Agent Information Server: a middleware for traveler information

Mahdi Zargayouna[1,2], Flavien Balbo[1,2], Julien Saunier Trassy[2]

[1] Inrets - Gretia, National Institute of Transportation Research and their Security.
2, avenue du Général Malleret-Joinville,
F-94114 Arcueil cedex.
[2] Lamsade, Paris Dauphine University.
Place du Maréchal de Lattre de Tassigny,
75775 Paris Cedex 16, France.
{zargayou,balbo,saunier}@lamsade.dauphine.fr

Abstract. This paper proposes an Agent Traveler Information Server for a daily trip in an urban area. It is based on the multi-agent paradigm and is using the Environment as Active Support of Interaction (EASI) model. It instantiates the mutual awareness concept. The purpose is to allow services, information sources and human travelers to be represented by a unified agent structure and to allow them to interact homogeneously although they are conceptually different. Given that the whole information process must be envisaged in a real time configuration, the increase of the interactions has to be taken into account and the classical interaction modes become rapidly inefficient. The EASI model enables agents to build their interaction interests egocentrically and delegates the interaction management to the multi-agent environment.

1 Introduction

Traveler Information Systems (TIS) are applications of great interest for the multi-agent paradigm. The multiplication of independent transportation services and of the information resources in the network are suitable to the design of distributed systems. These applications and the human users connected to the system try to achieve independent goals; they correspond to the agent conception in Multi-Agent Systems (MAS). In a TIS, the main purpose of an agent is to get the information concerning its context at the right time. The information flow management is then crucial, especially when the number of agents is very high (services, information resources and users) and very changing (users' agents are present temporarily, just as long as the traveler is not yet at his destination). The management of interactions between these heterogeneous agents is therefore the main task to be fulfilled by the system. The problem is to know which agent can match the other's functional needs. In MAS, the use of specialized agents (called middle-agents) that help others to locate service providers is a solution that is frequently used. The advantage of this approach is to locate an agent by its capabilities, thus allowing a matching with the needs of the agents.

During the interaction step, mobile agents can also be used to reduce the communication cost by limiting the number of remote interactions and the amount of data exchanged over the network. Our proposition combines the advantages of the middle agents and the mobile agents' approaches. It solves the problems dealing with a dynamic informational context and more generally each real time information management. Such problems include the fact that the information management process must take into account the reactivity of agents not to information sources only but to the information content too, and moreover the reactivity of agents may be based on a combination of information provided by different sources.

Our middleware belongs to the field of Distributed Agent Environment [14], and interaction within the platform is based on the EASI - Environment as Active Support of Interaction - model, which enables each agent to perceive information that is available in the environment [2]. The remainder of the paper is structured as follows: section 2 presents the traveler information domain, section 3 presents the EASI model; section 4 shows our Agent Traveler Information Server application; and section 5 draws general conclusions.

2 Application domain

The design of a Traveler Information System (TIS) should meet the growing needs of travelers, helping them to choose a mode of transportation, and facilitating their use of the networks [1]. In fact, travelers' requirements are on the rise, since the volume of information sources are increasing rapidly and with new technologies it should theoretically be possible for everyone to receive the information wherever he is. Thus, a TIS should provide two types of information: first, information before the trip starts providing the global offer over all transportation modes for a given request; second, it should provide information during the user's trip, notifying him about events that could occur on his route. Conscious of the importance of the traveler information in their relationship with their customer, the transport operators propose an answer which is adapted to their needs as it is presented in the next section. However, this answer is heterogeneous and disseminated and MAS are a solution to propose an aggregated answer to this problem as we describe it in the section 2.2

2.1 The operator' approach

Operator's responses include passive information - such as variable message boards - and/or interactive information such as web servers or Personal Assistants. However, the information provided by the operators usually only concerns their own transportation mode(s): each operator has access to his own data and provides information to his own customers. The advantage of this type of management is to facilitate information update and to ensure a good quality service for their customer. In addition, the operator traveler information is generally based on existing information provided and used by their operating system. However, operator information makes the mutual management of different sources difficult, and requires the user to be adaptable, that is why preparing a trip is still a hard task [10]. The multi-agent paradigm offers solutions for automating certain tasks of the travelers and to design advanced services.

2.2 The multi-agent approach

The multi-agent paradigm offers solutions to these kinds of problems and MAS are frequently used to develop systems that are adapted to the transportation domain [12] and particularly to the traveler information domain [4, 5, 9]. In the traveler information context, the first function for the system is to collect the information from heterogeneous and distributed systems. In fact, the user should have the possibility to express his needs without necessarily knowing the information sources that are able to answer it. In order to build a personalised answer, the second function is to integrate the obtained data. The traveler should be able to specify his preferences and to receive answers according to his profile. Finally, the last function is to ensure an information follow-up in order to supervise the good unfolding of the traveler's move. The traveler has to be notified about any event that could occur on his route, which is able to disturb it, and solutions (alternatives) have to be proposed to him.

This adequacy between the traveler information domain and the recognized characteristics of agents is at the origin of the application part of the work of the FIPA (Foundation for Intelligent Physical Agents). One example of this formalization allows a human user to book a trip simply by indicating the detail of his need to his PTA (Personal Travel Assistant) [11]. The organization proposed by FIPA is efficient for obtaining pre-trip information: it proposes a solution to the problem of collecting and customizing the information and an advanced function automating the negotiation phase, all of the process is ensured by a dyadic interaction, following a request-response pattern. But though this is essential for the automation of the activities of a travel agency, it is impractical for information about daily travel. For a traveler in a city or suburban network, the problem is not to identify the information sources (that are known) but to manage his moves dynamically, so he can receive personalized information only when the information concerns him.

The second phase of the traveler's information process, i.e. providing him with real-time personalized information about a given trip, cannot efficiently be achieved in the same way as pre-trip information. In a real-time configuration, the request/response pattern becomes expensive especially in a very dynamic context like daily information in an urban area. To solve this problem, we use the mutual awareness concept to convey the right information to the right user i.e. the one who is concerned by the information.

3 The EASI model

Interaction is in the center of the design of a multi-agent system. It should allow agents to locate each others corresponding to their needs. Also, as much as it is possible, it should not be scale sensitive and should avoid message overloads and communication bottlenecks. In open environments such as internet, where agents don't know each others a priori, middle-agents are used mostly as a service tracker [7]. They are based on the capability-based coordination which is a preference/ability matching, in order to identify the best provider for a given capability search. However, assimilating preference and capability is not sufficient when the problem is not the location of the information but the content itself, and the receiver can gain efficiency by choosing itself its sources and criteria. In addition, with a middle-agent, dynamic information rapidly increases the number of message exchanges in order to maintain a valid representation of the world for the agents [3]. This could lead to the apparition of bottlenecks when the agents' interactional needs are high and varying rapidly. Dugdale [8] has proved that, in a dynamic informational context like regulation, a large part of the interactions derive from the concept of mutual awareness, the ability of the agents to take into account not only the messages but also the information itself, as the unit of treatment to be processed.

As an agent knows better than any other agents what its interactional needs are, it should be able to choose which messages it wants to pay attention to, decided according to its interests and independently of the sender. As a mirror, the sender should be given the same possibility about the receivers and, since broadcast costs a lot, both in terms of network occupation and to every agent of the system obliged to process it even if it's not interested in the message content, mutual awareness should be enabled inside the environment itself. The matching of the relevant receivers should no longer be based on preference and abilities, but on the content of the message itself and the interests of the agents.

The solution proposed here is based on distant clients which communicate with representative agents located on the server-side. This scheme enables computation based communication to occur only on the server side and allows the agents to use local technologies fully, thus making possible to use the mutual awareness interaction model. The Agent Traveler Information Server proposed in section 4 is based on these features.

3.1 The interaction model

Mutual awareness is based on the sharing of interactions. To be efficient, this principle implies that agents share a common communication media. As a consequence, an agent has to find among all messages only those that it is interested in. In the reactive agent community, the environment is already used as a common media of interaction. In the cognitive agent community, we have proposed the EASI model [2]. It enables cognitive agents to use the environment to exchange messages and, more precisely, it enables an agent to send messages to an other agent that is located by the environment and it enables agents to perceive every exchanged message. To find useful information from a very large data set, we have grounded our model on symbolic data analysis theory. This theory is aimed at discovering information by modelling both qualitative and quantitative data grouped into what is named symbolic object. In our research, we consider that environment contains symbolic descriptions of messages and agents. The interactional problem is to make possible for agents the use of these descriptions to locate messages according to the environment state.

Let us introduce basic symbolic data analysis definitions that we found in [6]. A symbolic object is a triple $s = (a, R, d)$ where R is a comparison operator between descriptions, d is a description and a is a mapping from Ω (set of individuals, also called entities) in L ($L = \{true, false\}$ or $L = [0, 1]$). An assertion is a special case of a symbolic object and is written as follows: $\forall w \in \Omega\ a(w) = \wedge_{i=1,...,p}[y_i(w)R_i d_i]$ where $y_i(w)$ is the value of the individual w for the symbolic variable y_i. When an assertion is asked of any particular entity $w \in \Omega$, it assumes a value true ($a(w) = 1$) if that assertion holds for that entity, or false ($a(w) = 0$) if not.

In our EASI model, we have added this notation to formalize the knowledge about the description of interaction components (messages and agents). Because it enables to represent the agents, it is possible for agents to create their own interactional context as a set of assertions. In EASI, the set of individuals is the environment (noted E). It contains two types of entities: agents and messages. Symbolic variables will be called visible properties and noted $Pv_j,\ j = 1, ..., p$. Let $M = (Pv_j),\ j = 1, ..., r(r <= p)$ be the definition of the message class and $A = (Pv_j),\ j = r + 1, ..., p$ the definition of the agent class. This last one may be composed of several agent' subclasses, each of them is described by a subset of visible properties and has a null value for the other. Each agent description is updated by the agent itself, modifying dynamically the value of its visible properties. In E, filters co-exist with agents and messages, this will make possible to link a particular message with particular agents. Because an assertion concerns only one entity at the same time, we propose to define a filter as an extension of an assertion. For each agent $e_a \in A$, each message $e_m \in M$ and $(e_j),\ j = 1, .., ll \subset E$ when the filter $f_k(e_a, e_m, (e_j))$ is true, agent e_a is interested by message e_m, and $(e_j)j = 1, .., ll$ verify the imposed relations ((e_j) is optional).

A message put in the environment will be perceived by every agent that has a filter that is matched in the current informational context. The next section describes how we use the dynamicity in the interaction to propose an information server that is controlled by its users.

3.2 The middleware architecture: Agent Information Server

Because, in the EASI interactional model, the environment contains filters (created by agents) and entities (public description of agents and messages), we can propose an information middleware dynamically parameterized by its users. An efficient management of information exchange between several requesters and providers means taking into account the dynamicity of the interests of the requesters for heterogeneous providers and also the cost of the information exchange. Our architecture meets these requirements. The mutual awareness model proposed by EASI makes it possible to put together all the information, each agent perceiving only that information which, according to its filters, concerns its interests. This Agent Information Server (AIS) architecture does not duplicate information from the provider but organizes its use in a defined context. Our server is a common place where requesters and providers exchange information through a common environment.

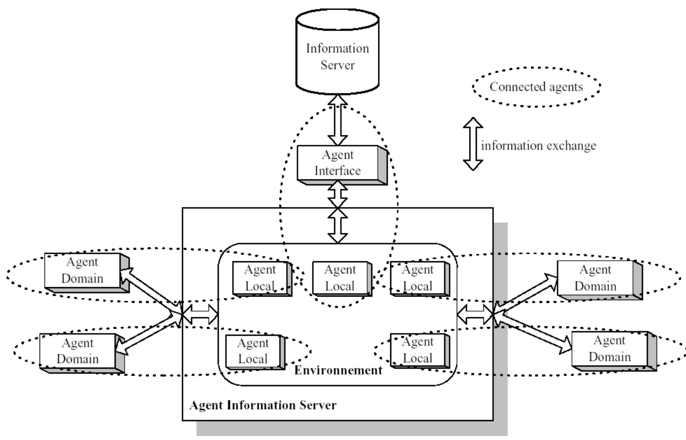

Fig. 1. Agent Information Server

Within our proposition, according to the dynamicity of their information, two types of provider behaviours are possible. For static information, a provider waits for users' requests. In this case, the server is used by requesters in order to identify the provider (with the use of the visible properties) and to interact with it in a normalized way. For dynamic information, a provider puts updated information into the environment. In this case, the server is used by requesters to identify which information has some interest for them from among all data available in the environment. New information is put in the environment once and is received by every interested users.

The multi-agent system upon which our architecture is based is made up of three types of agents (Figure 1). The first, *Interface* agent, is the link between an existing information server and our own. This agent is used by the others to interact with the external server in a normalized way (static information) and/or to gather relevant information for the MAS and to put it in the environment (dynamic information). Using an *Interface* agent within our multi-agent system makes it possible to keep a homogeneous system with heterogeneous components. Agents do not have to know the external server to interact with it; they only need to know which kind of service it provides. This implies that the server may be changed and that the technical means used to interact (http, ftp, SOAP, etc.) is hidden from the users' agents.

The second type of agent is the *Domain* agent. Contrary to *Interface* agents which are not interested in using information coming from the environment (they are only information providers), *Domain* agents may be requesters and/or providers of information. These two agent categories are not located on the server. Using them in our proposal solves the problem of provider identification and standardizes interaction with heterogeneous information providers. Nevertheless the communication cost remains high because each interaction has to be carried out with a message exchange between distant agents.

To solve this problem our multi-agent system has a third category of agent, *Local* agent, which is located on the server. Thus, a part of the processing may be done on the server, reducing the communication cost (as with mobile agents). Each distant agent (*Interface* and *Domain* agent) has a representative (*Local* agent) on the server (Figure 2).

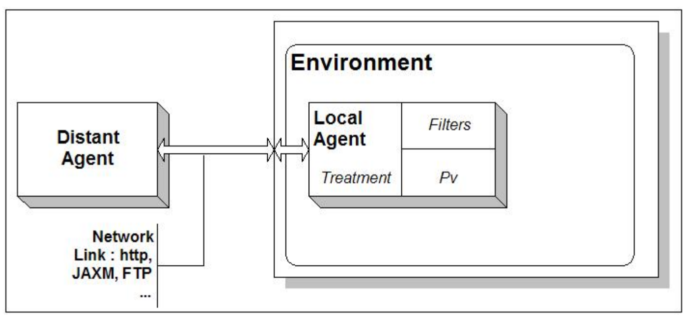

Fig. 2. Relation between distant and local agent

The role of this entity is to manage interaction for the distant agent, by creating/deleting filters according to the needs of the distant agent. For each perceived message, it decides what to do with it. Alternative is to deal with it or to forward it to the distant agent. In that way, the exchanged messages are limited to those that are essential for distant agents. The role of the *Local* agent implies a hybrid architecture, since this agent is the link between the informational environment (it can put and perceive messages inside the middleware) and the application environment (it can send and receive messages to and from distant applications).

Sharing responsibility for interaction processing between local and distant agents means that distant agents may participate in several MAS. For each participation they delegate their interaction needs to their local agent.

4 Agent Information Server

4.1 The software architecture

In order to use the AIS architecture to implement an agent-traveler information server, we have to introduce which agent in our architecture is equivalent to which agent in AIS. The distant information systems can be *Interface* or *Domain* agents. The MPTA (Mobile Personal Travel Assistants) are *Domain* agents, whose role is to "link" the human user to the system. They are the software interlocutors of our system (Figure 3).

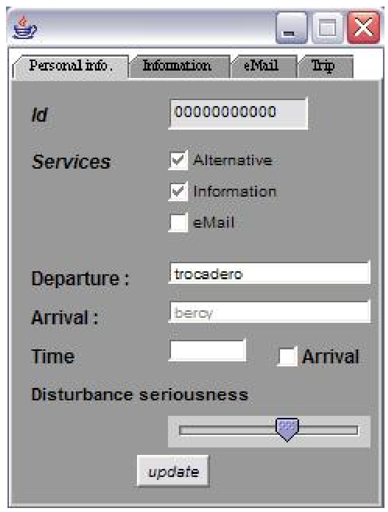

Fig. 3. Preference user interface

In order to test our proposal, we have implemented three information systems. The first is an application that is an *Interface* agent with an existing web service[3] - working following a request/response model - and that creates a trip as an answer to a request. The result of the http request is an xml file containing the sequence of transportation modes to be taken by the user to reach his destination. The second, the traffic system, is also an *Interface* agent connected to an existing service [13] that gives information about the traffic (accidents, deceleration, etc.) and the corresponding seriousness of the disruptions. The messages sent by this service are broadcasted through the environment with no specified receiver. The third and last system - called "alternative" - is specific to our application and gives the nearest alternative station rather than the one which has been received as a request parameter. This last agent is a *Domain* agent which uses information coming from the middleware to find the best alternative for travelers when there are disruptions.

[3] http://patriceb.users.mcs2.netarray.com

For each of these distant agents (*Interface* or *Domain*), *Local* agents representing them are created. *Local* agents are divided into two categories: the first are the agents that we called LA which are permanent, because they represent a distant information system. The second, that we have called PTA (Personal Travel Assistant), are agents which are transitory because they represent an MPTA: they are created the first time a user connects to the server and erased at the end of his session. The first time the user connects, his profile and preferences are uploaded from his MPTA to the corresponding newly created PTA.

The users' preferences are entered via a standard interface where he is asked to determine the categories of services that interest him (warning announcement, alternative) (figure 3). His choice determines the behaviour of his associated PTA. For instance, the seriousness of the disturbance determines in which cases the PTA will warn him. If the value is high, he will only receive the information on disturbances that changes his travel-plans considerably. The relation between MPTA and PTA underlines the information filtering process that a Local agent enables. The first level concerns the filters and visible properties. At this level only useful information (according to the Distant agent preferences) are perceived by the Local agent. To do that, a PTA will use the kind of services the user needs, the trip properties (departure and arrival location) and the seriousness value. This information will be used to create the personalized filters like this one :

$$(f2) \quad f_{PTA}(e_a, e_m) = [Pv_{networkPosition}(e_m) \in Pv_{networkPosition}(e_a)] \wedge [Pv_{seriousness}(e_a) \leq Pv_{seriousness}(e_m)]$$ [4]

The second level concerns the Local agent behaviour. If the human user chooses the alternative service it will forward not the filtered information that it receives about a disruption but an alternative to the initial trip if it exists (see section 4.2).

We focus now on the few technologies used to implement our traveler information server. Some problems had to be solved in order to instantiate the EASI model onto a real implementation. First, a problem is the synchronisation between distant application and the server. Since in the EASI model, messages are caught by filters without the intervention of the owner distant agent, we have to find a way to convey the message to the distant agent asynchronously, what cannot be done upon http which is a synchronous protocol. We chose to send our messages via an api (JAXM[5]) which enables us to send our messages via a provider in an asynchronous way, which releases our server and the distant agent of the synchronisation of http. Second, we want to do the same with the distant users, they shouldn't be obliged to wait for a response of the server i.e. to wait for a response to their messages, since the remainder of the interaction is asynchronous. The solution is to use an intermediate xml page dedicated to

[4] The visible property $Pv_{networkPosition}$ does not really exist. Let us assume that it regroups all data that are useful to locate an information on a network.

[5] http://java.sun.com/xml/downloads/jaxm.html

the user, page that is refreshed periodically (e.g. a few seconds, 10 in our test application). The messages exchanged over the network are SOAP[6] messages.

4.2 The scenario of execution

Figure 4 illustrates our matter. The numbers show the chronologic order of the exchanges. Figure 5 focuses on the internal interactions -the messages exchanged inside the middleware, presented by an AUML diagram. Chronologically, the traveler connects to the server (via his MPTA) and a PTA representing him is created. We suppose that our user is interested in the three services described above. After specifying his departure and destination points, he is asked to wait until his request is processed. His PTA creates a message with this information and deposits it in the environment. In this case, the message is intended to the LAs which have a planning capability - only one in our case (1). When the planning LA receives the message, it forwards it to the *Interface* planning agent. To do so, the request is wrapped into a SOAP message and sent via the Web (2). Then, the distant *Interface* agent requests the planning service for a plan (via http) and the latter sends it back in a message containing an xml trip plan (3); This is transformed - by the LA - into a message obeying the environment syntax, addressed to the user's PTA (4). Note here that the presence of the *Interface* agent between the LA agent and the planning system has the advantage that

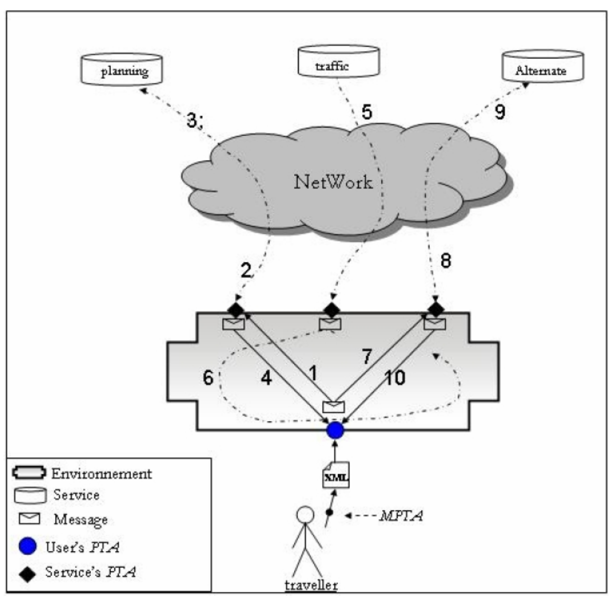

Fig. 4. Example of messages routing

[6] http://www.w3.org/TR/SOAP

the same *Interface* agent can have more than one LA agent representing it in different middleware servers, covering different transportation networks. This way, the user connects in exactly the same way to different networks, and the presence of different services is transparent to him.

When the user's PTA receives the xml plan corresponding to its initial user's request, it forwards it to its MPTA to inform the user, then it parses it and generates a filter for every plan segment (a plan segment is a part of a trip, provided by only one transport mode). This way, the PTA of a user restricts its reception conditions just to the information concerning its own trip. For each of the interaction within the AUML interaction diagram Figure 5, we have noted the visible properties that an agent uses to receive a message. For instance, each agent has an identifier and it is a visible property that can be used to address a message. This diagram is only related to the interaction between agents within the information server and based on the EASI model.

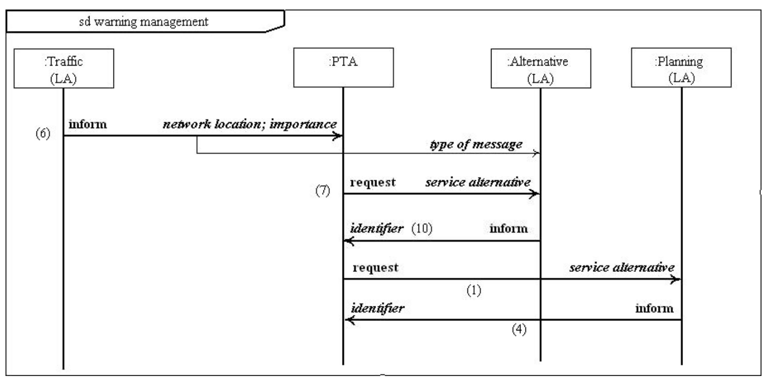

Fig. 5. Warning management protocol

The *Interface* traffic agent collects information - via ftp - on CLAIRE SITI [13] and forwards it to the middleware. If a warning concerns a part of the user's trip (which is the case in Figure 4(5)), the message is intercepted by its PTA (Figure 5 (6)). The type of a message (in this case a warning) is a visible property. Note that this information is also used by alternative LA to intercept this message in order not to send another station concerned by a traffic problem. This interaction is based on the mutual awareness paradigm because this is directed by the receiver and not by the sender of the message. This last one gives several visible properties to its messages and their values are used by receivers to choose which of the messages has interest for them. The network location and disturbance seriousness are visible properties for a warning message. Each PTA according to these values can intercept this information. This interaction directed by the receiver is also based on the mutual awareness paradigm. That implies that agents share the global environment knowledge that the visible properties are. This kind of interaction is represented by the double connection (Figure 5 (6))

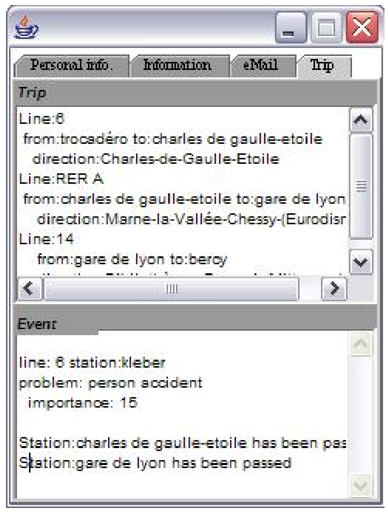

Fig. 6. Trip information interface

When there is no problem, the basic behaviour of the Local agent is to update its filters according to the traveler trip. The figure 4 pictures this comportment because each event relative to a station being passed corresponds to the corresponding rules removing from the environment. For instance when the station Charles-de-Gaulle has been passed the filter concerning the part of the trip relative to the line 6 has been removed.

In our example, we suppose that the disturbance is serious enough to be intercepted by the user's PTA (superior to the value of the disturbance seriousness given by the user). As the alternative service has been chosen by the user, the PTA puts in the environment a message "addressed" to the agent which has the alternative capabilities (7-8-9-10), Once the alternative station is received, the PTA sends an addressed message to the planning LA asking for a plan with the alternative station as a departure point (1-2). These interactions are based on a preference ability matching like with a middle-agent and thus EASI enables the same advantage to dissociate a service from a particular agent. Each LA has a visible property (called service) that is a list where all their abilities are recorded. Because each request contains the identifier of the sender, the answer of a LA to the PTA is based on a dyadic interaction.

When receiving the new plan (3-4), if the additional cost with the alternative trip is less than the current delay, the PTA proposes it to the human user and asks him if he wants to avoid the disrupted station. In this case, and if the new plan is validated by the user, the old filters are replaced by the new ones concerning the new plan; only the events concerning this new plan will henceforth be received.

Thus, with the use of EASI model in our application, it is possible, for the interaction of an agent (representing a user), to be dynamically parameterized by its context, through the update of its filters. Our middleware enables agent to interact in a normalized way. The use of existing classical web services is totally transparent to it; interaction with any kind of service is homogenized by the environment interaction protocol. Moreover, using the same middleware, agents can communicate through dyadic and ability matching if the sender and receiver are interested in the interaction or mutual awareness interaction if only the receiver is interested. Using AIS also enabled us to build a complex service based on different sources that had not been pre-defined to offer such a service.

5 Conclusion and perspectives.

The basic principles behind our Agent Information Server described in this paper are principles generally acknowledged to be of interests in the multi-agent community. The operationalisation of these principles for a dynamic informational context imposes to take into account the update of information and/or of the agents' interest. Our proposition to use mutual awareness to create a communication space where representative of distant agents interact limits the communication cost. Our mutual awareness model (called EASI) is based on a property-based interaction model, which generalizes the capability-based coordination model. In that way, our model enables to take into account the content of information and so, more specific interaction.

A real application of a Transportation Information System illustrates our proposition. This specialized middleware integrates several information servers and enables normalized interaction with them. Because a middleware regroups services relative to a local network, an information server may be used in a specific middleware or in several ones. For instance, the planning *Interface* gives information for several networks whereas a server for a local taxi network will have only one representative. On the same idea, a user making a trip between two towns will have a representative within the two networks.

We have several directions for future works. We plan to investigate the consequences of the admission or the exit of agents on services management that implies that a distant agent constructs its representative in a dynamic way. We also plan to propose a management process for taxonomy of available services. This process will have to take into account that our environment has to remain open with no specialized agent controlling its activity.

Concerning the implementation perspectives, we plan to apply our work to other domains whose problems are suitable to our interests. We think about Agent-Based Marketplaces where the attention of agents according to information varies very rapidly during a day, and could suitably be managed using our approach.

References

1. Adler, J. L., Blue, V. J.: Toward the design of intelligent traveler information systems. Transportation Research Part C 6 (1998) 157–172
2. Balbo, F., Pinson, S.: Toward a Multi-Agent Modelling Approach for Urban Public Transportation Systems. Omicini A., Petta P. et Tolksdorf R. (eds), Engineering Societies in the Agent World II, LNAI **2203**, Springer Verlag (2001) 160–174
3. Balbo F.: A new interaction model for agent based simulation. In European Simulation Multiconference, 2004.
4. Coyle, L., Cunningham, P. and Hayes, C.: A Case-Based Personal Travel Assistant for Elaborating User Requirements and Assessing Offers. Proceedings of the 6th European Conference, ECCBR 2002, Susan Craw, Alun Preece (eds.). LNAI volume **2416**, Springer-Verlag (2002) 505–518
5. Dia, H.: An agent-based approach to modelling driver route choice behaviour under the influence of real-time information. Transportation Research Part C10 (2002) 331-349
6. Diday, E., Hébrail, G.: Symbolic Data Analysis: some in and out. Kesda'98 Eurostat Luxembourg,pp 19-24, April 1998.
7. Decker, K., Sycara, K. and Williamson, M.: Middle-Agent for the Internet. In Fifteenth International Joint Conference on Artificial Intelligence, Morgane Kaufmann (1997) 578-583.
8. Dugdale, J., Pavard, J., Soubie, B.: A Pragmatic Development of a Computer Simulation of an emergency Call Center Designing Cooperative System, Frontiers in Artificial Intelligence and Applications, Rose Dieng et al, IOS Press' (2000)
9. Moraitis, P., Petraki, E. and Spanoudakis, N.: Providing Advanced, Personalised Infomobility Services Using Agent Technology. Proceedings of the 23rd SGAI International Conference on Innovative Techniques and Applications of AI (AI'2003), Cambridge, UK, 15th - 17th 2003.
10. O'Brien, P. D., Nicol, R. C.: FIPA - towards a standard for software agents BT Technol J, 16, 3 (1998) 51-59
11. O'Sullivan, D., Nnez-Surez, J., brochoud, H., Cros, P., Moore, C.and Byrne, C.: Experiences in the use of the FIPA agent technologies for the development of a Personal Travel Application In proceeding of international conference on autonomous agents (Agents), Barcelone, (2000)
12. Schleiffer, R.: Intelligent agents in traffic and transportation. Transportation Research part C: emerging technologies (special issue), Schleiffer R. (Guest) editor, Volume 10C, Numbers 5-6 (2002)
13. Scemama, G., Carles, O.: CLAIRE-SITI, Public and Road Transport Network Management Control: A Unified Approach. IEE Road Transport Information and Control Conference, London (2004)
14. Weyns, D., Parunak, H.V.D., Michel, F., Holvoet, T. Ferber, J.: Environments for Multiagent Systems, State-of-the-art and Research Challenges. Proceedings of Workshop on Environments for Multi-Agent Systems (E4MAS) LNAI **3374** Springer Verlag (2004) 1–47