus b_1 arrives at a stop. In the example of Fig. 4, bus b_1 arrives at stop s_1, thus a message is sent to LMA_1, UIA and TIA_2.

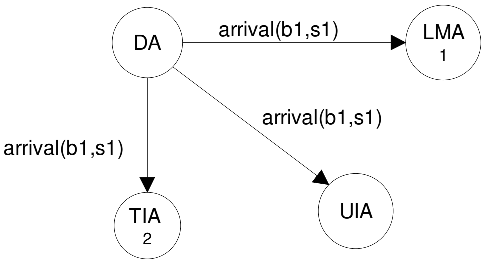


Fig. 4. Arrival of bus b_1 at a stop s_1.

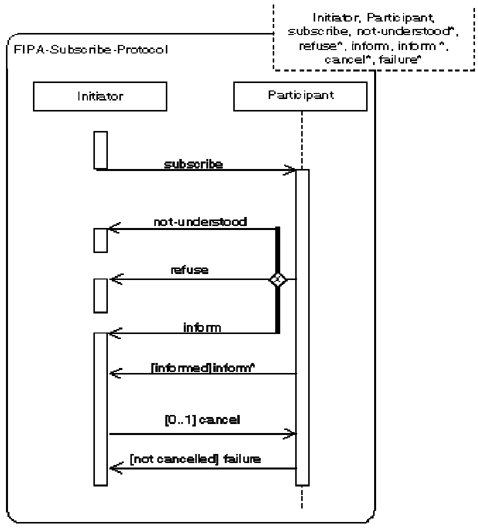


Fig. 5. The FIPA-subscribe-protocol

Example 2.

Suppose that *line 1* is running as scheduled. Now, bus b_1 arrives at stop s_4 5 minutes late, at 9:41. In this situation, *DA* sends a message to LMA_1 notifying that bus b_1 is 5 min delayed[4]. Later on, at 9:44, b_2 arrives at stop s_1 (8 minutes late). In this occasion, LMA_1 doesn't send a message to *UIA* notifying that bus b_1 is 8 min delayed. Instead, LMA_1 analyses the global situation of the line. LMA_1 then realises that there are two buses delayed. Consequently, a message of *generalised delay* in the *line 1* is sent to *UIA*. The user asks for an explanation and is told by LMA_1 that there are two buses (b_1 and b_2) delayed. Then, the operator asks for a control action recommendation. LMA_1 suggests doing a *timetable rotation*. Finally, the operator orders this action to be executed by the *AIA*. This interaction is shown in Fig. 6.

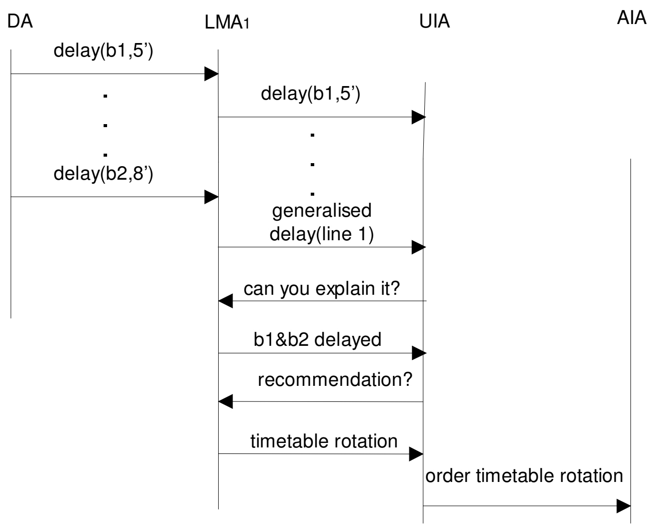


Fig. 6. Problem identification and control action recommendation

[4] In our case delays are detected by the Exploitation Support System, that is encapsulated by *DA*

Example 3.

Suppose that bus b_1 arrives at 9:29 at stop s_1 (8 minutes late). As a consequence arrival messages are sent by *DA*. Moreover, the *DA* submits a message to LMA_1 notifying this delay (see Fig. 7). LMA_1 then has to identify problems that might derive from such a delay. For example, LMA_1 will check whether or not there is a *connection problem* with other lines, i. e. passengers travelling in bus b_1 will likely miss another bus due to the b_1's delay. LMA_1 knows what stops are shared with others lines, therefore it can explore some stops ahead for possible problems. In our example, b_1 is scheduled to arrive at s_3 at 9:30 (s_2 is not shared). Therefore LMA_1 asks LMA_2 whether there will be any departure between 9:30 and 9:38, when b_1 is expected to arrive. No departure is expected during this interval in *line 2*. Then LMA_1 asks LMA_3 the same question, but now taking into account the arrival time at s_4 (9:36 in the example). In this case, there is one departure scheduled for 9:40. A *connection problem*, some passengers potentially will miss another bus, is identified.

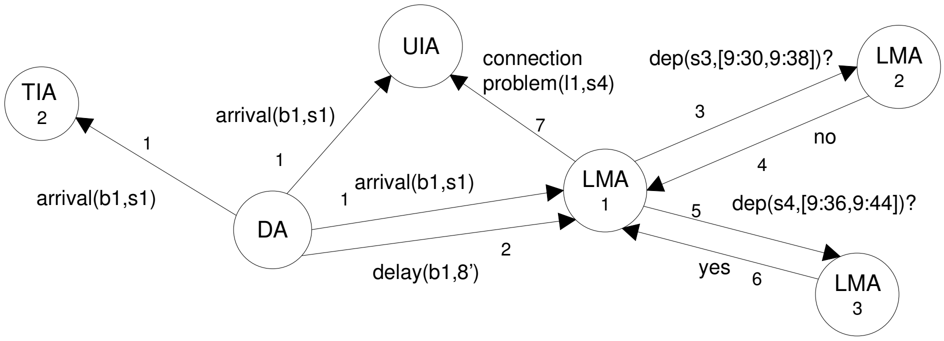


Fig. 7. Connection problem detection

5. Conclusions and further work

At present, telematic infrastructures allow operators in a control centre to monitor the state of the network and to maintain a good quality of the service. However the complexity of the data that arrive at the control centre makes it difficult for operators to process them. Therefore, the use of information systems is necessary in order to manage the resource efficiently. In this paper we propose a multiagent based decision support system to assist operators.

Currently we are applying this architecture in the construction of an intelligent system for bus fleet management, that will be evaluated with data from the EMT bus network of Malaga. JADE [1] is used as the agent platform for the development of the proposed system.

We think that multiagent systems can also be applied to other types of public transport management. In particular, we plan to extent the proposed architecture for rail management, using as example the city of London. These works should aim to an integrated MAS based solution to different kinds of public transport (bus, train, underground,...) management.

6. References

1. Bellifemine, F.; Poggi, A.; Rimassa, G.; Turci, P.: An Object Oriented Framework to Realize Agent Systems. Proceedings of WOA 2000 Workshop, Parma, May 2000, p. 52-57.
2. CLIPS. http://www.ghg.net/clips/CLIPS.html
3. Cuena, J.; Hernández J.: An Exercise of Knowledge Oriented Design: Architecture for Real Time Decision Support Systems. In Knowledge Based Systems: Advanced Concepts, Techniques & Applications, S.G. Tzafestas (ed.), World Scientific Publishing Company, pp. 497-524. 1997.
4. Cuena, J.; Hernández, J.; Molina, M.: Knowledge-Based Models for Adaptive Traffic Management Systems. Transportation Research, Part C, vol.3, nº 5, pp. 311-337, Pergamon Press, 1995
5. Cuena, J.; Hernández, J.; Molina M.: Advanced User Interfaces for Decision Support in Real Time Transport Management. 5th International Conference on Applications of Advanced Technologies in Transport Engineering, ASCE'98. Information Technology and Knowledge Systems, pp. 67-76, American Society of Civil Engineering. 1998.
6. FIPA - The Foundation for Intelligent and Physical Agents. http://www.fipa.org/
7. FIPA Subscribe Interaction Protocol Specification, 2000. http://www.fipa.org/specs/fipa00035/

8. Hernández, J.: Intelligent Interactive Knowledge Models to Support Public Transport Management: The FLUIDS Test Case. Rome Jubilee 2000 Conference, 'Improving knowledge and tools for transportation and logistics development', 8th Meeting of the Euro Working Group Transportation (EWGT). Rome, Italy, September 2000.
9. Hernández, J.; Ossowski, S.; Garcí-Serano, A.: Multiagent Architectures for Intelligent Traffic Management Systems. Transportation Research C, Kluwer. 2002.
10. Jennings, N. R.; Wooldridge, M. J.: Applications of Intelligent Agents. In: Agent Technology: Foundations, Applications, and Markets (eds. N. R. Jennings and M. Wooldridge) 3-28. 1998.
11. Kirschfink, H.: Collective Traffic Control in Motorways. Tutorial at the 11th EURO-Mini Conference on AI in Transportation Systems and Science. Helsinki
12. O'Hare, G. M. P.; Jennings, N. R. (editors). Foundations of distributed artificial intelligence. John Wiley & Sons. New York, 1996.
13. Ossowski, S.: Co-ordination in Artificial Agent Societies, Springer-Verlang. 1999.
14. Ossowski, S.; Hernández, J.; Iglesias, C.A.; Fernández, A.: Engineering Agent Systems for Decision Support. In: Proc. Engineering Societies in the Agents World (ESAW-02). Madrid. 2002.
15. Ossowski, S.; Serrano, J.M.: Agent-based Architectures for Advanced Decision Support. In: Proc. Workshop on Intelligent Physical Agents (WAF), URJC. 2001.
16. Ritchie, S.G.; Stack, R.: A Real Time Expert System for Freeway Incident Management in Orange County, California. ASCE, Fifth International Conference on Computing in Civil Engineering, Anaheim, California, USA. 1993.
17. Scemama G.: CLAIRE: A Context-Free AI Based Supervisor for Traffic Control. In Artificial Intelligence Applications to Traffic Engineering, M. Bielli, G. Ambrosino and M. Boero (eds.), VSP. 1994
18. Weiss, G.: Multi-agent systems. MIT Press. 1999.