BULETINUL Universităţii Petrol – Gaze din Ploieşti	Vol. LXI No. 1/2009	51 - 64	Seria Matematică - Informatică - Fizică

Advanced Traveler Information System: An Agent-based Approach for Itineraries Web-Services Composition

M.F. Feki, M.A. Kamoun, S. Hammadi

LAGIS, Ecole Centrale de Lille
Cité scientifique-BP 48, F-59651 Villeneuve d'Ascq Cedex Lille, France
e-mail: {firas.feki,amine.kamoun,slim.hammadi}@ec-lille.fr

Abstract

A wide and growing variety of transport's useful information systems and services are becoming available daily to assist travelers, but technical supports for discovering, selecting and integrating these resources are still limited. This is due to several reasons of different types: organizational, economic, juridical, technical and quality reasons. The purpose of this study was, essentially, to suggest an Agent-Based Cooperative Information System for Multi-modal transport's travelers, trying to make the existing information system cooperating efficiently to provide travelers with multi-modal and multi-operator information itinerary involving different means of transport and several operators. More precisely, this Agent-Based Cooperative Information System has essentially for its aim to calculate the shortest itinerary in terms of distance, time, or cost, to assist travelers using a new approach of web services composition.

Key words: *multi-agent, information systems, multimodal information, public transport, web services composition*

Introduction

Actually, we assist to an emergence of public transports modes and operators, so that travel useful information and mobility services, in public transport, become necessary daily to orientate a traveler toward his move and mobility. Besides, technical supports for discovering, selecting and integrating these resources still till now, limited. The purpose of this study is to suggest an agent-oriented cooperative information system architecture for multi-modal transport's travelers assistance, trying to make the existing operators' information systems cooperate efficiently together, to provide users with the optimized route to follow by compiling the needed information from the different operators information systems. We try in this work to develop a generic flexible integrating framework architecture: a middleware which is able to include any new transport's operator services.

At present researchers and industrialists are getting more and more involved in the development of public transport information systems and services. Some developments are to assist the efficient and effective operation of services by creating different means of information diffusion; others are centered on the delivery of improved information content quality to the customer. The aim is finally to give a precise and an optimized description of the passenger itinerary so that it can satisfy the user criteria and expectations.

The majority of the implemented public transport's information systems are mono-modal, dealing with only one means of transport. The majority of the French transport operator's web sites can be classified in this category of mono-modal systems. Other information systems are multi-modal, concerning several means of transport, but they always concern one transport operator. They provide a multi-modal itinerary description: multimodal information. Since each one belongs to a single transport operator, these last-mentioned information systems still geographically limited [1][8].

To federate the latter projects and to capitalize the transport's research experiences in terms of multi-modal information, the PREDIT [URL 2] - French National Program of Research, Experimentation and Innovation in land Transport - has recently created the PREDIM [URL 1] - the Research and Experimental Platform for the Development of Multi-modal travel Information. Among the several interesting projects realized within the latter framework, we should mention "PASSIM", which is a web travel information directory. Passim includes the several transport operators' web sites links, listed and detailed in each French department [URL 1].

Otherwise, in the others European countries, this travel information integration issue still up to date [URL 3]. A European integration project should be mentioned: TRIDENT [URL 4] is a project trying to use new technologies such as XML to integrate the disparate multi-modal information of the different transport operators. We can also mention the project TRANSASIST [2] which is based on modern information technologies and helps users to travel across urban areas by finding the best route between two points offering solutions for public transport, private transport and pedestrian movement at European standard level. The TRANSASSIST uses central data. In this paper we propose an agent based system able to find the best route between two points where data is distributed on independent sub-systems.

Integration Scenarios

In the industrial context, each Transport Operator's Information System (TOIS) is composed of a local database (DB) describing the different means of transport available (Metro, Bus, Tramway, with stations, timetables…) - in a specific format - and of an Itinerary Calculating Algorithm (ICA) which uses this local database to search optimal itineraries for users requests. To compute a global itinerary, we should integrate the multi-modal information from the different heterogeneous transport operators' information systems. We can proceed to data integration or to applications integration. The data integration consists on building a huge distributed database integrating the local ones; this issue requires a "strong algorithm" which is able to calculate itineraries using a huge database describing the global multi-modal transport network. While the application integration will take advantage of the current existing multi-modal or monomodal information systems, using their algorithms. In other terms, this application integration, that we have adopted in this work, tries to make the TOIS cooperate to support travelers. So we are trying in this work to conceive an agent-oriented integrating framework: a generic cooperative information system for travelers' assistance associating to each TOIS an Information Agent (IA). This system should be able to find the needed sources of information and applications to respond to the various users' requests. It should also be able to interrogate the existing IAs, to make them cooperate and communicate efficiently to compose and compute the needed global itinerary information, using an on-line distributed shortest path optimizing algorithm, and web services integration techniques.

To the question "Why we adopt this strategy?" we can answer that using such an approach permits to:

- Exploit the existing TOIS, not only in terms of data but also in terms of Itinerary Calculating Algorithms;

- o Extend the system to a large geographic transport covered area without constructing a huge database, assuring an open information system;
- o Let the responsibility of updating the transport information to the different operators.

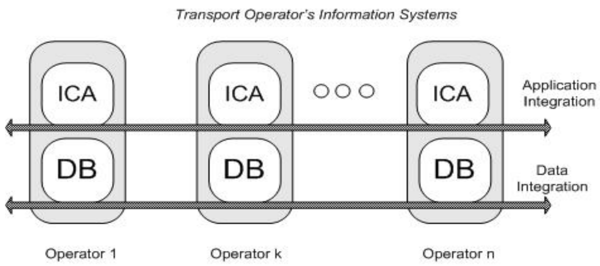

Fig. 1. Integration Scenarios

A Multi-Agent Cooperative Information System (MCIS)

The Suggested Multi-Agent Architecture for the MCIS

The composition of multi-operator routes implies the data availability of each operator which is not always the case. The used approach aims at overcoming this constraint by not integrating the operator data but by integrating its information system: accessing directly to the existing systems. Such a strategy enables us to recover the necessary data starting not from the base of the operator, but from its Transport Operator's Information System (TOIS).

Such integration will also enable us to benefit from the calculators of each operator. So the routes search problem between two stations belonging to different operators is brought back to a search problem for operators set separating the departure and arrival stations.

Indeed, if we could determine the operators' sets implied in displacement as well as the exchange poles, the composition of multi operators' route will become a routes concatenation where each route is calculated by the local processors of the operators' unit.

In addition, the route provided by each local (TOIS) isn't always unique. Thus, while varying these local routes, we obtain different ways with different durations but passing by the same operators' set according to the same order. We will use algorithms to find the shortest way in this set.

According to this description, we will associate to each transport operator's information system (TOIS), an Information Agent (IA). So each agent is responsible for its operator's network (Figure 2). In other terms, each agent is able to suggest an optimized itinerary for the user's request, if this request is dealing with a departure place A, and with an arrival one B which are included in the agent's area: the closest to the agent's network stations. So we make the hypothesis that each considered place can be associated to one station.

The different information's agents register themselves once starting in a specific agent called Directory agent in which they precise details concerning the intersections between their operators.

The composition and the search for routes will be assigned to Mediator agents. These agents will be called: Composer Agent (CA).

The CA is able to compose a total offer of route by communicating with different IA and recovering various local offers using an on-line shortest path algorithm [3][9].

CA does not have a total sight of all MCIS' IAs. Indeed, to answer a given request, CA is not able to determine a field of research from where interest of a Directory Selector Agent (DSA). The latter will be requested by the mediators to delimit the research fields related to the request.

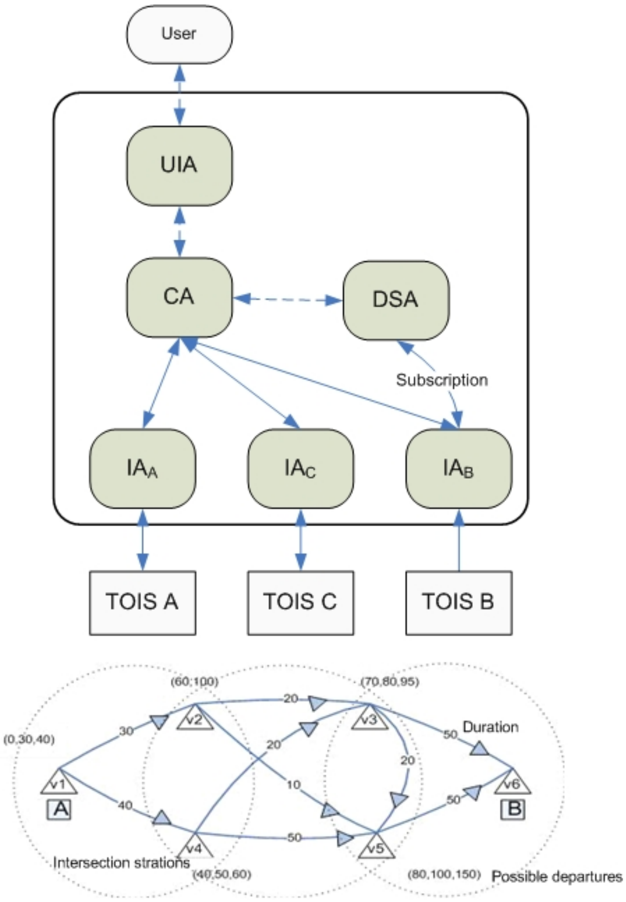

Fig. 2. The multi-agent architecture

Indeed, the Directory Selector Agent suggests several sets of operators by using a specific algorithm [3][4][7]. These sets of operators link the departure station to that of arrival. So we succeed to limit the field of research and thus the computing time. It is a question of a dynamic mediation (dynamic mediation which is concerned) since CA does not know in advance the selected IAs for a given request.

Finally, a User Interface Agent (UIA) will be associated to each user. This will enable him to formulate its request and to receive the results.

In order to distribute the requests' treatment, we associate each UIA to a Mediator CA.

Implementation of the Adopted Agent Architecture

To have a stable architecture, the system must separate the two main inside entities: The entity that manages the servlets and Jsp pages and the second one that manages the agents and the parallel computing.

The user communicates with the system via JSP pages and servlets constituting the interface part. Parallel computing is done by the entity named JADE [URL 5] where the different agents are implemented and communicate with each other. JADE (Java Agent DEvelopment Framework) is a software Framework fully implemented in Java language. It simplifies the implementation of multi-agent systems through a middle-ware that complies with the FIPA specifications [URL 6].

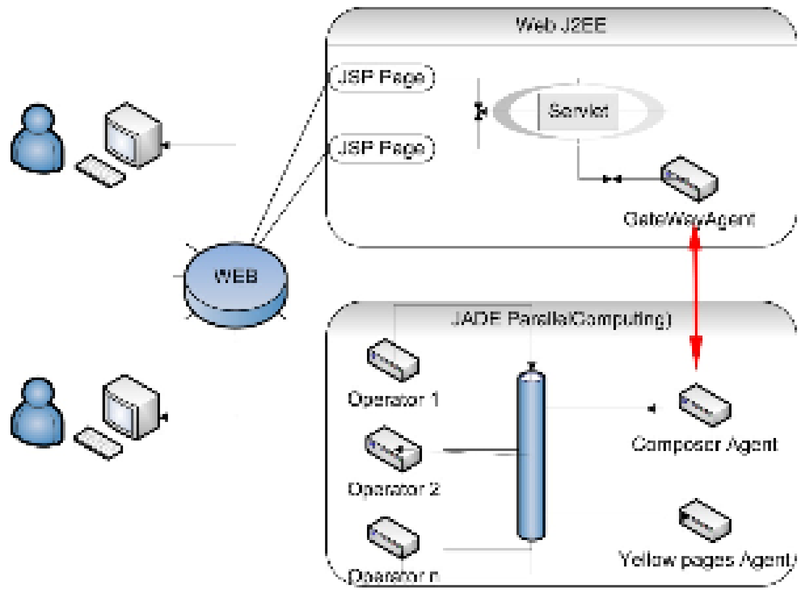

Fig. 3. The main architecture

The Web J2EE Interface

For our part presentation we opted for the J2EE Web architecture formed by the presentation layer, the processing layer and the parallel computing layer. The three layers are independent. This ensures a measure of stability and easy maintenance. Indeed, if one of these layers undergoes modifications, the two others are generally saved.

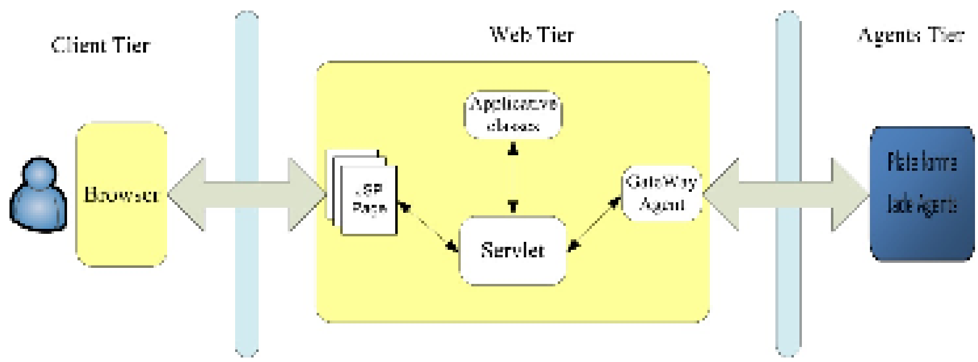

Fig. 4. The J2EE Web architecture

The JSP pages are the Web pages which present the services offered by the system. They are carried outside of the server and they are managed by the servlet. The Servlet Request is the heart of the J2EE interface; it indeed plays the role of an organizer between the various parts of the Web application. The information collected by the JSP pages will be directly forwarded to this servlet which will decide on their treatment.

The classes' treatments contain all the operations to be carried out to post the various pages. Once the page and the corresponding treatment are set up, the servlet orders the class treatment to carry out the necessary instructions; then recovers the results and posts the corresponding JSP page.

The GateWayAgent class connects the Web part of the system to the Jade platform where the parallel computing is carried out. It receives requests from the servlet and communicates them to the JADE's platform Agents.

The users communicate with JSP pages through a Web browser. After selecting the parameters of research in the optimization page, a request is generated and sent to the servlet. The latter receives the request and calls the action "sendMessage". The action "sendMessage" creates a new object called "BlackBoard" which will be the tool of communication between the servlet and the "GateWayAgent". As soon as the "GateWayAgent" receives the "BlackBoard", it extracts the receiver and the contents of the message (the request). It reformulates the request to be comprehensible by the receiver. Then, the servlet sends it and waits for the answer. Once the answer is received, the "GateWayAgent" injects it into the "BlackBoard". Then, sends it to the servlet which also posts it to the browser.

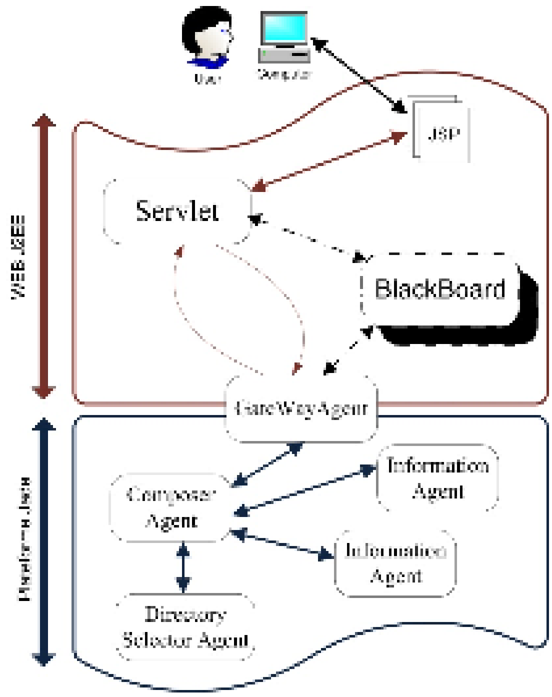

Fig. 5. The Servlet-agent architecture

The Information Agents' Implementation

As we mentioned previously, we associate an Information Agent (IA) with each transport's operator. This agent plays the role of a data extractor from the operator's database.

Indeed, the IA has the faculty to adapt with the operator's communication method with which it is associated.

It has the protocols and the functions which enable it to send requests and receive answers. IA is a mediator between the route search system in one side and the concerned operator's Web service in the other side.

The agent structure remains the same for all operators, but it is only the methods which communicate with the Web service that change in order to adapt to the request formulation and the result exploitation.

In contrast to the other type of agents, the Information Agent (IA) has the faculty to start during the functioning of the platform. Indeed, if the system is running and a new transport operator wishes to register itself, it is not necessary to alter all the implemented code.

In addition, this agent can be shutdown without affecting the system's functioning. This property will facilitate the implementation of traffic regulation solutions [5] and allowing some aspect of transport perturbation management. In fact, if one of the transport operators has some troubles (line cut between two stations for example), we can shutdown the IA related to this operator. This will make the system rejecting the solution using the troubled line.

Once launched, the IA carries out a set of instructions which enable it to adapt to the system organization and to be integrated into it. These instructions are generated by reading textual file which is given as an argument for the IA. The figure 5 shows the contents of this file for a transport operator which is composed by one subway line. The Information Agent offers two principal services which are an answer to these kinds of requests:

- Does station *X* exist in your operator?
- Which is the best way than your operator offers between these two stations?

All the means are placed at the disposal of the information agent to answer these requests.

The first service is a cyclic behavior which takes in entry a station name and answers by a Boolean. It checks if the station exists in the operator or not.

The second service is a behavior which allows the search for simple routes relating to the same operator. Indeed, it seeks an optimal route between two stations starting at a fixed time.

The cycle life of the Information Agent is described as follows:

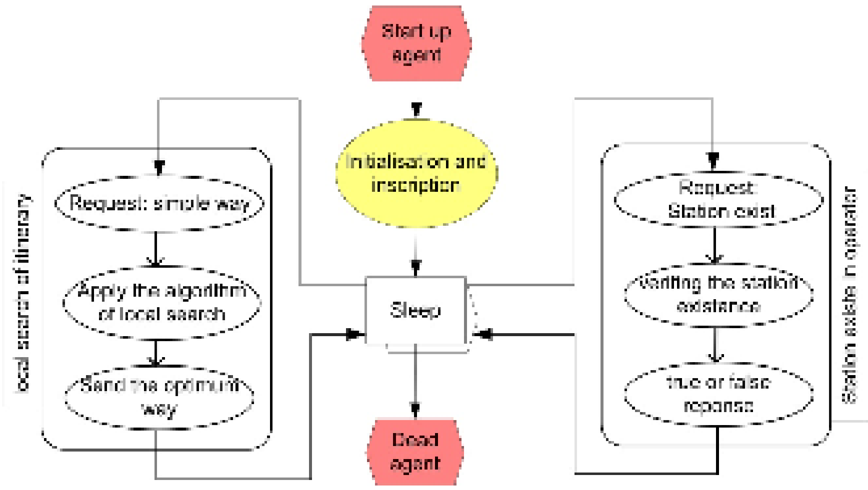

Fig. 6. The Information Agent Life Cycle

The first competence of an Information Agent (IA), also called in the literature adapter (wrapper) or robosoft, is the automated information extraction.

The information extraction from TOIS supposes a minimum of co-operation from the transport operator. Indeed, this last should place the communication protocol at the disposal of the IA's developer. This protocol specifies the syntax of a local request req(*A, B, departureTime*) and its required answer format.

In the absence of this co-operation, it remains possible to reach this local offer information from the operators Web sites.

The robosoft behaves then like a Net surfer: it reaches the Web site, analyzes the HTML Web page and automatically fills the forms with the necessary data its request (starting station, station of arrival and criterion of selection) then recovers the suggested answers.

Software such as "Automat deUnisyn" or "Visual Web task" allows the build of such agents [6]

The technical disadvantages of this method are:

- The developer must create an IA for each interface Web. Thus if the interface changes, IA cannot ensure the retrieval of the data any more;
- A very slow time response: Several Web pages must be analyzed before recovering the necessary solutions;
- The absence of a supplier service agreement does not facilitate the evolution of the system.

In addition, the standardization development relating to the transport's information systems (standards THREE-PRONGED FORK and SIRI)[1], was accompanied in Europe by a Web services standardization which give access remotely to the TOIS of the various operators. So we choose this technology to develop the Information Agents. With this intention, we start by developing the Web services associated with the TOIS sample

We define a transport's Web service by two remote access methods:

A method which can answer a local request req(*A, B, departureTime*) by the arrival time. This arrival time is the soonest time to be in *B* if we go from *A* towards *B* at *departureTime*. It is about the "getHeureArriveeDeProchaineCourseArrive" method.

Figure 6. Details the received answer in XML format by IA_A in response of the request req(*A1, A6*, 08:25) sent to TOIS A.

A method which can answer a local request req(*A, B, departureTime*) by the detail of the local route which begins from *A* towards *B* at *departureTime*.

It is the "getDetailsProchaineCourseArriveEnPremier" method. The Figure 7 details, in the same way, the response received in XML format, by an IA in response of the request req(*A1, A6*, 08:25) sent to TOIS A.

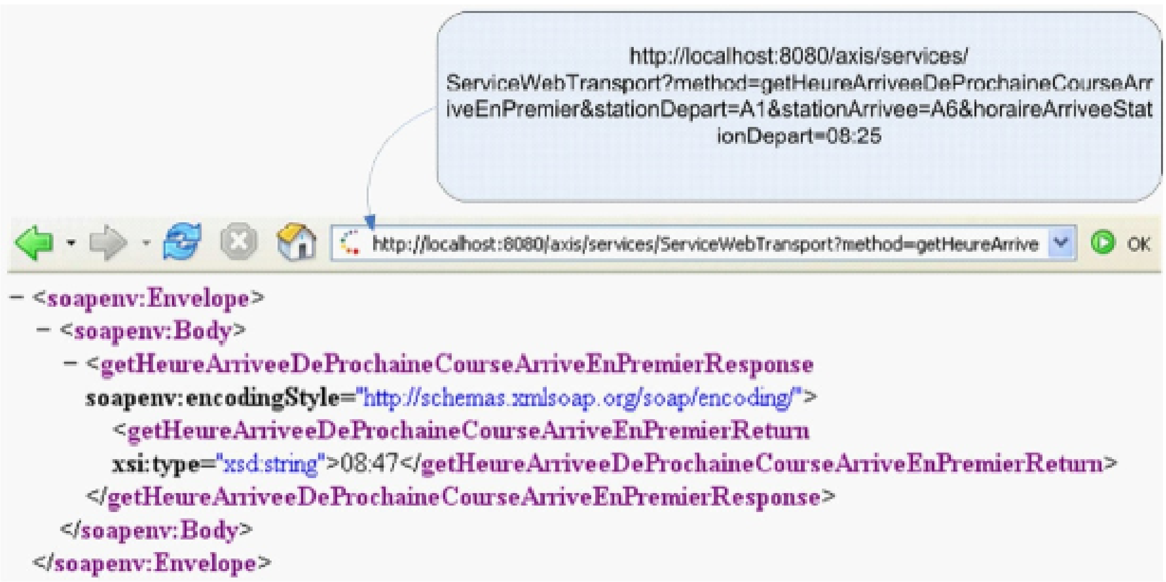

Fig. 7. XML response to the request req(A1,A6, 08:25)

Fig. 8. Detailed XML response to the request req(A1,A6, 08:25)

Deployment in an Industrial Context

The distribution of the various agents has a direct impact on the cost of the communications. The interval time separating the emission and the reception of a message depends primarily on typology of the network. So we should find a distribution which tends to reduce the number of messages exchanged and ensures the parallel execution of the system.

The aim of this approach is to distribute the set of the agents on a network to guarantee a parallel computing. The distribution can be done in a local area network. Thus we reduce the cost of the message transfer on the network. The parallel execution of the different agents allows a considerable profit in performances (see figure).

Moreover, the CA's task is to carry out the route composition algorithm which needs much computing time. So it is interesting to parallelize this treatment in the case of several requests.

We setup a network formed by the different information agents IA and the directory agents DA.

We also create a new Composer Agent CA for each user. Consequently, we use to the maximum these CA agents to treat several parallel requests. In fact, by duplicating the Composer Agents, we can treat several requests at the same time and then we can benefit from the availability of certain agents.

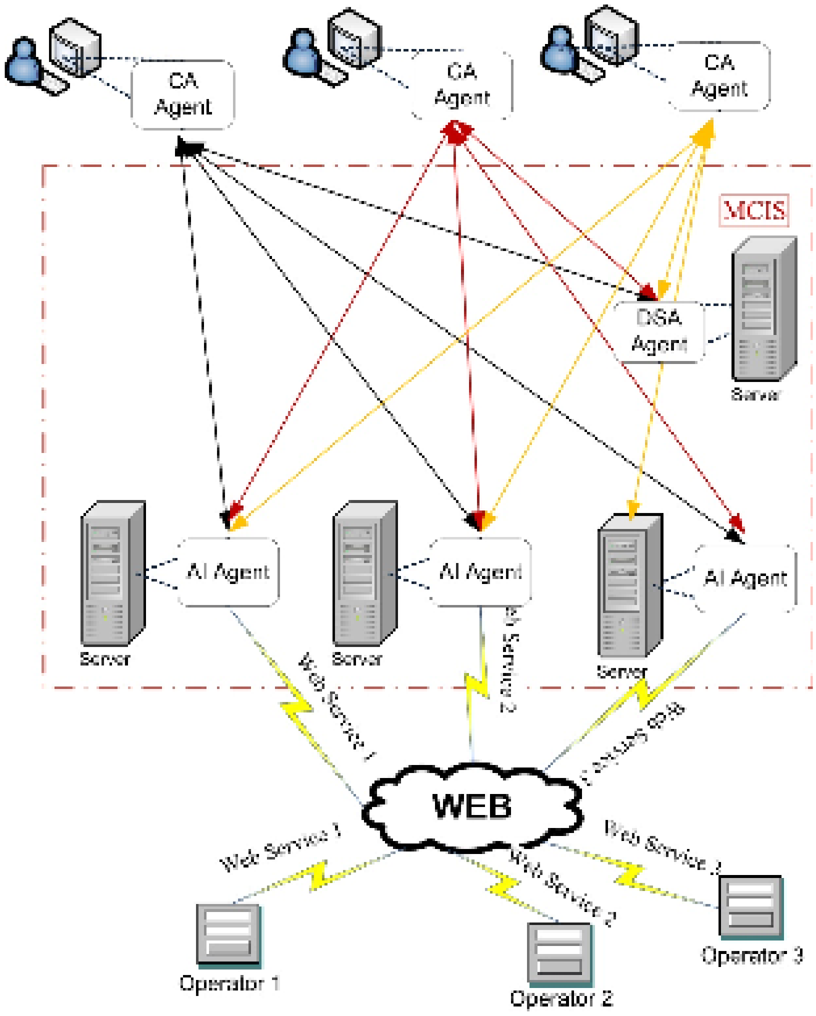

Fig. 9. Parallel deployment in the industrial context

An Execution Example

In this paragraph we will present a complete example of execution, by giving details of the technical and graphical aspects of the system. We will start by presenting the data used in this example then we will explain the different tasks realized by the system.

In the beginning of the MCIS development, the tests and validations were carried out on fictive transport operators and randomly generated schedules. But these latter remain close to reality. After the development of the system Web part, we integrated a tool to generate schedules for the existing transport lines.

In the example we suggest, we have implemented four transport lines of the operator “Transpole” which is the principal operator in the North of France.

The following figure presents the network formed by these four lines and their intersections.

The tests were carried out on the set of these four lines: two subway lines and two bus lines. Each line represents an operator exploiting only one transport line. Thus we can at best present the performances of our system on this reduced data set.

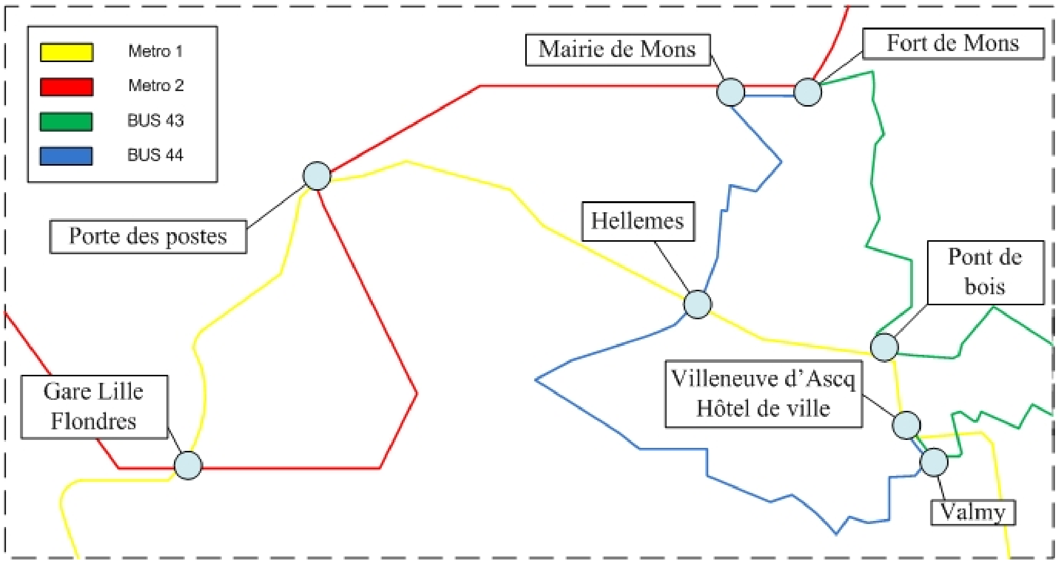

Fig. 10. The Network used to the simulation (Transpole Operator)

The "search page" offers four fields in order to have the request details. The first field is a drop-down menu containing the available stations for departure. In the example given previously we have selected the station "Florence". The second field is another drop-down menu which contains the destination stations. Here we have selected the station "Defaux". The third field is a text-field. It enables the selection of the departure time. The fourth field will be used by the system as a parameter indicating the maximal number of operators used in the route search. This parameter can limit the computing time in some cases.

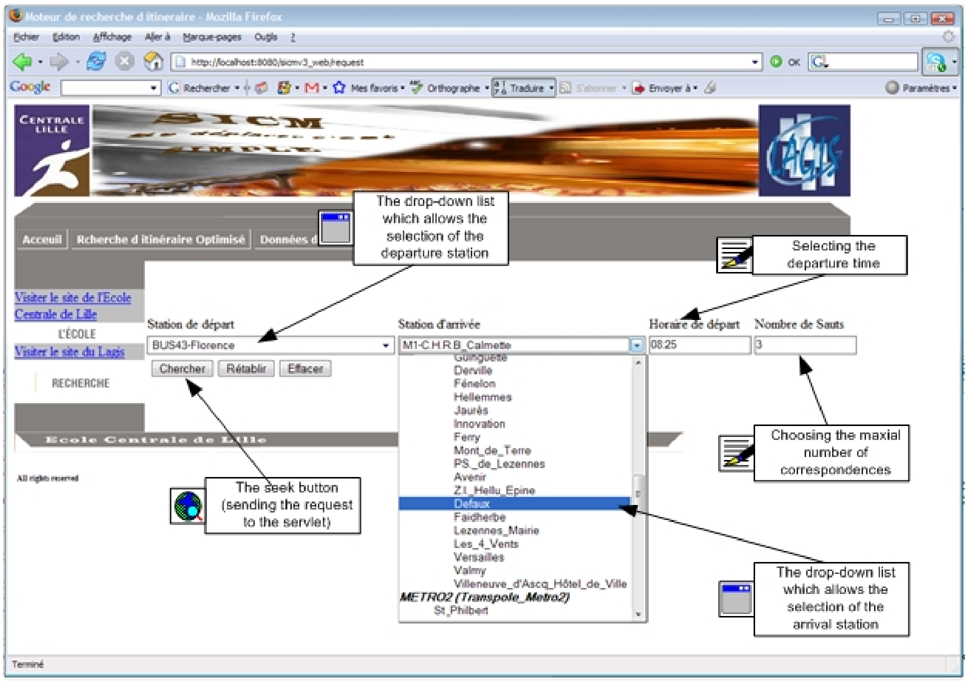

Fig. 11. The itineraries search page

After selecting all the parameters, we run the route search engine by clicking on the button "Chercher". At that time, a request is sent to the servlet which transforms it in order to be comprehensible by the agents. Then, it transmits it to the Composer Agent (CA) of the Jade platform.

As soon as the Composer Agent (CA) finishes its calculation, it sends the result to the servlet. This latter reformulates the result to be exploitable by the web server and transmits it to the JSP page.

We obtain then the page of Figure 12. In the result page, a table is presented for each possible route indicating the various stages of the way. Each stage concerns an operator and it consists of the departure time, the departure station, the arrival time, the arrival station and the used line.

In the suggested example, we can see that the system found two optimal routes.

The first route consists in taking the bus 43 from Florence at 8h29 and going to “Fort Mons”, waiting there until 8h38 then taking the bus 44 to arrive at destination “Défaux” at 8h53.

The second consists in taking the bus 43 at 8h32 and going to “Pont de bois” waiting there until 8h39 then taking the subway 1 to go down to "Villeneuve d' Ascq hotel de ville" and finally taking the bus 44. Thus we arrive to destination at 8h48.

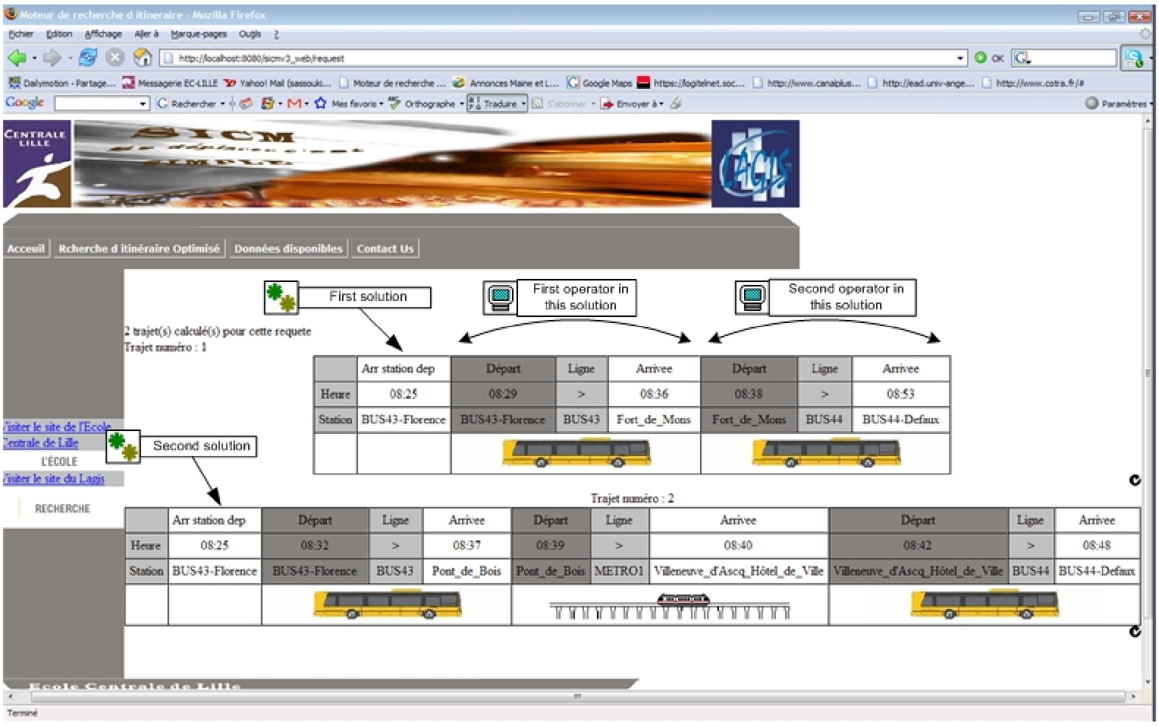

Fig. 12. The result page of the itineraries search

The Figure 13 shows the first 35 messages exchanged between the various agents CA, SA and the four agents of information relative to the four transport lines to compose the results provided by the different operators’ web services.

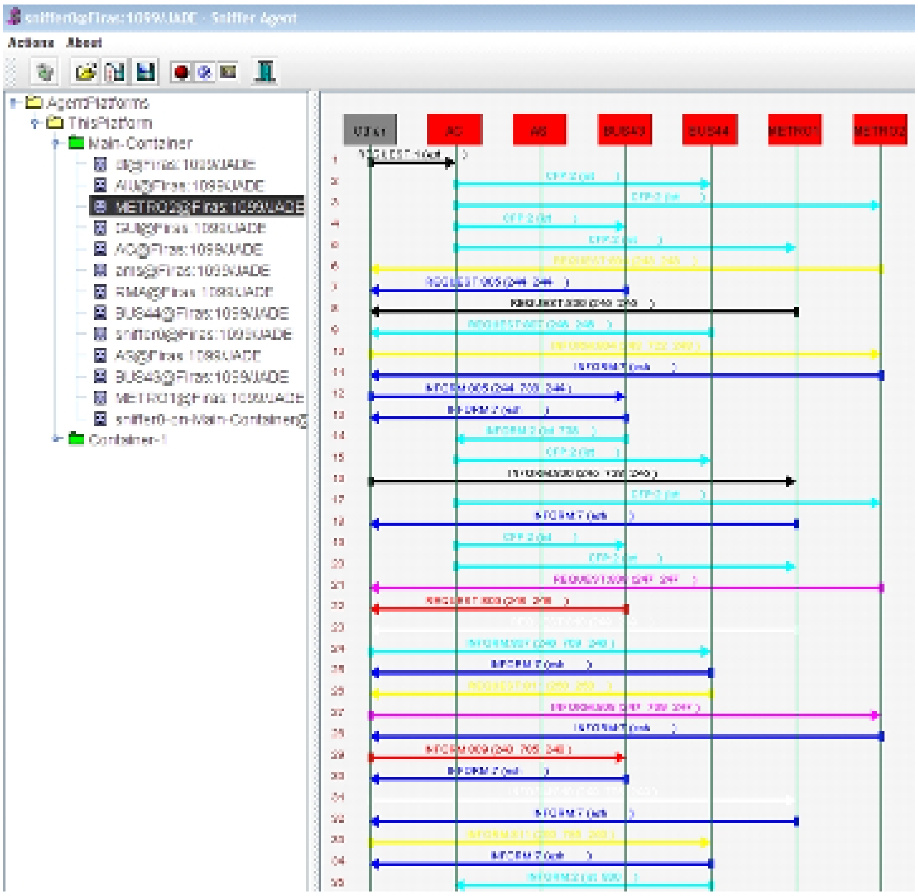

Fig. 13. The agents communications needed to the web services composition

Summary and Conclusions

Transport's multi-modal information systems development is still obstructed by juridical, economic, and by organizational constraints. But even if we suppose that there is a free accessibility to transport operators servers, managing the applications integration between the operators computer technical supports, is still too complicated on a daily basis. The appeal of multi-agent techniques makes the problem more flexible and offers more research perspectives in terms of information systems interoperability.

The latter studied cooperative information system has essentially for its aim to compute a multi-modal and a multi-operator route to assist travelers. It is also very tempting to include other information systems dealing with the traveling objectives, cultural or tourist events, or personal agenda activities. The global aim is to enhance a cocooning e-mobility: to make different transport related services accessible from the most common personal information supports, such as e-mail, PDA, mobile phone via an agent based web services composition.

References

1. Danflous, D. - *Deployment of national multimodal information systems*, Technical report, Centre de documentation du CERTU, CE06 5749, August 2000
2. Hrin, G.R., Anghel, L.E., Tomescu, M., Iliescu, I., Savu, D. - Solutions for Finding the Optimum Route between Two Urban Locations Using Public or Private Transport or Pedestrian Movement, *Studies in Informatics and Control*, Vol. 17, No 4, November 2008
3. Kamoun, M.A., Uster, G., Hammadi, S. - An Agent-Based Cooperative Information System for Multi-modal Travelers Route Computation, *Proc. IEEE International Conference on Systems, Man, and Cybernetics*, October 10-12, The Big Island Hawaii USA, October 2005
4. Kamoun, M.A., Uster, G., Hammadi, S. - An Agent-Based Cooperative Information System for Multimodal Transport's Travelers Assistance, *Proc. IMACS 2005 Scientific Computation: Applied Mathematics and Simulation*, Paris France, July 2005
5. Borne, P., Fayech, B., Hammadi, S., Maouche., S. - Decision Support System for Urban transportation networks, *IEEE/SMC special issue on Decision Technologies in honour of Prof Madan Singh*, Vol. 33, pp. 67-77, February 2003
6. Kamoun, M.A., Uster, G., Hammadi, S. - A Multi-Agent Architecture For a Transport's Multi-Modal Information System, *WSEAS Transactions On Systems*, Issue 5, Vol. 3, pp. 2062 , July 2004
7. Kamoun, M.A. - *Designing an information system for multi-modal travelling assistance: a multi-agent approach for an on-line routes composition*, PHD Thesis, Ecole Centrale of Lille, 2007
8. Petit-Roze, C. - *Organisation multi-agents au service de la personnalisation de l'information,* PHD Thesis, Université de Valenciennes et du Hainaut-Cambrésis, 2003
9. Wang, J., Kaempke, K. - Shortest route computation in distributed systems, *Computers and operation research*, Vol. 31, pp. 1621-1633, 2004

[URL 1] http://www.predim.org
[URL 2] http://www.predit.prd.fr
[URL 3] http://www.transportdirect.info/TransportDirect/en/
[URL 4] http://www.ertico.com/en/activities/activities/trident_website.htm
[URL 5] http://jade.tilab.com/
[URL 6] http://www.fipa.org/

Sistem informatic avansat pentru călători: o abordare bazată pe agenți pentru determinarea itinerariilor prin servicii Web

Rezumat

Actualmente sunt disponibile o gamă largă de sisteme informatice şi servicii, utile în transportul public, ce asistă călătorii, însă suportul tehnic pentru identificarea, selectarea şi integrarea resurselor implicate este încă limitat. Aceasta din cauza unor motive de ordin organizațional, economic, juridic, tehnic şi cu privire la calitate. Acest studiu propune un Sistem Informatic Cooperativ bazat pe Agenți pentru asistarea călătorilor implicați în transportul multimodal, sistem care încearcă o cooperare eficientă a sistemelor informatice existente, în scopul furnizării de itinerarii în situații care implică mijloace de transport diverse şi operatori diverşi. Mai precis, acest Sistem Informatic Cooperativ bazat pe Agenți are sarcina principală de a determina cel mai scurt itinerariu în termeni de distanță, timp sau cost, pentru a asista călătorii, folosind o abordare nouă a serviciilor Web.