A Generic Method for Personalizing Interactive Systems: application to traveler information

Mourad Abed, Abdouroihamane Anli, Christophe Kolski, Emmanuelle Grislin-Le Strugeon

A Generic Method for Personalizing Interactive Systems: Application to Traveler Information

Mourad ABED, Abdouroihamane ANLI, Christophe KOLSKI and Emmanuelle GRISLIN

3.1. Introduction

Following the wide development of distributed and networked information sources, more specifically on the Internet, the advantage of benefiting from reference points, such as "portal" sites that offer facilitated, even personalized access to all available resources became apparent. The personalization of information systems (ISs) aims to provide an adaptive and intelligent human-machine interaction with the goal of improving the efficiency of the interaction as well as the usability (in the sense of [NIE 93]) of systems [MOU 09]. It must offer the possibility of accessing ever greater quantities of information on increasingly varied media and support increasingly different interaction modes.

In order to meet the usability criteria and enable the user to easily find the information he wants, personalization appears to be an appropriate solution [MOU 09], [PET 03a]. In addition to the personalization of delivered information, other aspects of the interaction can be subject to personalization in this context, such as the consideration of different interaction modes (vocal, textual, etc.), the consideration of different interaction platforms (PC, personalized digital assistant or PDA, cell phone, etc.) and user assistance. Our objective is to provide a support to the design of personalization systems that covers these different aspects. The consideration of interaction modes and platforms used means, for us, that the personalization system must integrate them into its reasoning in relation to the preferences of the user. The personalization system must be able to know or deduce, for example, that for a given piece of information the user would prefer to listen to it from his cell phone, rather than read it on his PDA.

This notion of personalization is inscribed in the continuity of current research, which has been very active in man-machine interactions since the beginning of the 1980s, and targets new, increasingly adaptive and intelligent interactions [HOO 00], [KOL 98], [MOU 09], [KOL 92] (see also the *User Modeling and User-Adapted Interaction* journal). Initially, through its approach centered around adaptation to the user, personalization in human-computer interaction (HCI) can also be seen as very complementary to current research on plasticity, which currently most often puts emphasis on adaptation to the platform and the interaction environment [CAL 03], [CAL 07], [KOL 04,] (see also Chapter 11 in this book).

This chapter is based on different documents, including [ANL 05a], [ANL 05b], [ANL 05c], [ANL 06b], [ANL 07], [GRI 07,], and especially [ANL 06a] (a PhD thesis co-financed by Archimed in Lille and the Nord-Pas-de-Calais region of France). It begins with a non-exhaustive review of personalization systems at the origin of the proposal, which is the focus of this chapter. Then, an approach called *PERsonalization METhodology* (PerMet) for the development of personalized ISs, as well as a personalization system called *PERsonalization SYSTem* (PerSyst), which differs from the others in its distributed and evolving aspect, will be presented. Giving the main points, we then describe an application developed for the validation of our approach in the domain of transport; it is meant for itinerary planning based on the personal calendar of the user. A discussion and a conclusion will end this chapter.

3.2. Personalization in HCI: examples of existing approaches, at the origin of the approach proposed

Several systems are contributing to progress in the domain of personalization in human-machine interactions. These systems are, for the most part, destined to aid navigation on the Web (for example, Web browsers). They ensure the observation of user behavior, and the research, filtering and presentation of this information.

Arising from the works of Lieberman [LIE 01], Letizia saves the URLs chosen by the user, reads the pages and draws up a profile of the user as he visits pages; based on this, it searches for other pages that are likely to interest the user and presents its results in an independent window.

IFWeb [ASN 97] carries out the search and filtering of documents by taking into account the specific needs of the user; when a document is selected by a user, the system searches the web for similar documents and shows them to the user, ranking them in order of relevance.

IFM (*Intelligent File Manipulator*) [VIR 02] is a system that aids the graphic handling of file systems; it intervenes automatically and suggests advice when a user makes inconsistent approaches or commits handling errors. IFM bases itself on stereotypes and includes a mechanism for recognizing the goals of the user.

InfoSleuth [NOD 00] relates the queries of the user to the corresponding service providers. The work of InfoSleuth agents is based on the use of ontologies, which enable them to specify the queries or to break them down, and then to fuse collected information together based on heterogeneous and distributed sources.

WebMate [KEE 00] is an agent for aiding information search on the web. It learns the profile of the user, prepares personalized news for him and helps him improve his information search. The profile of a user is an ensemble of keywords found in the pages that he has selected. These pages are positive examples of the machine learning algorithm; this algorithm uses the frequency of appearances of words in the pages. There is a mechanism of keyword "expansion": the addition of words with a similar connotation to a given keyword in order to specify its meaning.

Stemming from previous works at the Laboratoire d'Automatique, de Mécanique et d'Informatique industrielles et Humaines (LAMIH), MAPIS (MultiAgent Personalized Information System) [PET 03a], [PET 04], [PET 06] is a Web application based on software agents that help users of public people transport with their itinerary choice. The profile of the user corresponds to the associated weighting given in relation to the different modes (bus, train, metro, walking, etc.), the length of the journey, the number of changes and the cost of the journey. MAPIS uses a reinforcement learning mechanism for management of the user profile.

Other systems are based on the profile of the user to provide personalized information in specific domains, such as *Gulliver's Genie* [OHA 03] for tourism and Smart Radio [HAY 04] for music. The former uses localization, direction and user preferences to search for cultural or tourist places that could be of interest to him, and sends this personalized information on a PDA. The system is based on the *Beliefs, Desires and Intentions* or BDI model [RAO 91] to infer the mental state of the user. The latter enables users to listen to their favorite music or to a musical program recommended by the system. In order to know their musical tastes, Smart Radio asks its users to rate an ensemble of musical pieces, or musical programs, from one to five (five being the highest mark), in order to establish user profiles.

The user therefore has the possibility of creating a personal *playlist* and can later listen to his favorite tracks. Thanks to this profile, the system can find one or more users with common tastes in terms of music and thus recommend to each person another person's or other people's playlist(s).

Another category of systems aims to facilitate the design of personalized ISs. They generally appear in the form of an application ensuring the representation of the user profile and the inference mechanisms for choosing which solution to suggest. Independent of the browser used, BroadWay [TRO 99] is a system that bases itself on user navigation to recommend links to a particular user, using a case-based reasoning motor for this.

The *Belief, Goal and Plan Maintenance System* or BGP-MS [KOB 95], [PHO 99] is a user modeling system that enables the goals, beliefs and knowledge of the user to be taken into account. It functions according to different types of inferences, from hypotheses based on an initial questionnaire, observed actions and the knowledge of an ensemble of predefined subgroups. This system can be used in a server with multiple users and applications.

The Eperson project [DIC 03] aims to provide a common open platform enabling software agents to be able to assist the user, all the while preserving the confidentiality of information about him. The system appears as a server providing Web services for the management of user profiles. The user data is organized as an ontology [CHA 99].

NetP 7 (www.tornago.com) is a system that enables prediction of the hobbies and interests of users for electronic commerce. The prediction is made based on inferences of data provided explicitly by the user and the data implicitly collected by user queries and commands (purchase of products). This system can be used for web applications, telephone call centers, e-mail, publicity catalogs, etc.

PassPort.Net [OPP 04] is a user profile server. The user joins the service, giving personal data. These data are used by external applications (integrating their own methods of personalization) to provide the user with personalized services corresponding to his profile.

In the WebSphere suite (www.ibm.com), a software application allows the detection of user trends and preferences. This application manages the content and the structure of the commercial site by adapting them in relation to the client. It has been used for the personalization of information on sites selling hardware and software, for example.

Table 3.1 presents a summary of the different personalization systems that have served as a basis for the discussion that led to our proposal.

		Systems	Multi-application	Distribution	Upgradeability	Information search	Information collection	Type of filtering	Type of personalization
Free or academic systems	Agent based	Letizia	-	-	-	√	implicit	[C]: Terme frequency-Inverse Document Frequency (TF-IDF)	Recommendation of a web page
		IFWeb	-	-	-	√	implicit	semantic networks [P]	Filtering and sorting of information
		ConCall	-	-	-	√	implicit/ explicit	[P][RC]: stereotypes and TF-IDF	Filtering, sorting and recall of information
		InfoSleuth	-	√	√	√	implicit/ explicit	[P]: ontology	Reformulation of the request
		Gulliver's Genie	-	√	-	√	implicit/ explicit	[P][C]: Belief, Desire and Intention (BDI)	Recommendation of touristic places, guiding
		WebMate	-	-	-	√	implicit/ explicit	[P]: TF-IDF	Information filtering
		MAPIS	-	-	-	√	implicit/ explicit	[P]: reinforcement learning	Information filtering
		Eperson		√	√	√	implicit/ explicit	[P][FC]: ontology	Information filtering
	Others	RESCUER	-	-	-	-	implicit	[C]: human plausible reasoning	Help and advice
		IFM	-	-	-	-	implicit	[C]: stereotype, human plausible reasoning	Help and advice
		Smart radio	-	-	-	√	implicit/ explicit	[FC]: Pearson	Music recommendation
		BGP-MS	√	-	-	-	implicit/ explicit	[P] [RC]: modal logic	Adaptation of the interface
		BroadWay	√	-	-	-	implicit/ explicit	[FC]: case based reasoning	Link recommendation
Commercial systems		NetP 7	√	?	?	?	implicit/ explicit	Different data mining techniques	Recommendation of products, personalization of content
		PassPort.Net	√	-	-	-	Left to third party applications	Left to third party applications	Left to third party applications
		WebSphere	√	?	?	?	implicit/ explicit	Different data mining techniques	Recommendation of products, personalization of content

Table 3.1. *Comparative study of personalization systems at the origin of the proposal [ANL 06a] ([FC]: collaborative filtering; [P]: profile-based filtering; [C]: contextual filtering; and [RC]: community recommendation)*

Two approaches can be distinguished for the construction of personalization systems.

The first approach consists in providing an interactive system that itself (*ad hoc*) incorporates personalization, such as IFM, RESCUER and SmartRadio. These systems generally appear as software agents (see Letizia, IFWeb, Concall, InfoSleuth, Gulliver's Genie, WebMate and MAPIS), which ensure search, filtering and information presentation functions. These systems have the advantage of being directly in contact with the user, which facilitates the collection of data regarding user behavior for a more subtle personalization. Their main disadvantage is in the fact that reuse of the systems is very limited.

The second approach consists of providing a system that is dedicated to personalization and interacts with a third party for personalization (see e-Person, BGP-MS, Broadway, NetP 7, Passport.Net and WebSphere). The main functions of these systems include managing the user profile and the selection of relevant data that the third party applications are tasked with presenting to the user. These systems are much more flexible than the previous ones. They are generally used to personalize several applications likely to be used by the same users. Of course, a communication protocol is necessary to enable communication between the personalization system and the third party applications. Besides the advantages that final users will benefit from (the same profile for different applications, use of user experiences from one application to another by the system, etc.); the development costs of personalized applications are dramatically reduced.

The existing systems are generally meant for a particular kind of personalization and incorporate well-defined methods of personalization. It is very rare to find a system that ensures personalization of the container as well as the content, for example. This would require the integration of different (often cumbersome) methods of collection and management of the user profile in a single system. The ideal would be to have a personalization that is generic enough (which can support the different types of personalization) and favors an incremental integration of the different methods of personalization. As the existing systems do not do this or only to a small extent, we propose an approach for the design of a personalized information system (PIS) to satisfy this need[1].

1 The approach put forward, called PerMet, is based on an ensemble of foundations coming from state-of-the-art, available in [ANL 06a], several development models (enriched or not from the point of view of human-machine interactions [KOL 01]) and methods of analysis and design coming from the literature, representative of several classes of methods. Of these classes let us in particular note: MERISE (method for the computing study and realization of business systems) [NAN 01], which is representative of the systemic methods of analysis and design of information systems; *Two Track Unified Process* [ROQ 07], which is representative of object-oriented analysis and design methods, such as UP [JAC 99]; WAE (*Web Application Extension*) [CON 00], which is representative of web applications analysis and design methods; and AODPU (*Agent-Oriented Design Process with UML*) [CHE 00], which is representative of methods for the analysis and design of software systems based on software agents.

3.3. PerMet: method for the development of personalized information systems

The approach proposed by PerMet is in line with approaches that separate the IS from the PS, as [KOB 95] and [TRO 99] did initially. This choice of separation between IS and PS is driven by the personalization objective of the former, which possibly already exists. In transport, this is important if we wish to use information coming from different operators of modes of travel.

This separation is also necessary in order to meet the multi-application criteria that enables the unique *Single Sign On* (SSO, authentication or identification allowing a user to access several IT applications or secured websites) [PFI 04]. It must also favor personalization of the multi-modal, multi-channel and multi-platform HMI (according to the initial objective of these works [ANL 05a], [ANL 05b], [ANL 05b]).

The IS is considered an ensemble of services. A personalized service corresponds to a functional unit enabling a personalized HMI. A PIS can therefore be seen as an IS providing at least one personalized service.

To meet the objective of distributiveness and upgradeability, like [DIC 03] and in the continuity of works initially carried out [MAN 02], [PET 01], [PET 03a], [PET 06] at the LAMIH, we advocate the use of a PS based on software agents. The distributiveness is ensured thanks to the characteristics of autonomy, communicability and mobility of software agents, which can be seen as being at the service of users [GRI 01], [KOL 98]. Upgradeability is favored thanks to the characteristics of adaptability and reproducibility.

The PerMet method adheres to a development model following three parts (Figure 3.1). The IS part concerns the development of an IS service. The PS part concerns the adaptation and the configuration of a PS composed of software agents to meet the objectives of service personalization. These two parts, IS and PS, follow a classical development model that can take place in parallel and join together to form the middle part. As we can see in the figure, it starts from a requirements analysis (see Chapter 2 on this subject).

The PerMet development model is iterative and incremental. Each iteration gives rise to an increment aiming to improve the usability of the service. However, it is not necessary to specify all the services to be personalized in an IS. The other services can be specified and developed according to needs. There are different platforms made up of agents (Jade or *Java Agent DEvelopment Framework* [BEL 07] currently being the most well known). Thus, the effective realization of different phases can vary slightly according to the platform used.

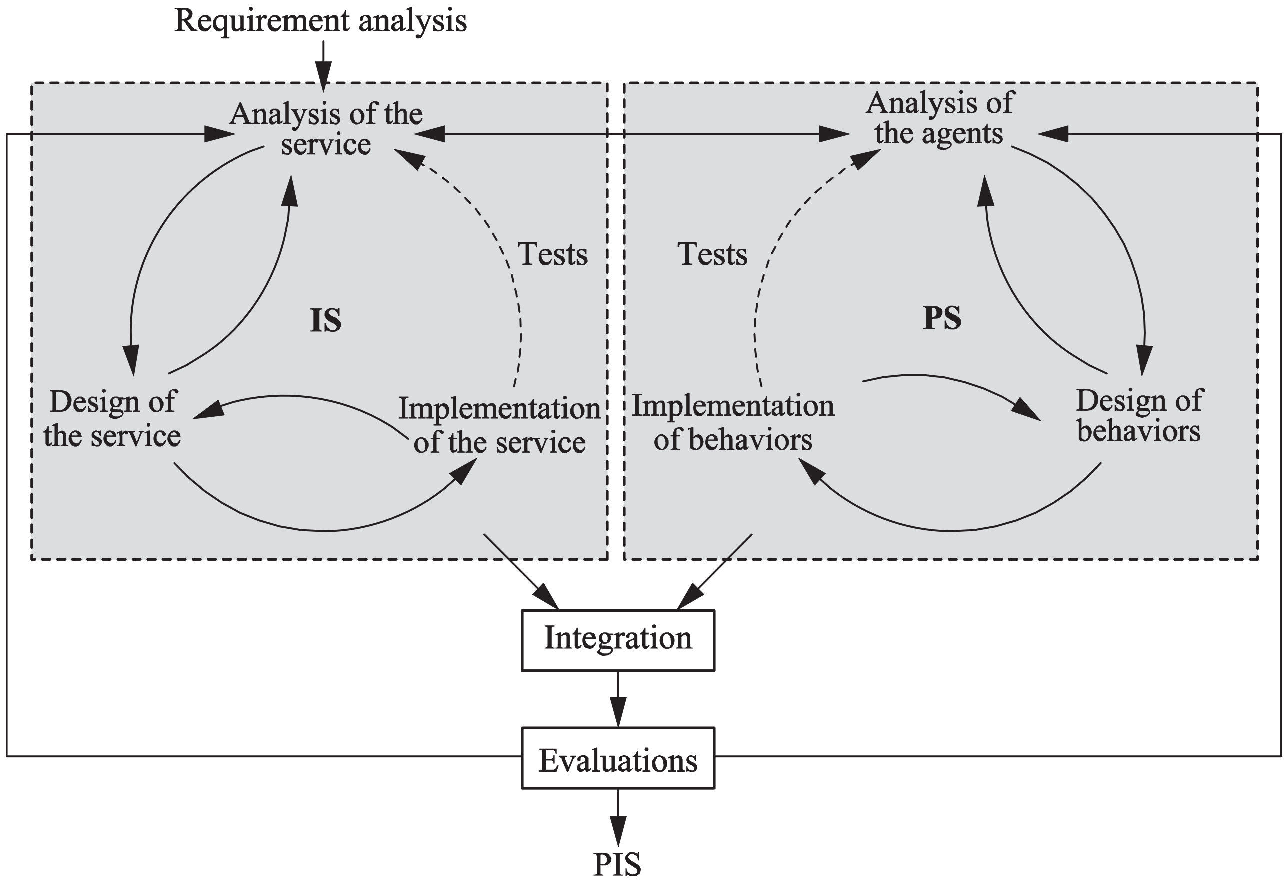

Figure 3.1. *The development model of the PerMet method*

3.3.1. *Analysis of the service*

This phase implies that a preliminary study has been carried out and that the personalized service to be implemented has been identified. It follows the analysis model of the 2TUP method [ROQ 07] to meet the evolution constraints of the service. The analysis phase therefore comprises four stages:

– The *capture of functional needs* describes the service to be developed. It consists of an exhaustive description of the functional and operational needs for the modeling of behaviors expected of the service. Certain use cases can be specified by scenarios, in particular through sequence and/or activity unified modeling language (UML) diagrams.

– *Functional analysis* describes the structure of the service based on functional needs. It consists of structuring and representing objects of the business domain of the service through a series of class diagrams. The behavior of the service is then described using dynamic diagrams (sequence, activity and transition-state diagrams). It is also in this stage that data models (modeled by class diagrams) exchanged between the service and the PS are defined.

– The *capture of technical needs* makes an inventory of all the constraints that do not deal with the description of the business domain or the service. It consists of the specification of tools (software), structure of the hardware to use and constraints for integration with what already exists. The material configuration is modeled through diagrams of deployment and the software specification is modeled via component diagrams.

– *Generic design* defines the elements that are necessary for construction of the technical architecture independently of the functional aspects and functional analysis. The architecture must be constructed so as to favor its reuse. This generic design can lead to the development of one or several prototypes in order to verify and validate the principles defined. The models used in the generic design essentially concern diagrams of class and components.

3.3.2. *Design of the service*

The design aims to specify the analysis model in such a way that it can be implemented with the elements of the architecture. It is a matter of expressing static and dynamic models that will be directly translated in the form of codes, which can be executed in the analysis phase of the service. The class diagram of the analysis phase of the service (functional analysis and generic design) is developed to enable a direct passage from the design model to implementation. The types of attributes of classes and types and parameters of inputs/outputs of the methods need to be specified. Dynamic diagrams are used to specify the states of classes (state-transition diagrams), the algorithm of methods (activity diagrams) and interactions between the different classes (by sequence diagram, for example). The component diagrams that come from the analysis phase will be completed and specified after the design of all the classes useful to the development of the service.

3.3.3. *Implementation of the service*

This is the stage of effective realization of the service. The service is developed in accordance with the conceptual models defined during the design phase. During implementation, the developer will make sure that the design model exactly reflects the IT code produced. Service tests can be carried out by data simulation (in accordance with data models defined in the analysis phase) to be provided to the service (these data will come from the PS when the integration phase is over). These tests can lead to a revision of the analysis model of the service, which will require another realization of different phases of the IS and PS parts.

3.3.4. *Analysis of agents*

This can start at the end of functional analysis of the service analysis phase. The models of agents, behaviors, communication links and deployments of agents are identified here for personalization needs. They are structured into four subphases:

– The *analysis of agent models* identifies the types of agents necessary for the realization of functional needs. The agent models can be identified following the rules below:

- rule 1: an agent model by the functionality expected from the service. For example, the functionality "itinerary search" will be translated by an agent model of "itinerary search",

- rule 2: an agent model by type of interaction platform. For example, an agent model for a PDA, another model for a PC, etc.,

- rule 3: an agent model by external resource with which the system must interact. For example, an agent model for itinerary search on transport operator servers, another for the search of restaurants on web services, etc. In this stage of analysis of agent models, the diagrams for use and deployment cases of the analysis phase are taken up again from the angle of personalization. Only the aspects that are judged as being part of personalization are considered;

– The *identification of communication links* describes relations between the different agents and their beliefs regarding the abilities of various individuals. Here, relationships of the interactions of each agent are modeled. The modeling of communication links can be done via UML extensions put forward by [ODE 99] which are used in *Agent-Oriented Design Process with UML*.

– *Behavior analysis*, for each agent to adequately attain its objective. This analysis determines the granularity necessary for breaking down behaviors. This will therefore depend on the intuition and know-how of the developer. The main rule to apply consists of breaking down a behavior for as long as the task associated with this behavior can be associated with another behavior.

– *Deployment information* describes the different physical locations where the agents go to be executed. This description can concern information of static location (the agent is located at a same place and never changes place) as well as information of dynamic location (the agent can dynamically change place according to the tasks it wishes to accomplish). The deployment information is modeled via deployment diagrams.

3.3.5. *Design of agent behaviors*

This phase enables the models that have come out of the agent analysis phase to be refined to become models that can be directly translated into code. Class diagrams directly modeling the behaviors of agents are specified to enable direct implementation of agent behaviors. The services provided by the behaviors are modeled using activity diagrams. State proactive actions can be represented by state-transition diagrams.

3.3.6. *Implementation of agent behaviors*

This consists of the effective realization of agent behaviors. For each behavior, unitary tests are carried out. The creation of agents, their deployment and the integration of their behavior is possible thanks to tools for agent administration. After having deployed the agents, tests by simulation of PS must be carried out, to check whether the multi-agent system (MAS), which constitutes the PS, meets the objectives defined in the agent analysis phase. These tests can lead to a revision of the analysis model of agents, which will require another realization of the different phases of the PS part.

3.3.7. *Integration*

This is the phase of integration of IS with PS to form the PIS. It consists of making the IS and PS communicate for adaptation of the service developed in the IS part. This phase can also be seen as a classic problem of electronic data interchange [MAN 01]. Several methods are put forward in the literature for communication between heterogeneous applications [ABO 03]. The approach that is most currently used consists of communication by Web service [MON 04].

3.3.8. *Evaluations*

At the end of the integration phase, it will be a matter of testing and evaluating the personalized service obtained. These evaluations can be qualitative (the quality of the personalization realized), quantitative (global performance of the PIS and scalability) or ergonomic (constituting a research domain in its own right, for which numerous methods are possible [HUA 08], [SEA 08]; see also [SOU 10] for works in the domain of PIS evaluation). These evaluations can lead to iteration of the IS part and/or PS part again.

3.4. PerSyst: personalization system supporting the PerMet method

PerMet separates the IS from the PS to take into account the personalization process of different input-output modalities (sound, image, Braille, etc.), different communication channels (Internet, SMS, e-mail, etc.) and different interaction platforms (PC, smartphone, television, etc.). The PerMet method emphasizes the need for an *evolving and distributed* PS that can take into account different types of personalization.

To do this, PerMet recommends the use of a PS based on software agents. In what follows, the general architecture and the design of the PerSyst PS will be described, as well as general models that are useful for the development of a PIS. This is described in full in [ANL 06a].

3.4.1. *General architecture and design of PerSyst*

The two main characteristics that the PS must have are: the possibility of communication with external applications (not necessarily based on software agents); and upgradeability. It is therefore natural that the architecture of PerSyst comprises an agent enabling this communication (communication agent) and an agent enabling the upgradeability of the MAS which makes up the PS (administration agent) that is to be managed. Other agents could then appear in the PS depending on the needs of a project (these agents are established during progression of the PS part of the PerMet method). To make the link between the different agents of the PS, another agent (a coordination agent) has been defined.

Indeed, as the agents are completely autonomous and can be located at various points in the network, it is necessary to have a reference source that will allow the developer to locate the agents and possibly interact with them (evolve their skills, change their location, etc.). This coordination agent also intervenes for the transmission of different messages that agents can exchange between themselves to meet a global objective of the PS. The general architecture of the PerSyst therefore consists of different agents. Figure 3.2 presents the general architecture of PerSyst and its interactions with the existing ISs.

The three agents of communication, coordination and administration (contained in the ellipsis with the grey background in Figure 3.2) form the core of PerSyst. The other agents, which we call applicative agents (A for *Assistant*, P for *Profile*, and S for *Search)*, are examples of agents (they are the most used agent models for the construction of PISs), which could be defined to meet specific objectives according to a particular project.

The behaviors and coordination mechanisms of these agents are therefore established in the phases of agent model analysis and PerMet method of behavior design. The only constraint that these agents must respect is to have a communication link with the coordination agent, in order to be located for upgradeability needs. It is not necessary that it be a direct contact link. There need only be a "contact path" linking an agent with the coordination agent. For example, if agent A1 has a communication relationship with the coordination agent, all that is necessary is for agent A2 to have a communication relationship with A1 in order for PerSet to be able to locate it in the system.

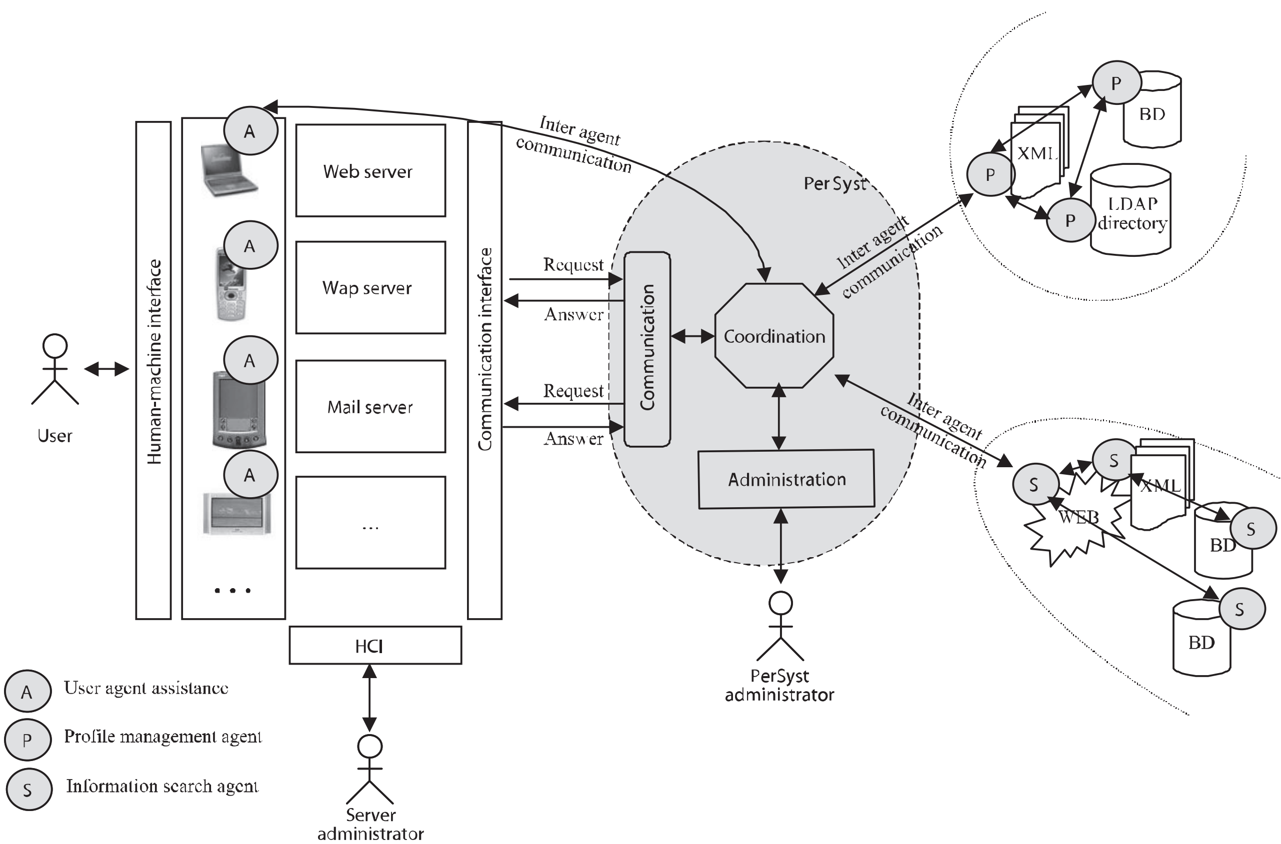

Figure 3.2. *General architecture of PerSyst*

3.4.2. *The coordination agent*

The coordination agent enables the communication link to be made between the different agents that constitute PerSyst. The coordination agent ensures three roles:

– The coordination of the tasks of applicative agents. The coordination agent coordinates the messages exchanged between the various applicative agents. PerSyst recommends only managing the coordination of messages exchanged between agents of different business areas at this level. The coordination between agents in the same business area is delegated to another applicative agent. This avoids overloading the coordination agent and allows for better structuring of the system (for maintenance requirements). The analysis of agent models enables the different business areas of the agents to be distinguished. An agent model is an abstraction of a type of agent carrying out the same activities. Agents that come from the same model can be organized according to a hierarchical structure that does not reflect the physical organization of agents that come out of these models in any way. However, this hierarchical organization informs as to the activity areas to which the agents resulting from these models belong.

– Coordination with external applications. Messages exchanged between external applications and applicative agents transit via the coordination agent. Of course, transformation of the messages is carried out (by the communication agent, described hereafter) in order for the messages sent by external applications to be understood by software agents and vice versa. It is the coordination agent that distributes messages to the applicative agents concerned and sends the responses back to the external applications.

– Coordination with the user. Messages exchanged between the user and applicative agents transit via the coordination stage. The queries of the user are interpreted by the administration agent (described further on in section 3.4.4) and then transmitted to the coordination agent that deals with sending the request to the concerned agent. The transmission of user messages is ensured thanks to the skill the coordination agent has. This skill provides services enabling administration queries to be transmitted to a particular agent.

3.4.3. *The communication agent*

The communication agent enables the translation of queries sent by external applications into messages comprehensible to the coordination agent and vice versa. It is broken down into two parts:

– The first part deals with communication with external applications. It uses the *Simple Object Access Protocol* (SOAP) [NEW 02] for interaction with external applications which can be written in any programming language. It provides four communication primitives (*Ask*, *Request*, *Perform* and *Send*) presented in the form of Web services [MON 04]. The first two primitives enable synchronous communication and the two others enable asynchronous communication.

– The second part deals with interaction with the coordination agent. It recuperates messages from communication with the external applications part to translate them into queries comprehensible to the coordination agent. This part is provided in the form of an API Java, which can be imported into a Java application independently of PerSyst.

3.4.4. *Administration agent*

The administration agent enables the management and administration of PerSyst. Its main role is to enable a user (in the case of PerSyst, a developer) to interact with software agents for the adaptation and configuration of PerSyst. In view of all these constraints imposed by the existing tools of different agent platforms, PerSyst provides its own administration tool. This tool appears in the form of an agent that has the necessary skills enabling the administration and management of agents (it is explained in [ANL 06a]). It is associated with a graphic interface (see Figure 3.3).

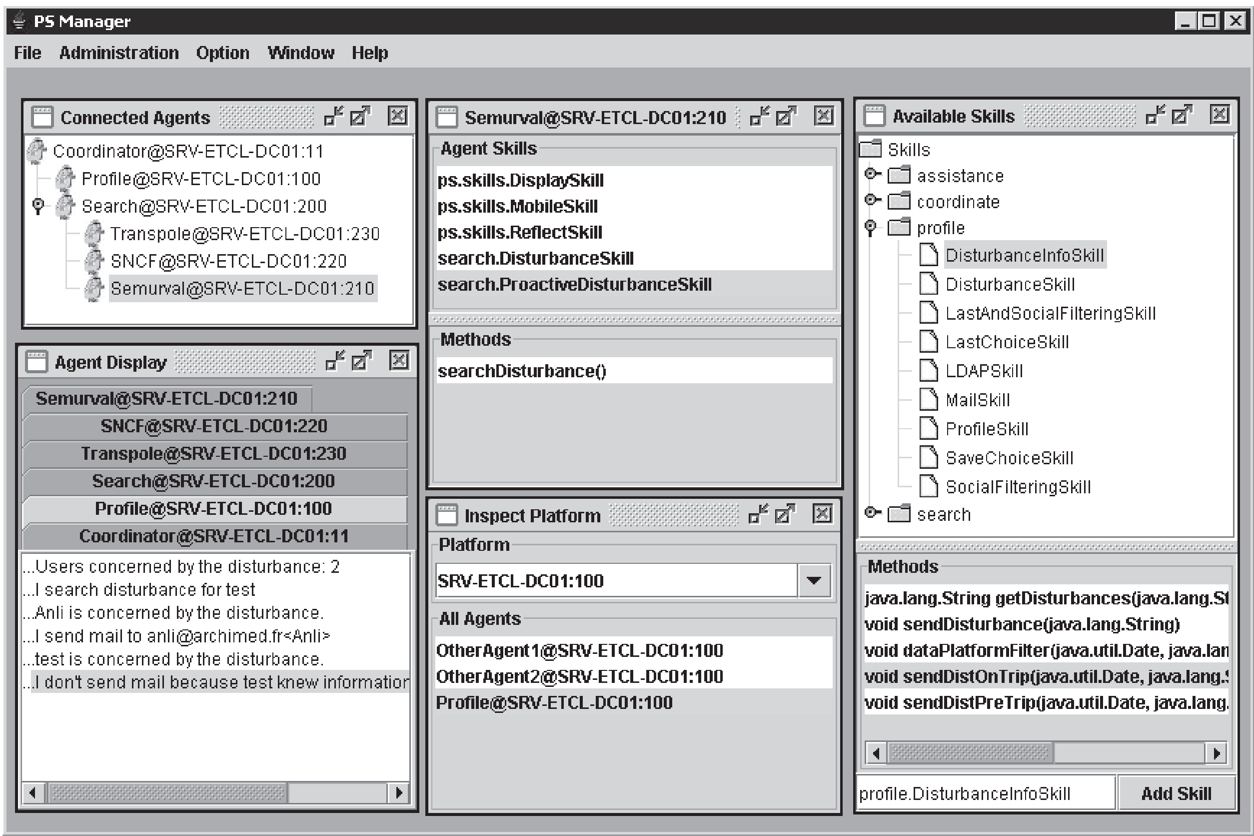

Figure 3.3. *The administration agent of PerSyst*

The administrator has a view of active agents, their organization and location (*Connected Agents* window), on skills (*Agent Skills*) and the content of their exchanges (*Agent Display*). It can create and remove agents, as well as acting on each of them to add or delete one of the available skills (*Available Skills*). It can also connect to an existing agent (*Inspect Platform*) or remove it from PerSyst.

3.5. Application to the public transport of people: itinerary search

This section describes an application of the PerMet method using the PerSyst PS. This application deals with a personalized itinerary search service in the area of the terrestrial transport of people. The objective is to facilitate access to multi-mode information for a user wishing to make a trip.

Indeed, the user is often confronted with an ensemble of disparate information (times and fares issued by the different operators, cards or tickets, etc., on paper, interactive terminals or by Internet) which sometimes turn out to be difficult to integrate into a specific and unique plan for a trip, adapted to their needs and preferences. In this context, personalization turns out to be a promising approach to take up this challenge. In the context of information to users, which integrates several modes of transport and their connections, our objective is to help the user in his approach to an information search and to provide him with a personalized result, i.e. all the necessary information and only the necessary information. The user provides the starting place, the arrival place and the arrival time. The system suggests an itinerary to him according to his preferences. The user has the possibility of validating the proposal or choosing another itinerary.

3.5.1. *Scenario*

The scenario acting as a base for illustration of the concepts is the following. It deals with the transport network given in Figure 3.4, simplified on purpose, but initially based on real data. In fact, it would have been possible to add other modes of transport, both current and future (see Chapter 1 on this subject): these modes can be individual (bicycle, rollerblades, electric or non electric wheelchair, Segway, etc.), public (tramway − one is being inaugurated in Valenciennes in parallel with our simulations), cyber-vehicles [MEL 10], [SER 08], auto-sharing, etc. Other modes, such as the taxi or carpooling, can be considered, both individual and public, according to their management system.

Thus, this simplified network (see Figure 3.4), describes the connection points enabling a user to carry out a trip from the LAMIH laboratory at the University of Valenciennes and Hainaut-Cambrésis, to the Archimed company in Lille (Valenciennes and Lille are two towns separated by 60 km). Three transport operators intervene in the realization of the journey:

– Semurval (bus operator), at the time of our first simulations, ensured the movement of the user in the town of Valenciennes and its surroundings. The transport modes considered in this scenario are mainly the bus, the car and walking.

– Transpole (bus and metro) is set up in the region of Lille. The transport modes considered in this scenario are mainly the bus and the metro.

– The SNCF (train) links the two towns of Valenciennes and Lille by the TER (regional express train) of the Nord-Pas-de-Calais region. We have also included the possibility that a user takes a car to go from the LAMIH to Archimed, without going via the public transport network.

We consider that the user has nine possible itineraries to go from LAMIH to Archimed. In the context of this search, the IS used is a Web portal developed based on the MASC[2] platform of the Archimed company.

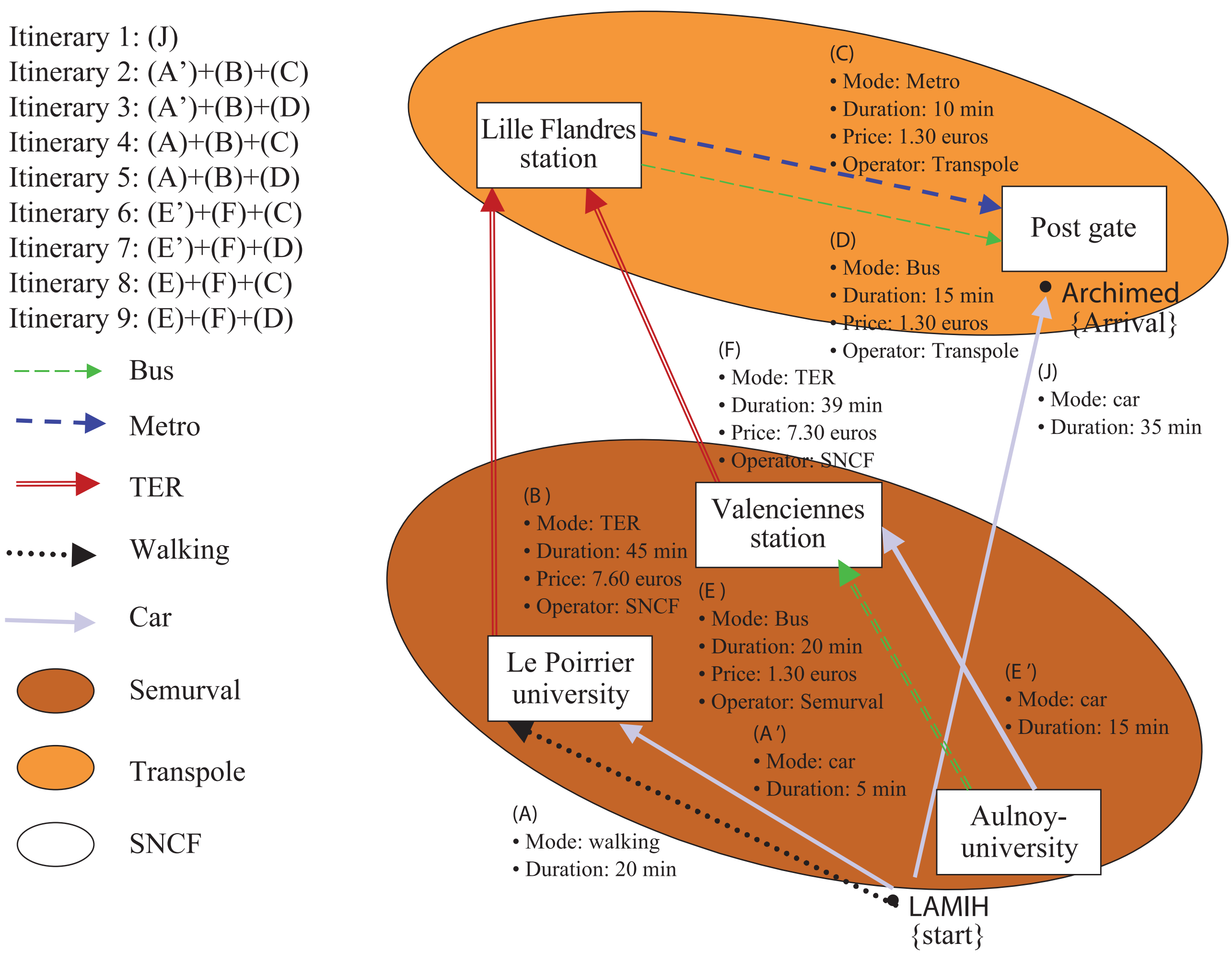

Figure 3.4. *Choice of itineraries from the trip, example*

3.5.2. *Analysis of the personalized transport service*

The main stages useful for global understanding of the personalized transport service analysis will be presented in succession. The interested reader will find a complete description in [ANL 06a].

3.5.2.1. *Capture of functional needs*

The functional needs have been modeled using use case diagrams. Figure 3.5 presents the functional needs for an itinerary search service.

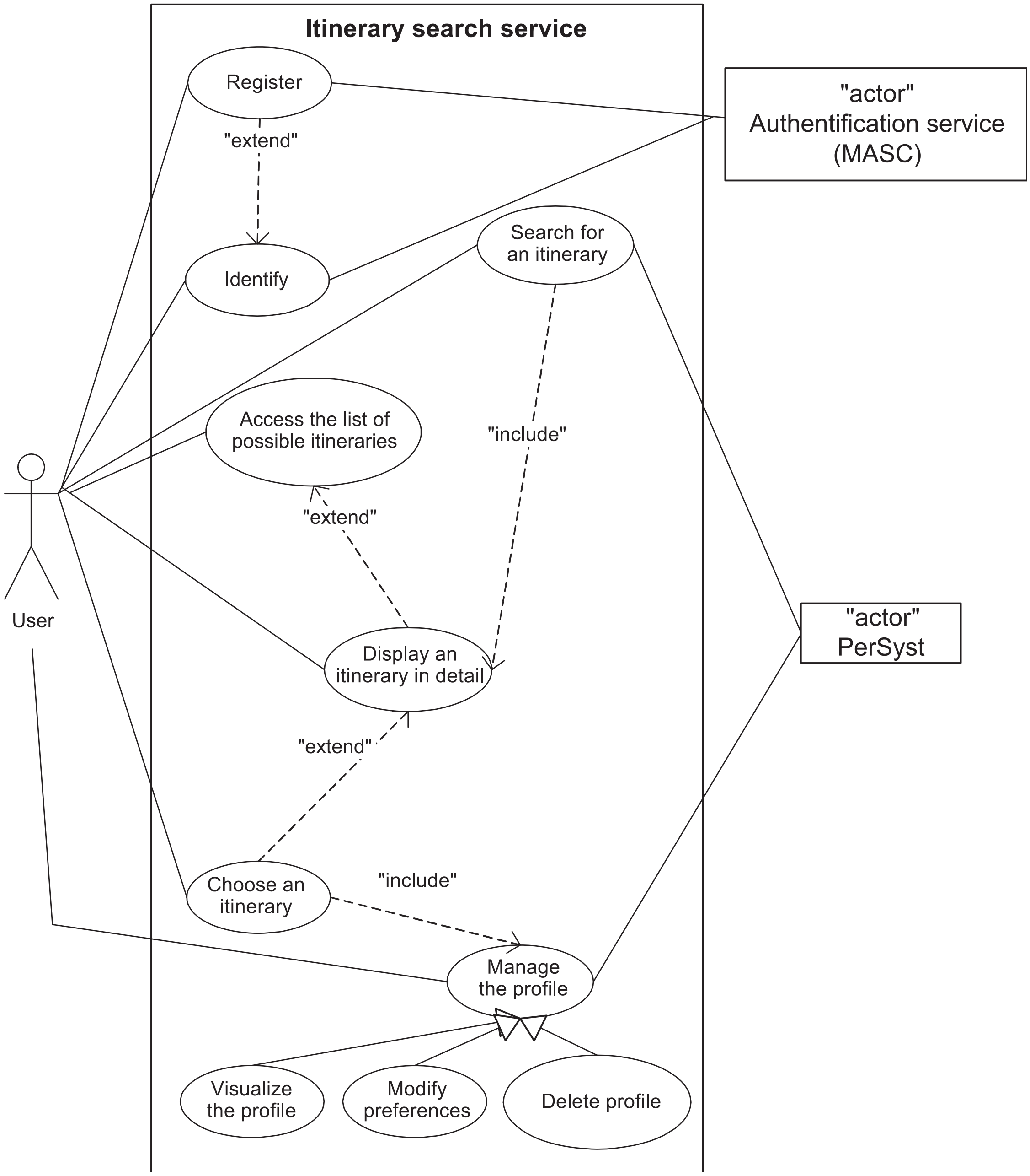

Figure 3.5. *Use case diagram of personalized itinerary search service*

The use case (each being associated with a textual description file) can be structured into three categories:

– the first category concerns identification of the user: use case *register* and *identify oneself*. These functionalities are already provided by the MASC platform and are directly used by the itinerary search service;

– the second category groups together the functionalities for the personalized itinerary search: use case search for an itinerary, access the list of possible itineraries, display the specifics of an itinerary and choose an itinerary;

– the third category concerns the use case for manipulation of the user profile: *manage the profile*, *visualize the profile*, *modify preferences and delete the profile*.

3.5.2.2. *Functional analysis of the personalized service*

Starting from the use case diagram stemming from the capture of functional needs, a static modeling of the business objects of the itinerary search service is carried out (see Figure 3.6).

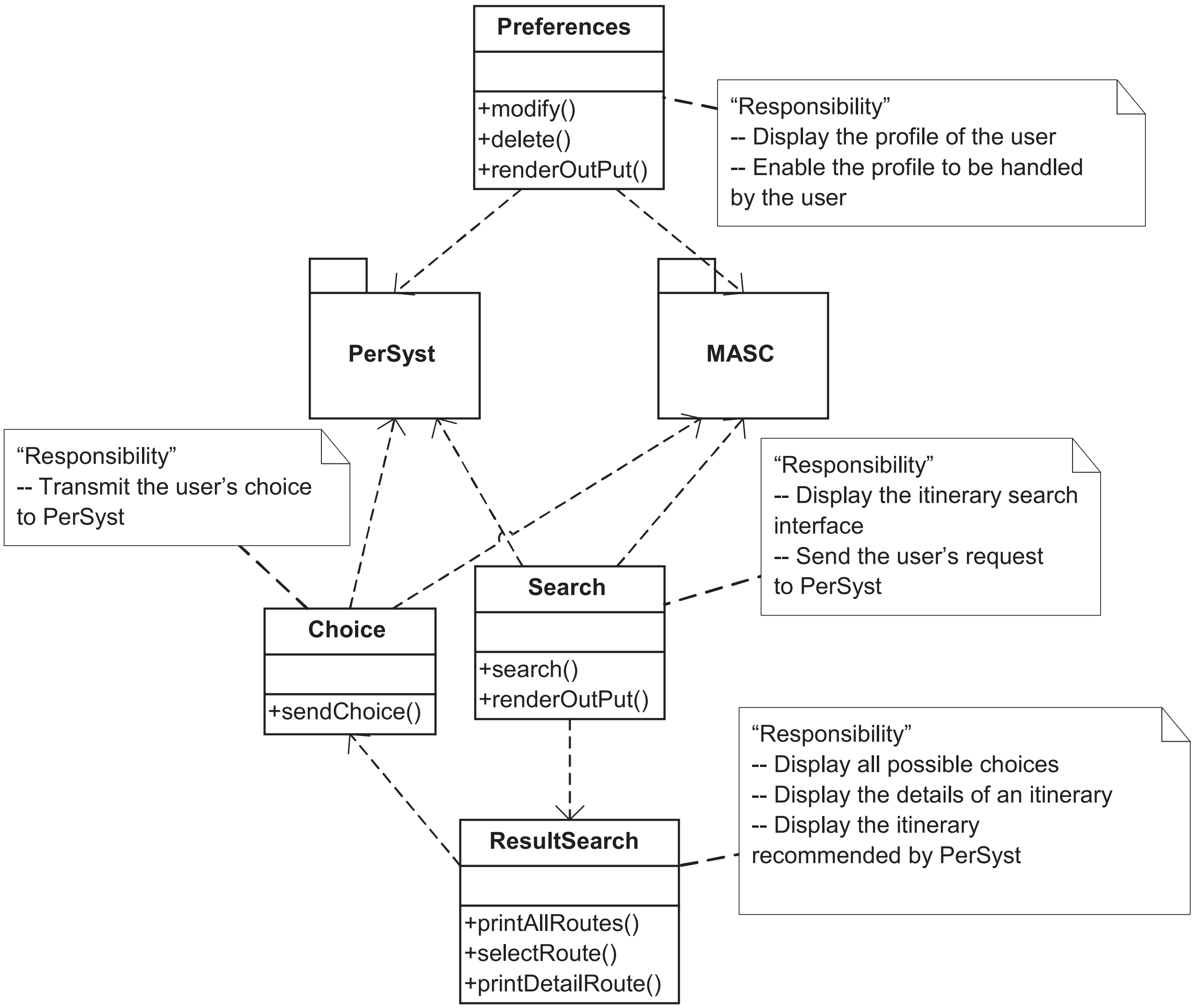

Figure 3.6. *Class diagrams for the itinerary search services*

The classes necessary to ensure the different responsibilities are defined. The modules that are not part of the service (but are useful to it) are represented in the form of *packages*. To refine the static model, a series of dynamic models was established. For example, Figure 3.7 presents a scenario for the itinerary search by giving details of the interactions between different objects of the service. After connecting to the MASC, the user begins his request using the personalized itinerary search service, *Search*, which asks the name of the user connected to MASC and sends the completed request of the user name to PerSyst. PerSyst sends all the possible itineraries answering the request to *Search* and recommends an itinerary that is likely to interest the user. *Search* creates *ResultSearch*, which provides the personalized results to the user.

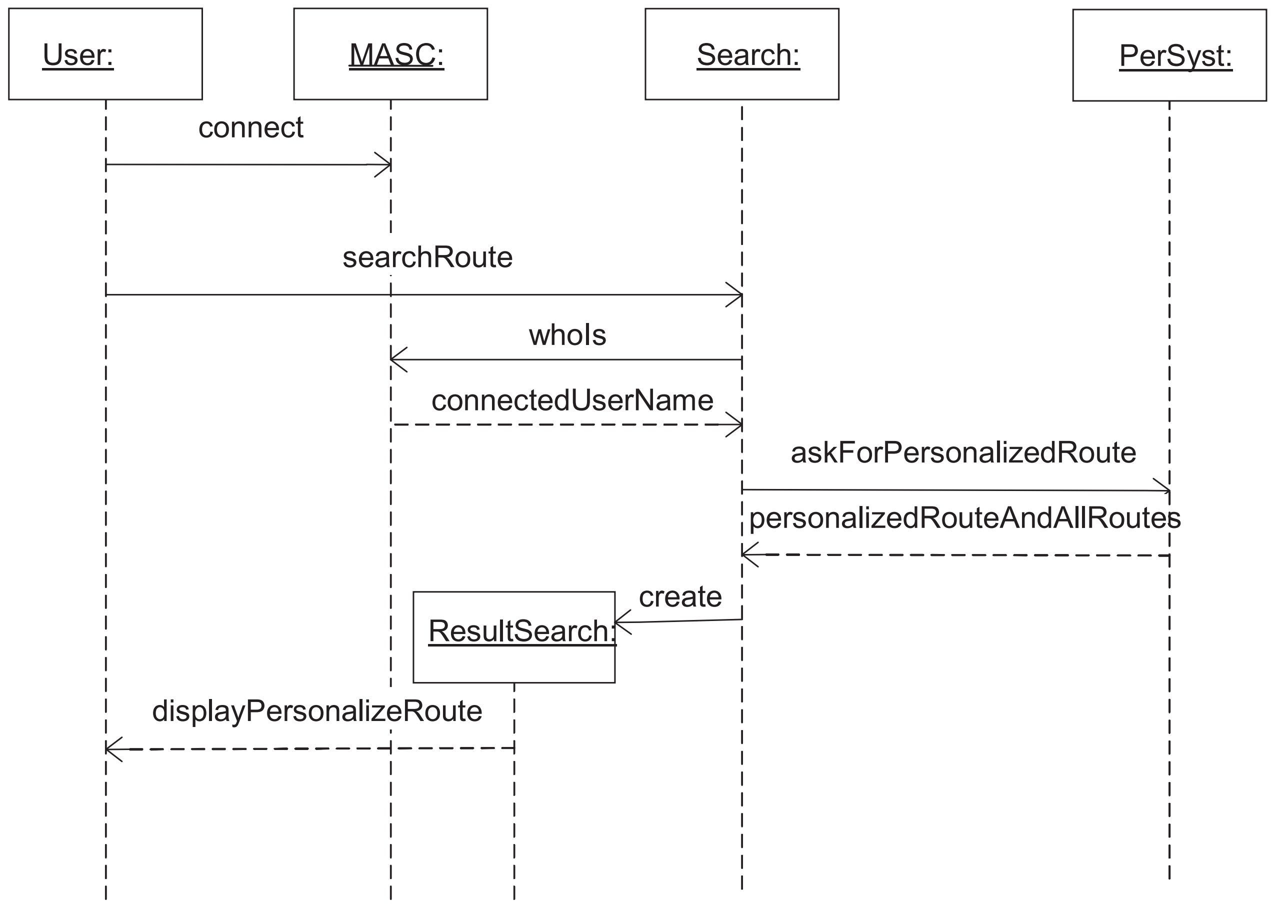

Figure 3.7. *Sequence diagram for the personalized itinerary search*

The data models that the service should exchange with the PS are defined (see Figure 3.8).

A request specifies the username of the user (DN) and the data (*Departure*, *Arrival*, *ArrivalTime*, etc.) for the itinerary search. A response (*Response*) takes the request (with the DN) and adds all the possible itineraries (*Result*).

This response is associated with a choice (*Choice*), which enables the itinerary likely to interest the user to be referenced. An itinerary (*Result*) is made up of several ways (*Way*) and includes its own characteristics (*Criteria*).

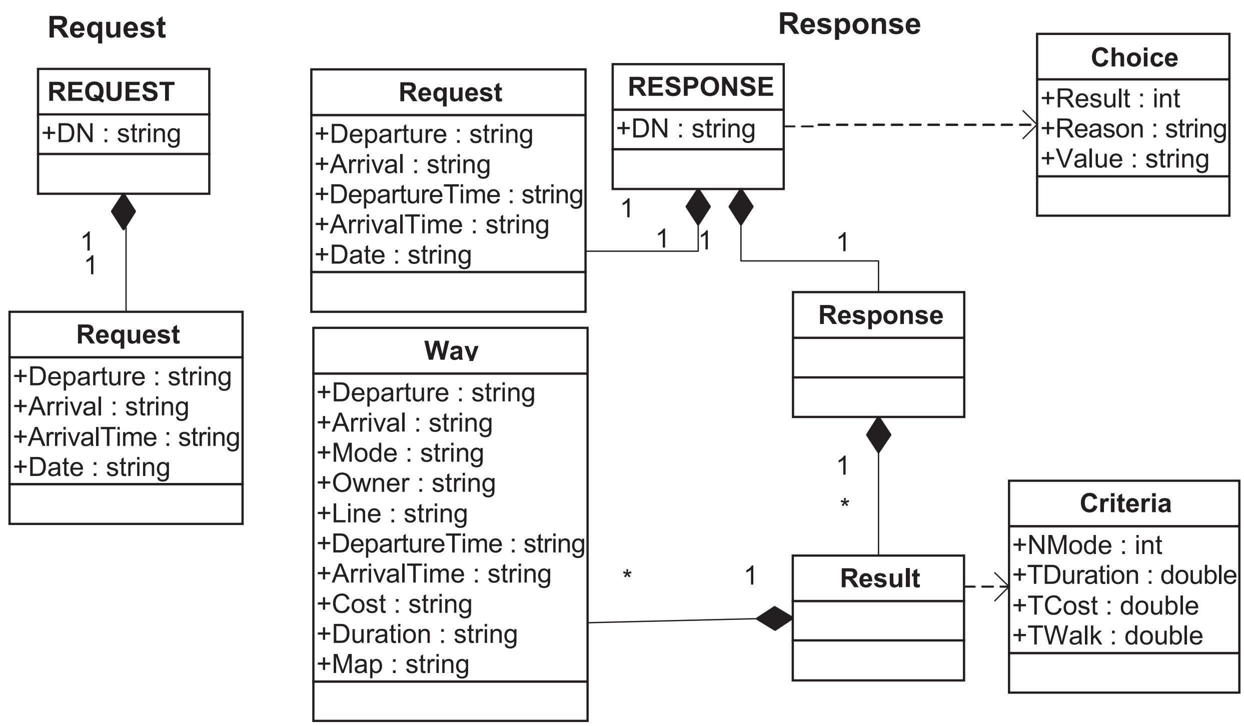

Figure 3.8. *Data models exchanged between the itinerary search service and PerSyst*

3.5.2.3. *Capture of the technical needs*

The application has a server where *Windows 2003 Server* is installed. All the software components are installed on this server. The user accesses the itinerary search service from his own work station (which can be a personal computer, a PDA, etc.) using the Internet network (see Figure 3.9).

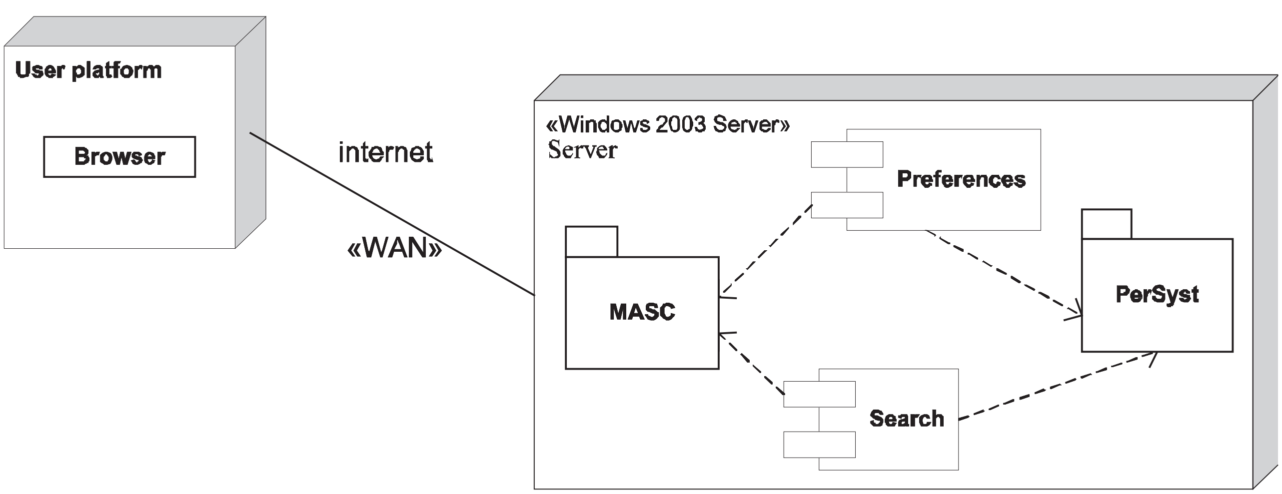

Figure 3.9. *Capture of technical needs for the itinerary search service*

3.5.2.4. *Generic design of the personalized service*

The service is broken down into three subsystems (manipulation of user preferences, itinerary searches and access to PerSyst; see Figure 3.10). Each subsystem corresponds to a grouping of modeling elements providing the same behavior unit. For example, the "access to PerSyst" subsystem contains the *PersonalizeSystem* class.

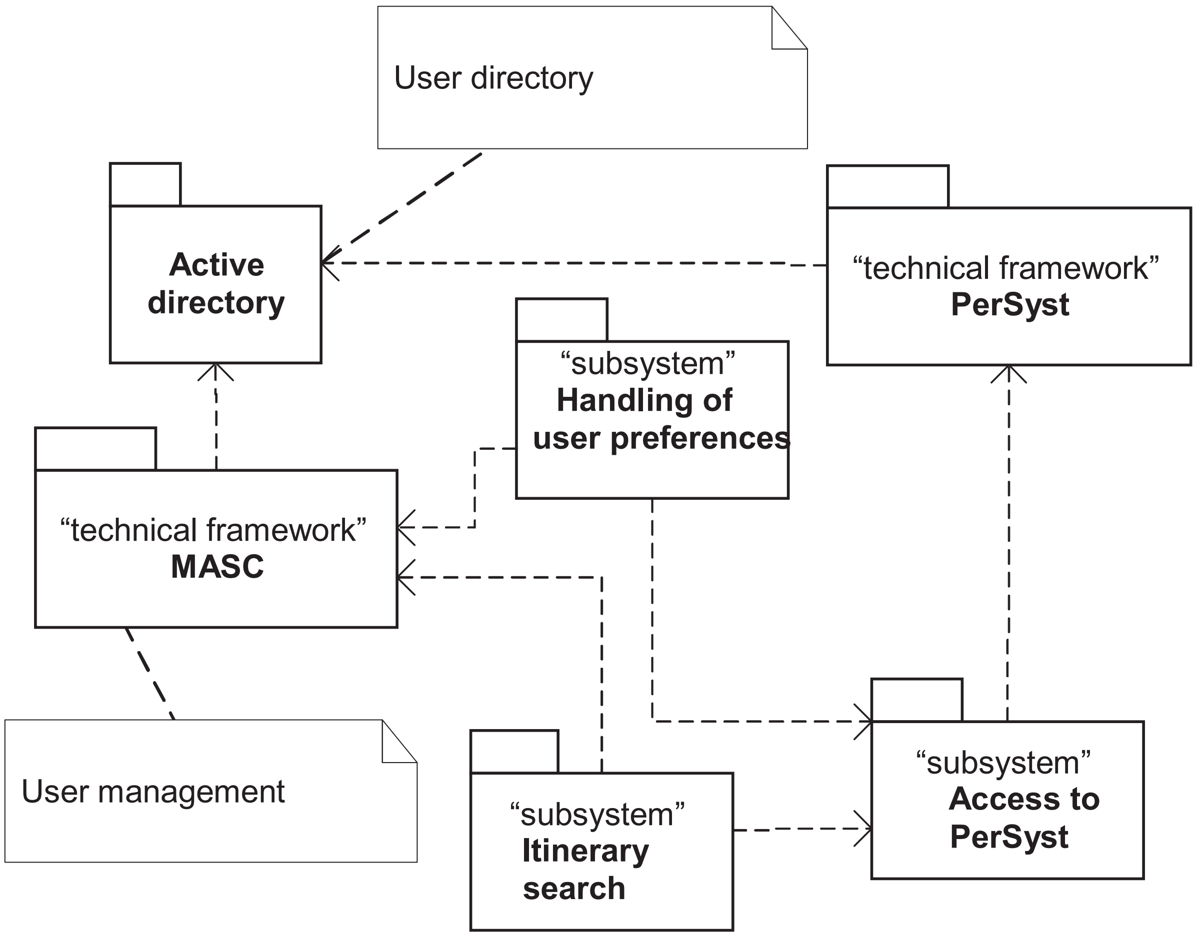

Figure 3.10. *Generic design for the itinerary search service*

3.5.3. *Design of the personalized service*

The service being a Web application, we have used the UML extensions put forward by [CON 00] for conceptual modeling of the service.

Figure 3.11 presents the diagrams of conceptual classes for personalized itinerary search (without the classes for handling of the user profile, nor those for authentication of the user).

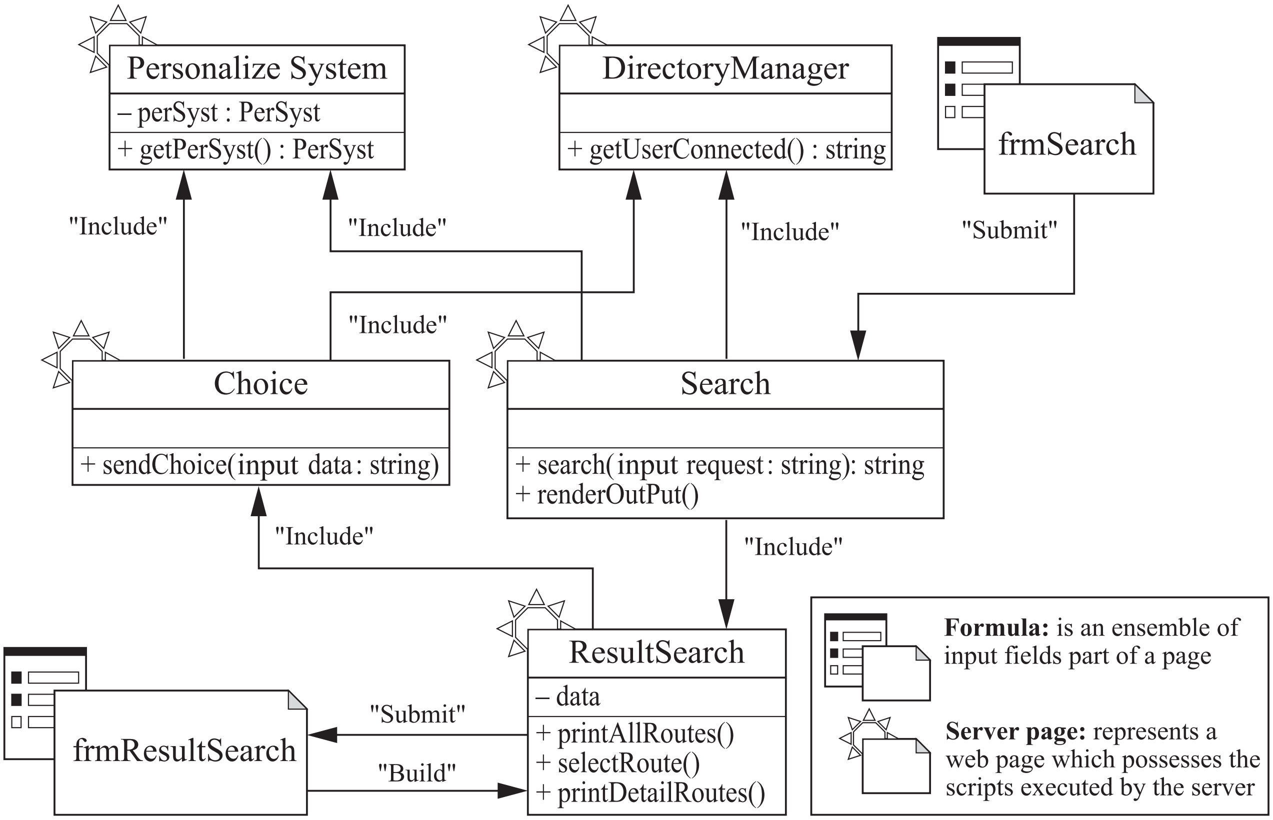

Figure 3.11. *Diagram of design classes for the itinerary search service*

3.5.4. *Implementation of the personalized service*

The applications (*Search*, *ResultSearch*, *Choice*, etc.) have been implemented through *ASP.NET* pages. Implementation of the *PersonalizeSystem* module provides a function (*getPerSyst*) that has the objective of sending back an object enabling PerSyst primitives to be invoked.

This object determines the data that the service and PerSyst should exchange while respecting the data models defined in the phase of service analysis. These data are transcoded in the XML format (Figure 3.12 gives an example of XML data for an itinerary search request)[2].

The presentations are carried out via an XSLT transformation using Archimed's JSE[3] transformation motor. Figure 3.13 presents the itinerary search page also enabling the visualization and modification of user preferences.

2 XML (*eXtensible Markup Language*, www.w3.org) is a standard for the exchange of data between applications. It also facilitates the development of an adaptive interface [HAB 04] that is multi-targeted [PUE 02] and multi-modal (with VoiceXML [ROU 04] or UsiXML [STA 05]) and even multi-context [LIM 05].

```
- <REQUEST>
    <DN>LDAP://SRV-ETCL-
      DC01:389/cn=john,OU=Sitp,DC=DOMTRANSPORT,DC=local</DN>
  - <Request>
      <Departure>LAMIH</Departure>
      <Arrival>Archimed</Arrival>
      <DepartureTime>07:00</DepartureTime>
      <Date>12/07/2006</Date>
    </Request>
  </REQUEST>
```


Figure 3.12. *Example of XML data for an itinerary search request*

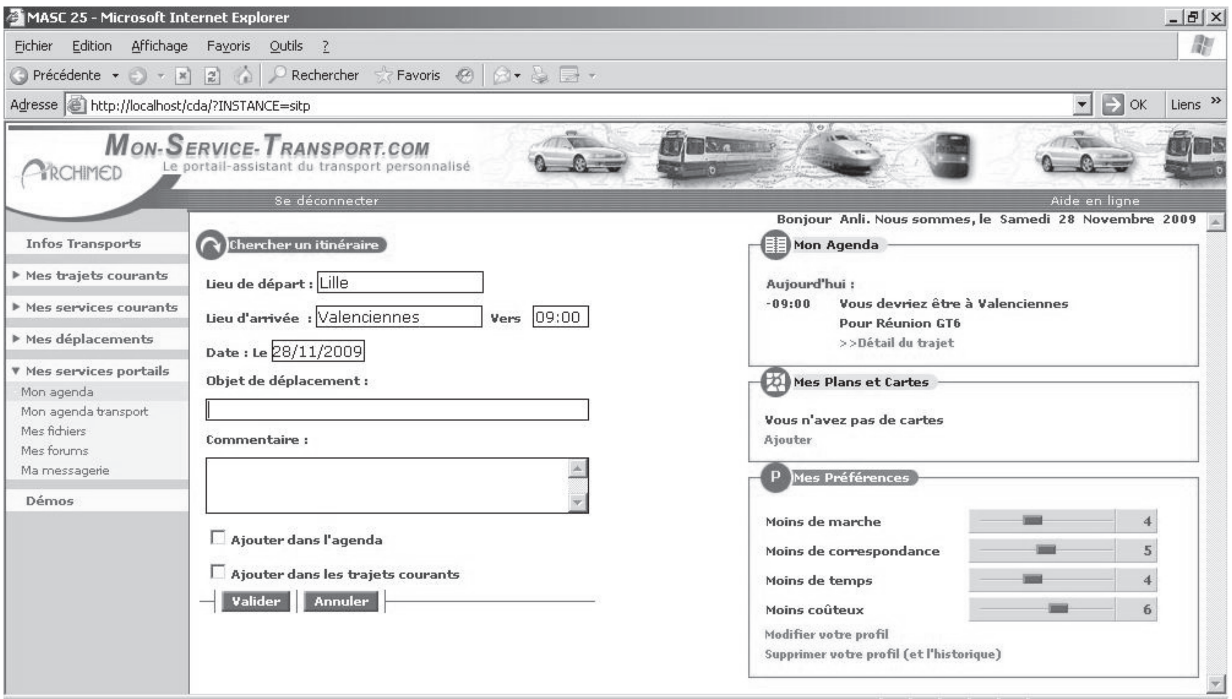

Figure 3.13. *Itinerary search page including the visualization and modification of the user profile*

At the end of implementation of the service, preliminary tests of usability and verification of the functionalities of the PS were conducted. Even if these tests used simulated data (in the *PersonnalizeSystem* module), they were useful as they enabled us to add attributes (*Reason*, *Value*) in the *Choice* class of the data model of responses sent by PerSyst. These attributes were necessary to provide the user with an explanation of the recommendations made by the PS.

3.5.5. *Analysis of constitutive agents of the personalized system*

3.5.5.1. *Analysis of agent models*

The use case diagram in Figure 3.14 shows PerSyst as a user of itinerary search service. In this application, we assume that there is a federative authority that groups together the databases of the Transpole, Semurval and SNCF operators. We have simulated this authority by creating an XML database enabling the itinerary search for a trip from Valenciennes to Lille. The user information is stored in the *Lightweight Directory Access Protocol* (LDAP) directory, which is the same one that MASC uses for user management.

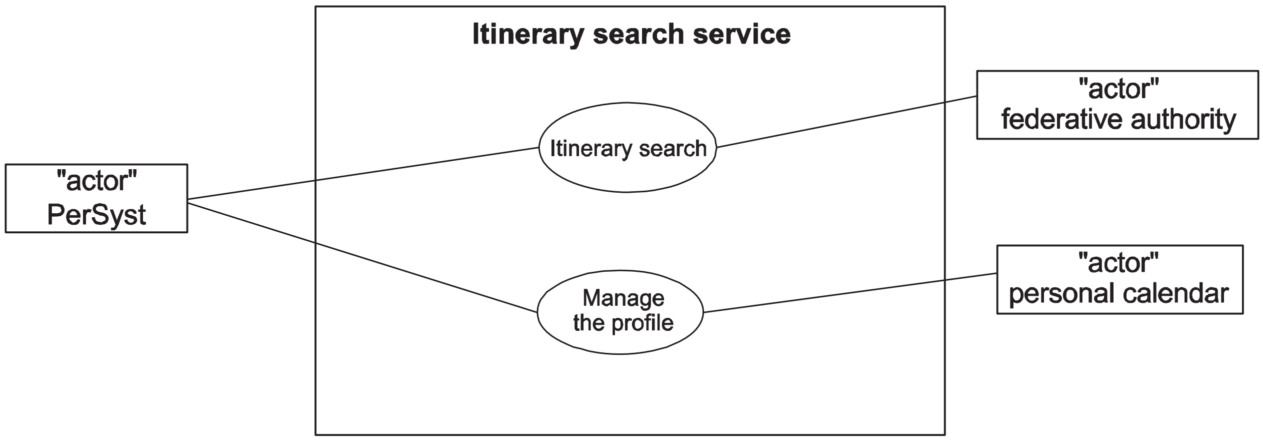

Figure 3.14. *Use case diagram of the itinerary search service seen from the perspective of personalization*

By analyzing this use case diagram and following the rules of the PerMet method for agent model analysis (see section 3.4.4), we obtain an agent model for the itinerary search and an agent model for management of the user profile. Indeed, rule 1 (an agent model of expected functionality of the service) and rule 3 (an agent model by external resource), led us to an agent model for the itinerary search and for the management of user information. Rule 2, which stipulates that one agent model is needed per interaction platform, will not be applied here because in this application the objective is to leave management of the interaction between the user and service to the MASC platform. The user accesses the service via a Web browser. The different roles that these agents must play are described in Figure 3.15.

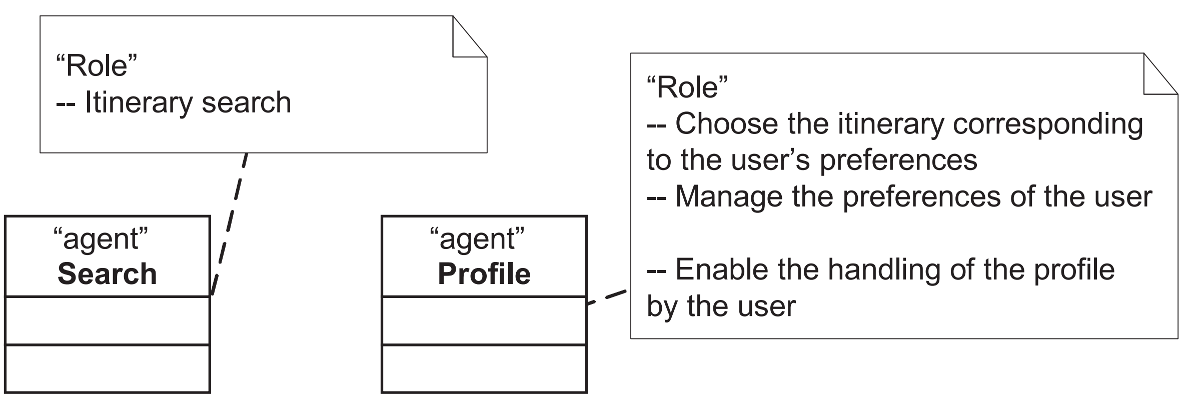

Figure 3.15. *Agent models for the itinerary search service*

3.5.5.2. *Identification of contacts*

Figure 3.16 represents the interactions of the agents for the itinerary search. Let us note the intervention of the two PerSyst agents: the communication agent that enables the transmission of messages which come from the itinerary search service integrated into MASC; and the coordination agent that ensures coordination of messages between the *Search* agent and the *Profile* agent. The analysis of this diagram enabled us to define the communication links described in Figure 3.17.

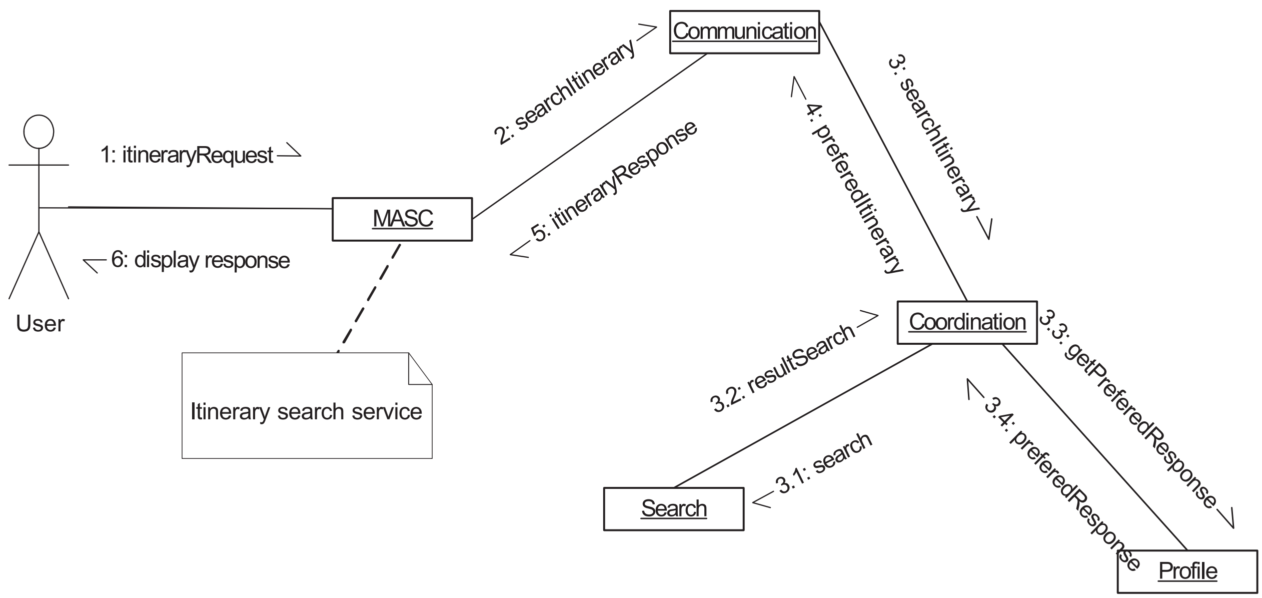

Figure 3.16. *Interaction between agents for the itinerary search service*

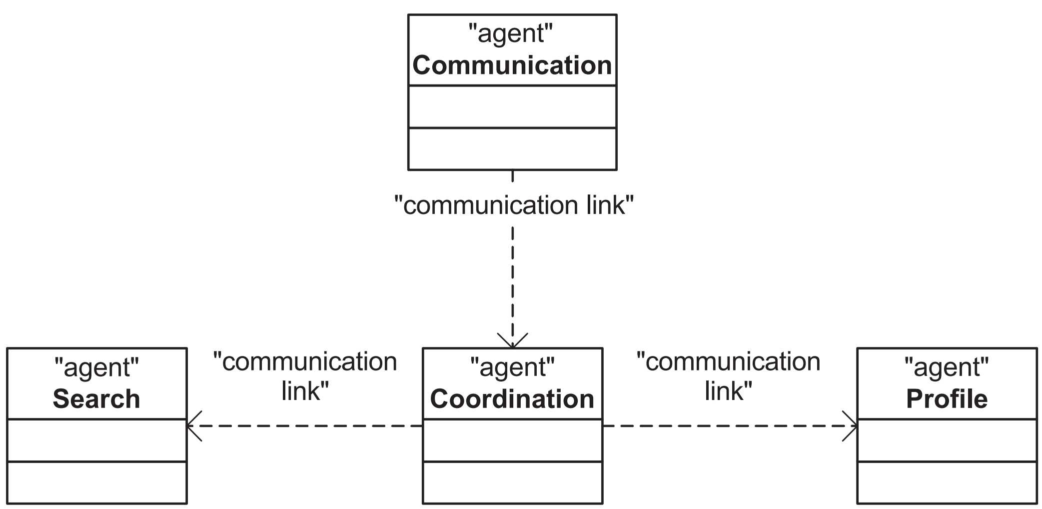

Figure 3.17. *Communication links between agents for the itinerary search service*

3.5.5.3. *Analysis of behaviors*

We have analyzed the behaviors that each agent model must have in order to carry out its roles. The analysis of agent models enabled us identify the behaviors of the *Search* and *Profile* agents.

For example, Figure 3.18 presents the behaviors required by the *Profile* agent for the itinerary search service.

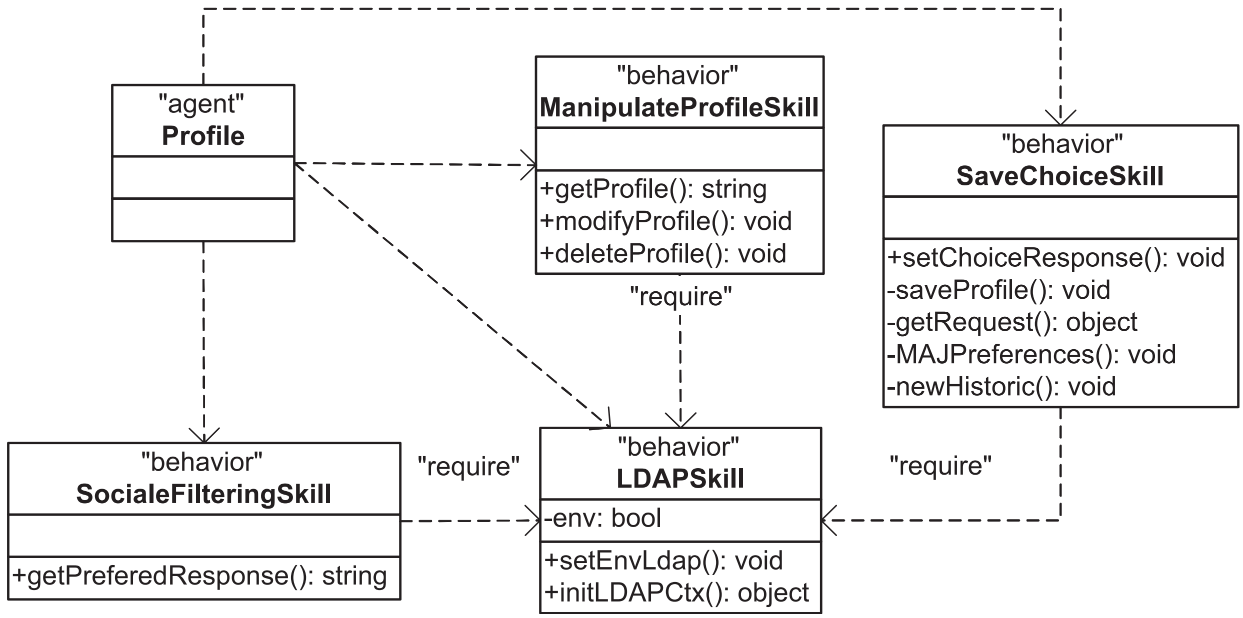

Figure 3.18. *Profile management agent skills*

3.5.5.4. *Deployment information*

In the context of our prototype, all PerSyst agents are located in the same machine (see Figure 3.19). It is the same machine as the one where the itinerary search service is deployed.

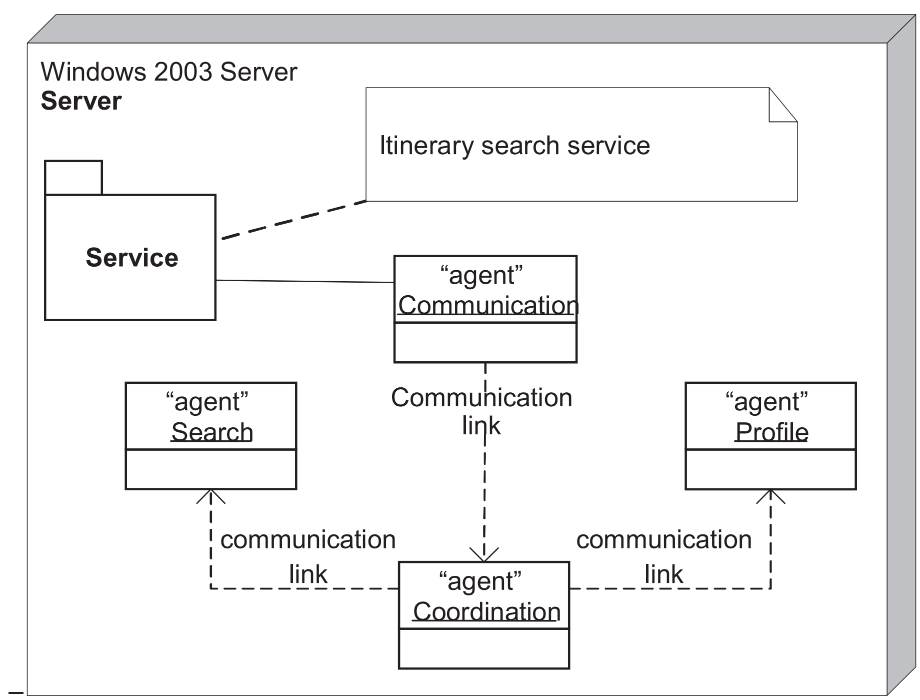

Figure 3.19. *Agent deployment*

3.5.6. *Design of agent behaviors*

In this phase, for each agent model we have refined and specified the agent behaviors. For example, for the profile management agent we have applied our user model put forward in [ANL 06a].

This model is made up of three parts:

– static data represent the authentication information (*login*, password) and the user's personal data (surname, name, place of work, etc.). These data come from the LDAP directory where MASC saves the user's information when he registers;

– the weighted data models the preferences of the user in relation to transport criteria (least connections, the quickest, least amount of walking, the cheapest). Each criterion is associated with a mark between 0 and 10, which represents the degree of preference of the user in relation to this criterion;

– the history keeps track of the user interactions with PISs. This can act as a knowledge base for the updating of static and weighted data. Analysis of the information can, for example, inform the system that the user lives in Valenciennes and works in Lille (as the user leaves for Lille in the morning and comes back to Valenciennes in the evening, except on bank holidays). By analyzing the itineraries chosen by the user, the system can deduce the preferences of the user in relation to the modes of transport.

This user model is stored with the profile management agents in the form of an XML document (an example is available in [ANL 06a]). Figure 3.20 presents the activities carried out during the execution of the internal action *MAJPreferences* for the skill *SaveChoiceSkill*. The objective here is to deduce the preferences of the user according to criteria associated with the itineraries (least connections, the quickest, the least amount of walking, the cheapest).

Figure 3.21 presents the activities carried out during the execution of the *getPreferedResponse* service of the *SocialFilteringSkill* skill. To select the itinerary that is likely to interest the user, a majority vote (select the itinerary that was the most chosen by users) is carried out on the itineraries if the current user has no profile. If the user possesses a profile, and has already made his request, the itinerary that he chose will be recommended. If the user has a profile but has never made a request, a collaborative filtering[3] (it is the collaborative filtering method based on the preferences and the behaviors of user which is applied) is carried out to choose which itinerary to propose. This model therefore combines a cognitive method (recommendation in relation to the profile) and social methods (majority vote and collaborative filtering).

---

3 A synthesis of collaborative filtering techniques is available in [SU 09].

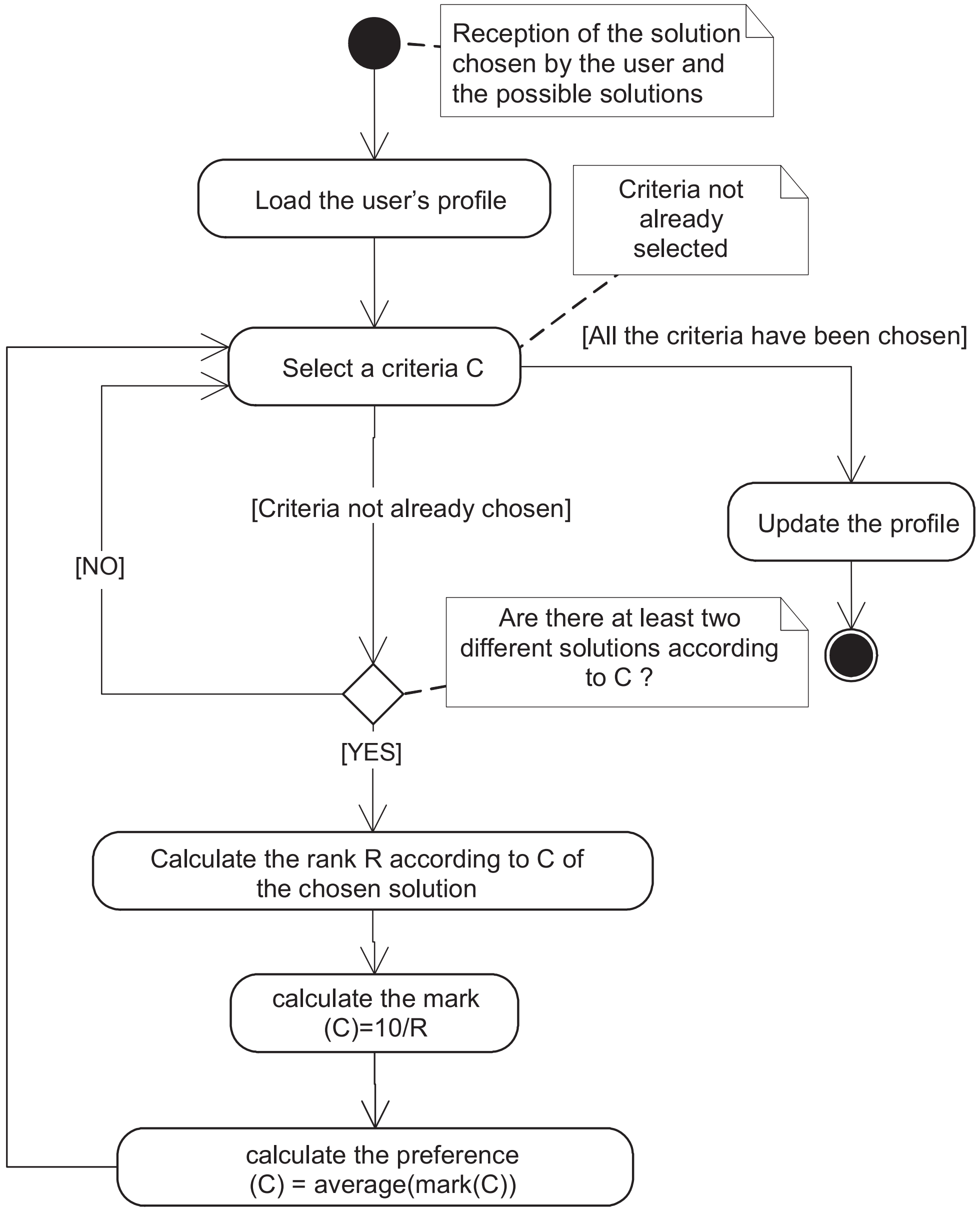

Figure 3.20. *Activities for the updating of user preferences*

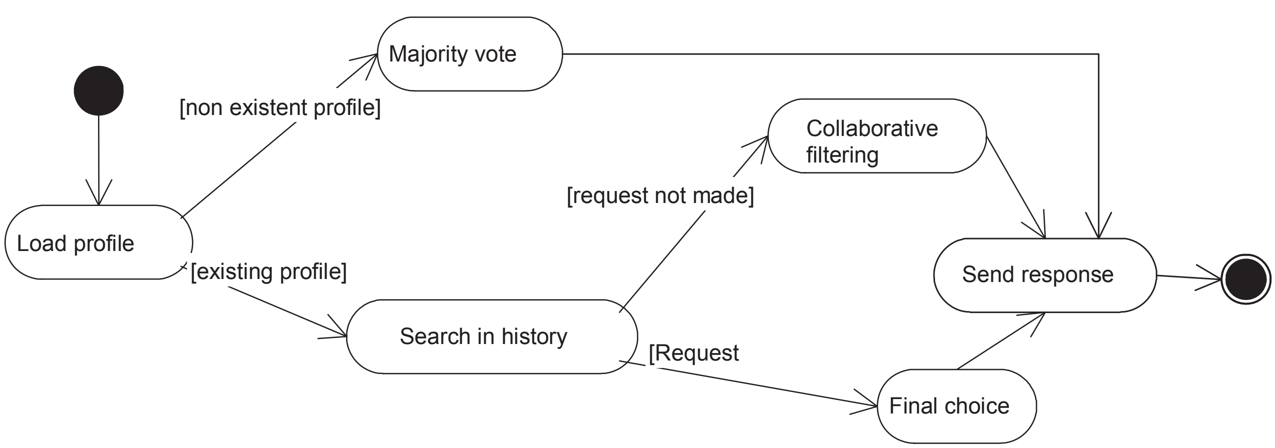

Figure 3.21. *Activities for the choice of itinerary*

3.5.7. *Implementation of the agent behaviors*

The agent skills were implemented and tested as they were being developed. The creation of agents, the association of their skills and deployments were carried out via the graphic administration interface of PerSyst. After the deployment of agents, verification tests for the smooth running of MAS, which makes up PerSyst, were carried out by a simulation of data that should come from external applications.

3.5.8. *Integration*

The communication between the applications of the itinerary search service and PerSyst goes via the *PersonalizeSystem* module. The *getPerSyst* function was implemented for it to provide a reference of an object enabling the primitives of PerSyst to be invoked via SOAP messages [NEW 03]. The data coming from PerSyst are displayed on the Web pages of itinerary search service (see Figure 3.22). The *Request* primitive was used for the itinerary search to enable the user to visualize his profile. The *Send* primitive was used to send the user's choice to PerSyst and for modification of the user profile.

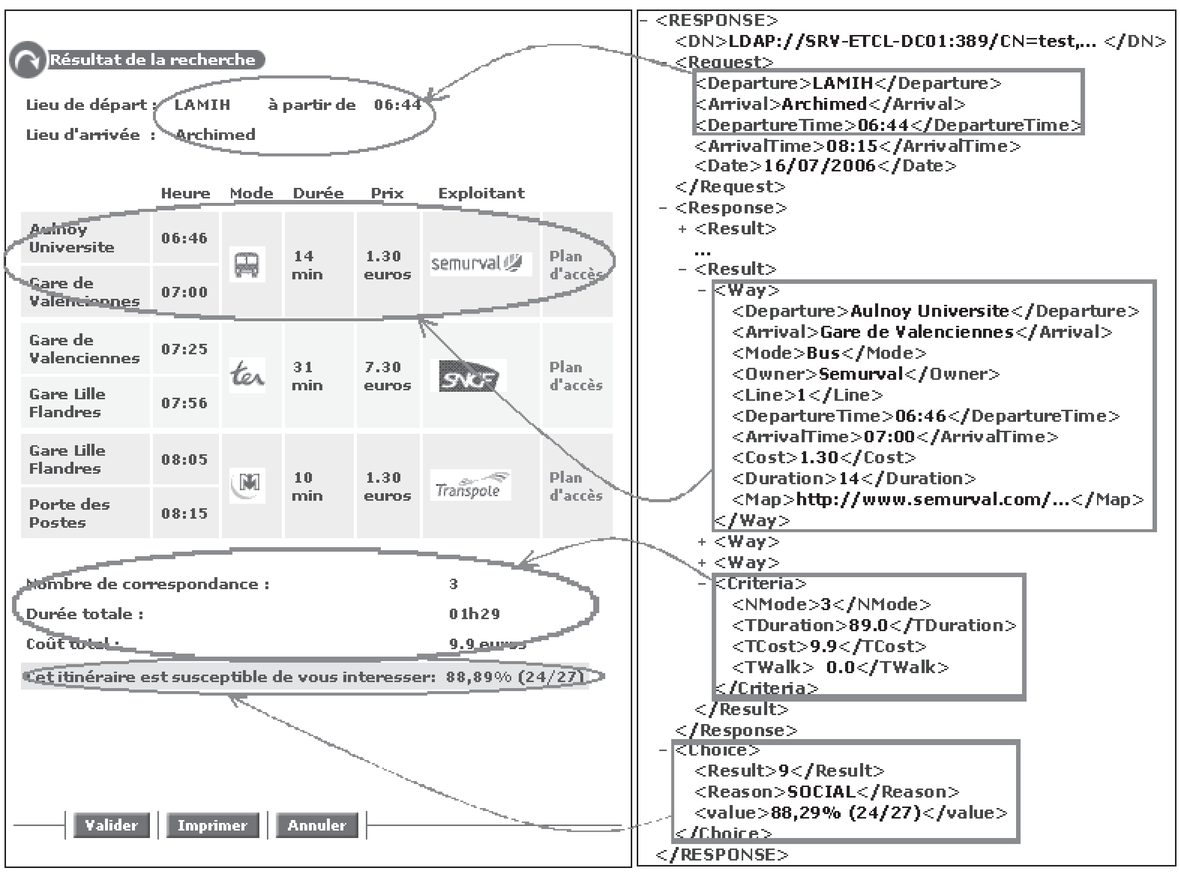

Figure 3.22. *Use of data coming from PerSyst by itinerary search service*

3.5.9. *Evaluations*

Here, it is a matter of a global evaluation combining the IS part and PS parts. More specifically, different evaluations were carried out. Three of them are described in the following; they are within the context of the PREDIM Mouver.Perso project (Mobility and mUltimodality travelers studying in the Nord-Pas-de-Calais region – personalized multimodal information system) [GRI 07], bringing together LAMIH, INRETS and Archimed.

3.5.9.1. *Functional and technical evaluations*

The functional evaluations were fairly quick as they had already been carried out in the implementation phase of the service. We checked in particular that the functionalities of the service at the end of the integration phase still conformed to those defined in the stage of capturing functional needs. Technical evaluations of the service were carried out. The conformity of the HTML code produced in relation to several browsers was also checked.

3.5.9.2. *Performance evaluation*

We carried out tests to evaluate the performance of the personalized itinerary search service. These tests involved the gathering of service response times for an itinerary search request. We measured the average response time in relation to the number of users contained in the user base. Figure 3.23 presents the response times according to the filtering method used with the profile management agents. A cognitive method (filtering based on the profile of the user: last made choice) and two social methods (the first is that of the majority vote; the second is a method of collaborative filtering based on the preferences and behaviors of users on the basis of a Bayesian network, put forward in [ANL 06a]) were tested. We expressly chose these three methods as they are included in the activity model of the profile management agent for the choice of itinerary.

The results obtained show that when it is a matter of a majority vote, the response times increase according to the number of responses registered in the system, but that these times remain acceptable (less than a second for 500 users). For filtering based on a Bayesian network, the response time is exponential from 100 users. Above 100 users, the response time exceeds 3 seconds. We believe we can improve performances by improving the algorithm implemented. For filtering based on the user profile, the number of users registered in the system does not influence the performance of the system. Other performance evaluations have been envisaged, for example evaluation of the performance of the service in relation to the number of users simultaneously connected to the service or study of the impact of the physical distribution of PerSyst agents on the global performance of the IS.

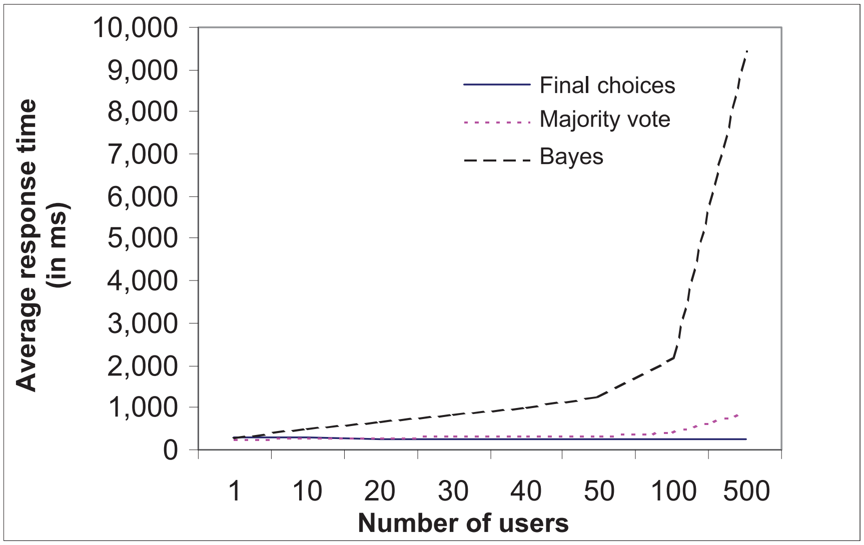

Figure 3.23. *Response time of the system according to the number of users registered with the service*

3.5.9.3. *Ergonomics evaluation*

The objective of this evaluation was to gather, in the field, the opinions of the people mainly targeted by the application. We therefore carried out a study in a student population at the University of Valenciennes in France [GRI 07] (see Figure 3.24).

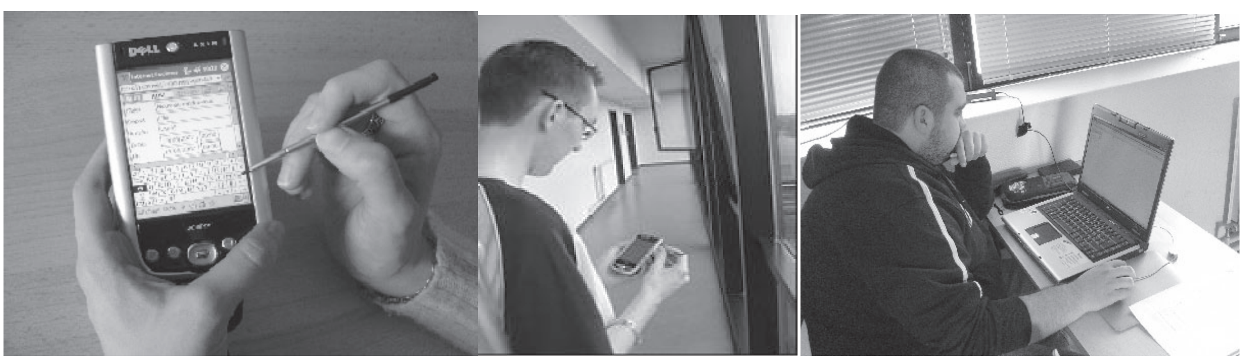

Figure 3.24. *Subjects using the PIS on PDA and PC*

An original evaluation approach [SOU 07], [SOU 08a], [SOU 08b] was put forward, broken down into three phases[4]:

– A preparation phase: the evaluator must choose tasks that are representative of the system and prepare a general questionnaire on the experimentation subjects and one form per criterion evaluated. The task chosen here is the itinerary search after

4 This research on the evaluation of a PIS was extended in the context of a thesis [SOU 10].

the addition of a rendezvous. This task has the uniqueness of requiring the consultation of at least three application interfaces.

– A test phase of the model: the evaluator presents the chosen tasks to the subjects as well as the evaluation criteria and their definition, and asks them to select the most important criteria according to their needs and preferences. Then the subjects evaluate the system according to the defined criteria, during the execution of tasks, with a mark representing their level of satisfaction. Forms allowing the possible problems to be noted are also available. Twenty-three people participated in this evaluation (two evaluators-experts, 20 students-subjects and one technician). The length of an evaluation period was around three hours per subject.

– An analysis phase: the evaluator calculates the level of satisfaction of each user relatively to each criterion, and their average levels of satisfaction.

The system is evaluated according to seven criteria, of which three are linked to content and four are to the container:

– *Personalization of the content* (see Figure 3.25): 47% of users are very satisfied, 41% are satisfied, 6% not very satisfied and 6% of users are not at all satisfied. Figure 3.25 shows that the subjects are favorable to the quality of personalization linked to preferences and to the experiment (average > 0.6). Concerning the experiment, subjects spoke of the lack of help during first use. The consideration of interests (professional or personal) is less satisfying.

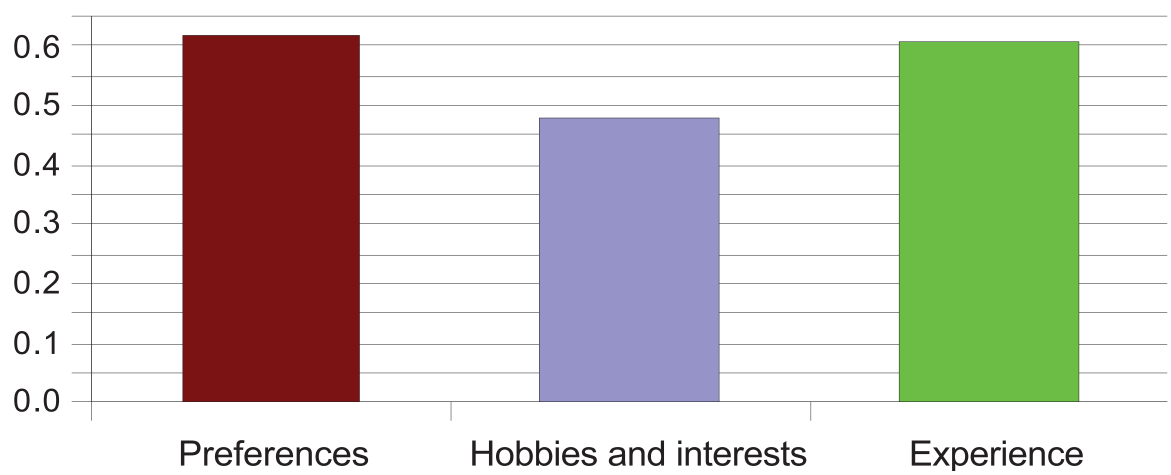

Figure 3.25. *Average satisfaction levels according to criteria linked to content*

– *Personalization of the container* (see Figure 3.26): the satisfaction rate is 33% for very satisfied subjects, 17% for satisfied subjects and 50% for not very satisfied subjects. After having tested the application using two different interactive media (PC and PDA), the subjects gave a favorable opinion for adaptation of the system to the interactive platform. As for the adaptation to behavior (adaptation to the goals and plans of the user), most respondents gave a neutral response. Students had an unfavorable opinion concerning the adaptation of the application to the physical capacity of the user and to the environment (luminosity, noise and geographic localization). With accessibility, a problem cited several times, respondents commented on the smallness of characters, which resulted in poor legibility. A recurring problem with behavior was the lack of "feedback" during validation actions.

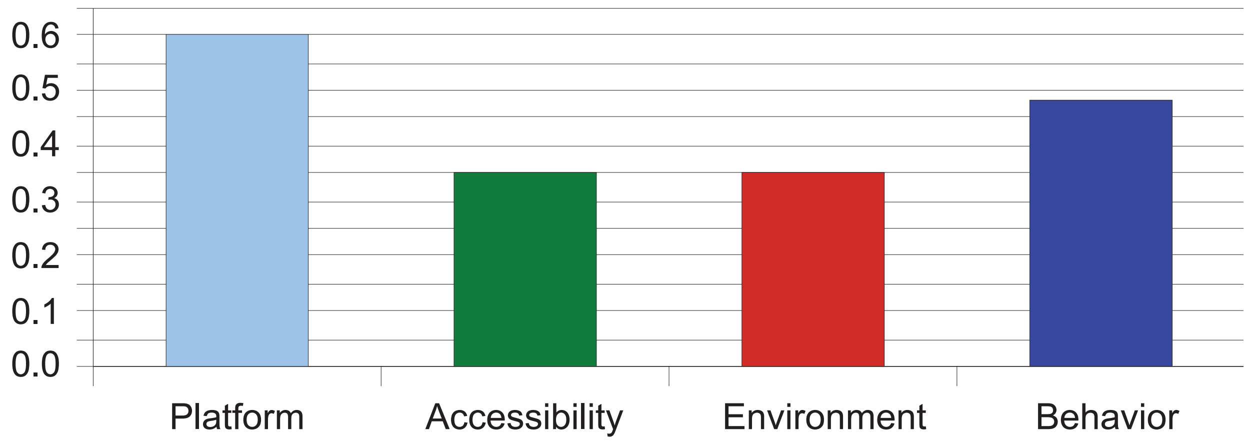

Figure 3.26. *Average satisfaction levels according to criteria linked to the container*

Other general results linked to utility and usability (in the sense of [NIE 93]) were obtained:

– Utility determines whether the interface meets the needs of the user: 55% were satisfied (satisfaction level [0.75; 1.0]), 40% satisfied ([0.5; 0.75]) and 5% not very satisfied ([0.25; 0.5]). Users make few mistakes.

– Usability accounts for the quality of the human-machine interaction in terms of how easy it is to learn and use. The satisfaction rate of the users in terms of usability is 41% satisfied and 59% very satisfied. These good results can be explained by the fact that the service is relatively simple, and learning it is quick and easy.

Such results are promising and call for other experiments and developments.

3.6. Discussion about the possibility of generalization relative to personalization

Elements of personalization applied to traveler information have been described in this chapter. However, it seems to us that it is possible to go much further in terms of personalization if we generalize the idea of having a detailed schedule of the activities of the user, made available to intelligent software agents at their service.

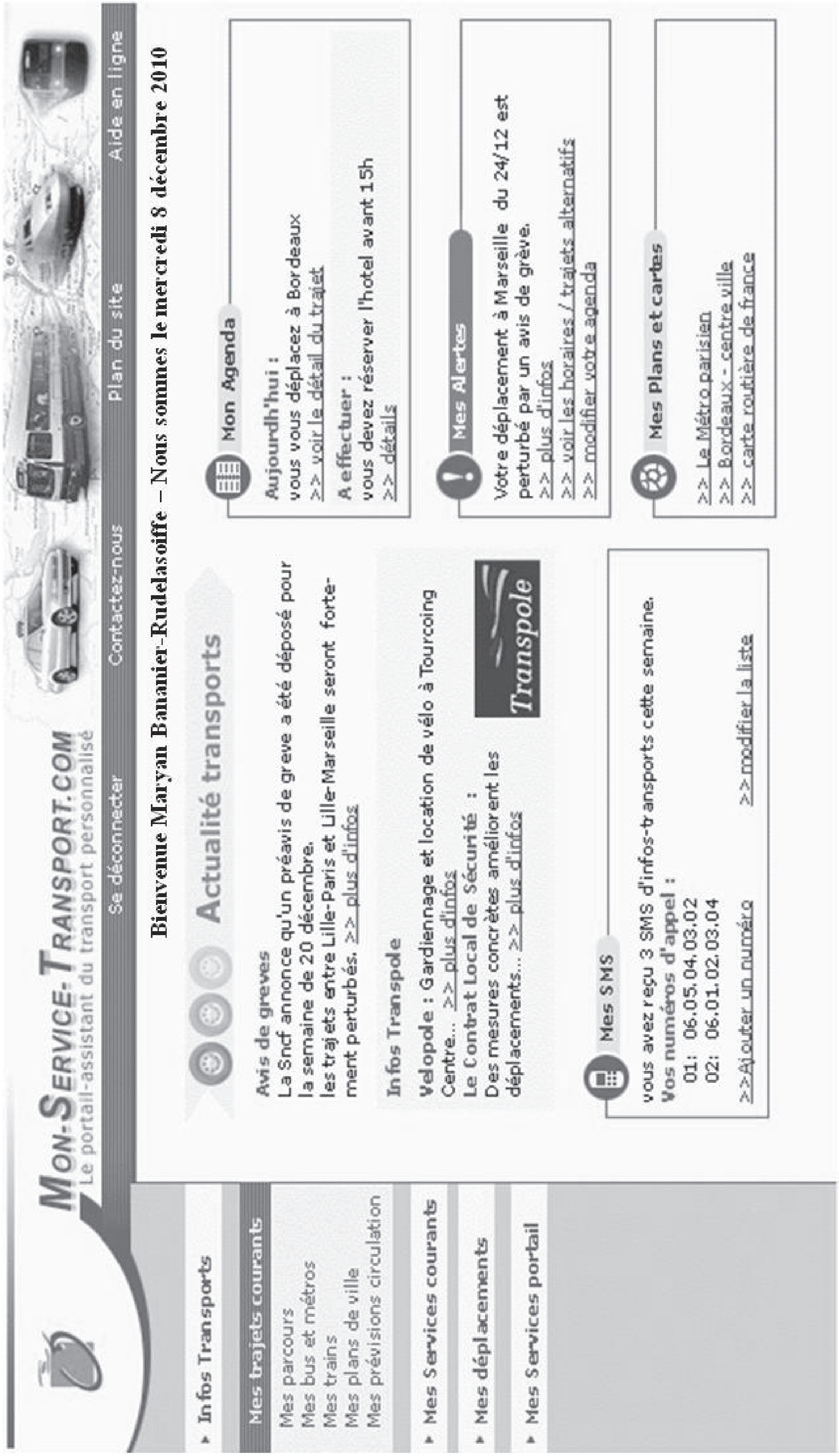

Figure 3.27. *Illustration of a personalized Web page*

Figure 3.27 can provide us with an insight into this matter. It takes the elements of a demonstrator developed in the context of the PREDIT AGENPERSO project (human-machine interfaces based on PERSOnnal software AGENts of information to the collective transport users) [PET 03b], bringing together LAMIH, INRETS and Archimed. The personalized Web page, visible in this figure, is exclusively meant for an identified user. The system knows via the intermediary of his schedule that he must go to Bordeaux. The agents put themselves at the service of the user by recuperating information that is relevant to him: the need to reserve the hotel before 3pm, collection of plans likely to be useful in the context of this trip, etc. The agents also anticipate future trips. For example, several days later the user must go to Marseille, but the intelligent agents detect that there is advanced notice of a strike that could disrupt the trip; the user perhaps needs to consider a re-planning the journey and is warned via the intermediary of the personalized system.

By generalization, other elements of personalized information can be expressed with mobility information, via knowledge of the preferences of the user in terms of leisure, for example.

Intelligent agents can indeed go in search of information likely to complement the user's trip by informing themselves as to the possibilities of shows and events at the destination place, as well as about the *best* restaurants (best in the sense of the adaptation to the user's criteria), the most *interesting* museums (for example, "the museum of pans, dishes and other kitchen utensils" if the user enjoys cooking), all the while preparing the *best* way (according to the user's criteria) to access it. Agents can obtain information about television programs if the user is too tired to leave the hotel in the evening (the study taking into account the hundreds of channels that might be available). Numerous other generalization ideas can of course be envisaged, which opens up new research avenues.

3.7. Conclusion

This chapter has described a contribution to the personalization of ISs, in view of improving the HMI and moving towards adaptive, intelligent HMIs, within a huge international research movement. We have put forward a method called PerMet (*PERsonalization METhodology*) for the development of PISs. This method enables both the implementation of a new PIS as well as the personalization of an already existing IS. PerMet proposes an iterative and incremental development model and allows the specific phases linked to the development of services and the specific phases linked to personalization to be carried out in parallel. We have also put forward PerSyst (PERsonalization SYSTem), which is a PS that supports the PerMet method, consisting of agents at the service of users. PerMet and PerSyst were validated in different applications based on real or simulated data for the personalization of terrestrial transport information for people. One of these applications was presented in this chapter.

The research perspectives are numerous. It would be possible to improve both the PerMet method as well as the PerSyst system. As explained in the discussion (section 3.6), other perspectives of these works would consist in using PerMet for the personalization of information connected to transport (hotels, restaurants, museums, etc). It would also be interesting to work on new machine learning methods for the personalization of transport information. In the end this should contribute to new, friendlier and more personalized interactive services.

3.8. Bibliography

[ABO 03] ABOU-HARB G., RIVARD F., *L'EAI au Service de l'Entreprise Évolutive*, Maxima, Paris, France, 2003.

[ANL 05a] ANLI A., GRISLIN-LE STRUGEON E., ABED M., "A Generic Personalization Tool based on a Multi-agent Architecture", in: STEPHANIDIS C. (ed.), *Proceedings HCI International (Las Vegas, Nevada, July 22-27, 2005), Volume 7 - Universal Access in HCI: Exploring New Interaction Environments,* Lawrence Erlbaum Associates, Mahwah, USA, pp. 1-8, July 2005.

[ANL 05b] ANLI A., GRISLIN-LE STRUGEON E., ABED M., "Une plate-forme de personnalisation basée sur une architecture multi-agents", *Revue des Nouvelles Technologies de l'Information (RNTI-E)*, vol. 5, pp. 95-100, 2005.

[ANL 05c] ANLI A., KOLSKI C., ABED M., "Principes et architecture pour la personnalisation d'information en interaction homme-machine, application à l'information transport", *Proceedings of IHM 2005*, International Conference Proceedings Series, pp. 123-130, ACM Press, Toulouse, France, September 2005.

[ANL 06a] ANLI A., Méthodologie de développement des systèmes d'information personnalisés - Application à un système d'information au service des usagers des transports terrestres de personnes, PhD thesis, University of Valenciennes and Hainaut-Cambrésis, Valenciennes, France, October 2006.

[ANL 06b] ANLI A., ABED M., "PerSyst: Un système de personnalisation de l'information transport multimodale", *Sciences et Technologies pour l'Automatique*, vol. 3, pp. 1-6, 2006.

[ANL 07] ANLI A., ABED M., "PerMet, méthode pour le développement de systèmes interactifs personnalisés: application au transport collectif de personnes", in: I. SALEH I., REGOTTAZ D. (eds), *Interfaces Numériques*, Hermès, Paris, France, pp. 71-89, 2007.

[ASN 97] ASNICAR F.A., TASSO C., "IfWeb: a prototype of user model-based intelligent Agent for document filtering and navigation in the World Wide Web", *Proceedings of the Workshop "Adaptive Systems and User Modeling on the World Wide Web*", pp. 3-12, Chia Laguna, Italy, June 1997.

[BEL 07] BELLIFEMINE F.B., CAIRE G., GREENWOOD D., *Developing Multi-Agent Systems with JADE*, Wiley Series in Agent Technology, 2007.

[CAL 03] CALVARY G., COUTAZ J., THEVENIN D., LIMBOURG Q., BOUILLON L., VANDERDONCKT J., "A unifying reference framework for multi-target user interfaces", *Interacting with Computers*, vol. 15, no. 3, pp. 289-308, 2003.

[CAL 07] CALVARY G., *Plasticité des Interfaces Homme-machine*, Mémoire d'Habilitation à Diriger des Recherches, University of Grenoble, France, November 2007.

[CHA 99] CHANDRASEKARAN B., JOSEPHSON J.R., BENJAMINS V.R., "What are ontologies, and why do we need them?", *IEEE Intelligent Systems*, vol. 14, no. 1, pp. 20-26, 1999.

[CHE 00] CHELLA A., CONSSENTINO M., LO FASO U., "Designing agent-based systems with UML", in: *International symposium on Robotics and Automation, ISRA'2000*, Monterrey, Mexico, November 2000.

[CON 00] CONALLEN J., *Concevoir des Applications Web avec UML*, Eyrolles, Paris, France, 2000.

[DIC 03] DICKINSON I., REYNOLDS D., BANKS D., CAYZER S., VORA P., "User Profiling with privacy: A framework for Adaptive Information Agents", in: KLUSH M. *et al.* (eds), *Intelligent Information Agents,* LNAI 2586, pp. 123-151, Springer-Verlag, Berlin, Germany, 2003.

[GRI 01] GRISLIN-LE STRUGEON E., ADAM E., KOLSKI C., "Agents intelligents en interaction homme-machine dans les systèmes d'information", in: KOLSKI C. (ed.), *Environnements Évolués et Évaluation de l'IHM. Interaction Homme-machine pour les SI*, vol. 2, pp. 209-248, Hermès, France, Paris, 2001.

[GRI 07] GRISLIN-LE STRUGEON E., ANLI A., PETIT-ROZE C., SOUI M., CONREUR G., DOS SANTOS P., ABED M., KOLSKI C., MOUVER.PERSO, MObilité et mUltimodalité Voyageurs Etudiants, Système d'Information Multimodale Personnalisée, Intermediate report (May 2006) and Final report (July 2007) of Research Contract PREDIT DTT 216-75-01, LAMIH, Valenciennes, France, 2007.

[HAB 04] HABIEB-MAMMAR H., TARPIN-BERNARD F., PREVOT P., "Adaptive presentation of multimedia interface case study: Brain story course", *Lecture Notes in Computer Science*, vol. 2702/2003, pp. 15-24, 2004.

[HAY 04] HAYES C., CUNNINGHAM P., "Context boosting collaborative recommendations", *Knowledge-Based Systems*, no. 17, pp. 131-138, 2004.

[HOO 00] HÖÖK K., "Steps to take before intelligent user interfaces become real", *Interacting with Computers*, vol. 12, pp. 409-426, 2000.

[HUA 08] HUART J., KOLSKI C., BASTIEN C., "L'évaluation de documents multimédias, état de l'art", in: MERVIEL S. (ed.), *Objectiver l'Humain? Volume 1, Qualification, Quantification*, pp. 211-250, Hermès, Paris, France, 2008.

[JAC 99] JACOBSON I., BOOCH G., RUMBOUGH J., *The Unified Software Development Process*, Addison Wesley Longman, UK, 1999.

[KEE 00] KEEBLE R.J., MACREDIE R.D., "Assistant agents for the world wide web intelligent interface design challenges", *Interacting with Computers*, vol. 12, no. 4, pp. 357-381, 2000.

[KOB 95] KOBSA A., POHL W., "The user modeling shell system BGP-MS", *User Modeling and User-Adapted Interaction,* vol. 4, pp. 59-106, 1995.

[KOL 92] KOLSKI C., TENDJAOUI M., MILLOT P., "A process method for the design of "intelligent" man-machine interfaces: case study: "the decisional module of imagery"", *International Journal of Human Factors in Manufacturing*, vol. 2, pp. 155-175, 1992.

[KOL 98] KOLSKI C., GRISLIN-LE STRUGEON E., "A review of intelligent human-machine interfaces in the light of the ARCH model", *International Journal of Human-Computer Interaction*, vol. 10, pp. 193-231, 1998.

[KOL 01] KOLSKI C. (ed), *Analyse et Conception de l'IHM*, Hermès, Paris, France, 2001.

[KOL 04] KOLSKI C., PETIT-ROZE C., ANLI A., ABED M., GRISLIN-LE STRUGEON E., EZZEDINE H., TRABELSI A., "La plasticité vue sous l'angle de la personnalisation ou selon les besoins vis-à-vis de l'information transport", *Journées thématiques de l'AS Plasticité du RTP 16 IHM*, Namur, Belgium, August 2004.

[LIE 01] LIEBERMAN H., FRY C., WEITZMAN L., "Exploring the web with reconnaissance agents". *ACM Conference on Human-Computer Interface*, ACM Press, pp. 69-75, August 2001.

[LIM 05] LIMBOURG Q, VANDERDONCKT J., MICHOTTE B., BOUILLON L., LOPEZ-JAQUERO V., "USIXML: A language supporting multi-path development of user interfaces", in: BASTIDE R., PALANQUE P., ROTH J. (eds), *Engineering Human Computer Interaction and Interactive Systems, Joint Working Conferences EHCI-DSVIS 2004*, pp. 200-220, Hamburg, Germany, July 11-13 2005.

[MAN 01] MANOUVRIER B., *EAI: Intégration des Applications d'Entreprise*, Hermès, Paris, France, 2001.

[MAN 02] MANDIAU R., GRISLIN-LE STRUGEON E., PENINOU A. (eds), *Organisation et Applications des SMA*, Hermès, Paris, France, 2002.

[MEL 10] MELKI A., HAMMADI S., SALLEZ Y., BERGER T., TAHON C., "Advanced approach for the public transportation regulation system based on cybercars", *RAIRO-Operations Research*, vol. 44, no. 1, pp. 85-105, 2010.

[MON 04] MONFORT V., GOUDEAU S., *Web services et Interopérabilité des SI: WS-I, WSAD/J2EE, Visual Studio.Net et BizTalk*, Dunod, Paris, France, 2004.

[MOU 09] MOURLAS C., GERMANAKOS P., *Intelligent User Interfaces: Adaptation and Personalization Systems and Technologie*s, IGI Global, Hershey, USA, 2009.

[NAN 01] NANCI D., ESPINASSE B., *Ingénierie des Systèmes d'Information: MERISE, 2ème Génération*, Vuibert, Paris, France, 2001.

[NEW 02] NEWCOMER E., *Understanding Web Services: XML, WSDL, SOAP, and UDDI*, Addison-Wesley Professional, Boston, USA, 2002.

[NIE 93] NIELSEN J., *Usability Engineering*, Academic Press, Boston, USA, 1993.

[NOD 00] NODINE M., FOWLER J., KSIEZYK T., PERRY B., TAYLOR M., UNRUH A., “Active information gathering in Info Sleuth”, *International Journal of Cooperative Information Systems*, vol. 9, no. 1/2, pp.3-28, 2000.

[ODE 99] ODELL J., VAN DYKE PARUNAK H., BOCK C., Representing agent interaction protocols in UML, OMG Document/ad99-12-01, Intellicorp Inc, December 1999.

[OHA 03] O’HARE G.M.P., O’GRADY M.J., “Gulliver’s Genie: a multi-agent system for ubiquitous and intelligent content delivery”, *Computer Communications*, vol. 26, pp. 1177-1187, 2003.

[OPP 04] OPPLIGER R., Microsoft.NET passport and identity management, Information Security Technical Report, vol. 9, no. 1, pp. 26-34, January-March 2004.

[PET 01] PETIT-ROZÉ C., GRISLIN-LE STRUGEON E., “Interaction with agent systems for intermodality dans transport systems”, in: M.J. SMITH, G. SALVENDY, D. HARRIS, R. KOUBEK (eds), *Usability evaluation and Interface design: Cognitive Engineering, Intelligent Agents and Virtual Reality*, vol. 1, Lawrence Erlbaum Associate Publishers, London, pp. 494-498, 2001.

[PET 03a] PETIT-ROZE C., Organisation multi-agents au service de la personnalisation de l’information, application à un système d’information multimodale pour le transport terrestre de personnes, PhD thesis, University of Valenciennes and Hainaut-Cambrésis, Valenciennes, France, January 2003.

[PET 03b] PETIT-ROZE C., ANLI A., GRISLIN-LE STRUGEON E., ABED M., KOLSKI C., AGENPERSO: Interface Homme-machine à Base d’AGENts Logiciels PERSOnnels d’Information aux Usagers des TC, Final report of project PREDIT, LAMIH, Valenciennes, France, January 2003.

[PET 04] PETIT-ROZE C., ANLI A., GRISLIN-LE STRUGEON E., ABED M., USTER G., KOLSki C., “Système d’information transport personnalisée à base d’agents logiciels”, *Génie Logiciel*, 70, pp. 29-38, 2004.

[PET 06] PETIT-ROZÉ C., GRISLIN-LE STRUGEON E., “MAPIS, a multi-agent system for information personalization”, *Information and Software Technology*, vol. 48, pp. 107-120, 2006.

[PHO 99] PHOL W., “Logic-based representation and reasoning for user modeling shell systems”, *User Modeling and User-Adapted Interaction*, vol. 9, no. 3, pp. 217-282, 1999.

[PFI 04] PFITZMANN B., Privacy in enterprise identity federation – policies for Liberty 2 single sign on, Information Security Technical Report, vol. 9, no. 1, pp. 45-58, January-March 2004.

[PUE 02] PUERTA A., EISENTEIN J., *XIML*: A Universal Language for User Interfaces, White paper, 2002.

[RAO 91] RAO A.S., GEORGEFF M.P. “Modeling rational agents within a BDI architecture”, *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning (KR’91)*, pp. 473-484, 1991.

[ROQ 07] ROQUES P., VALLEE F., *UML2 en Action*, 4th edition, Eyrolles, Paris, France, 2007.

[ROU 04] ROUILLARD J., *VoiceXML, Le Langage d'Accès à Internet par Téléphone*, Vuibert, Paris, France, 2004.

[SEA 08] SEARS A., JACKO J.A., *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies*, 2nd edition, CRC Press, Washington, USA, 2008.

[SER 08] SERRATE O., BERGER T., TABARY D., KOLSKI C., "Towards context-aware interactions between humans and a self-organized cybercar system", *27th European Annual Conference on Human Decision-Making and Manual Control, EAM'08*, Delft, The Netherlands, June 2008.

[SOU 07] SOUI M., ABED M., KOLSKI C., GHÉDIRA K., "Criteria devoted to evaluate personalized interactive systems", in: M. ABID, A. HADJ KACEM, M. JMAIEL, M. LAHIANI (eds), *Nouvelles tendances technologiques en génie électrique & informatique, GEI'2007*, Edition CPU, Sfax, Tunisia, pp. 119-125, 2007.

[SOU 08a] SOUI M., ABED M., "Evaluation of personalized interactive systems in transport applications: criteria and evaluation methods", *27th European Annual Conference on Human Decision-Making and Manual Control, EAM'08*, Delft, The Netherlands, June 2008.

[SOU 08b] SOUI M., KOLSKI C., ABED M., USTER G., "Evaluation of personalized information systems: application in intelligent transport system", *Proceedings of the 20th International Conference on Software Engineering and Knowledge Engineering, SEKE2008*, pp. 877-880, San Francisco, USA, July 2008.

[SOU 10] SOUI M., Contribution à l'évaluation des systèmes d'information personnalisés, Application au transport collectif de personnes, PhD thesis, University of Valenciennes and Hainaut-Cambrésis, Valenciennes, France, February 2010.

[STA 05] STANCIULESCU A., LIMBOURG Q., VANDERDONCKT J., MICHOTTE B., MONTERO F., "A transformational approach for multimodal web user interfaces based on UsiXML", in: LAZZARI G., PIANESI F., CROWLEY J.L., MASE K. and OVIATT S.L. (eds), *Proceedings of the 7th International Conference on Multimodal Interfaces*, ICMI 2005, pp. 259-266, Trento, Italy, October 4-6, 2005.

[SU 09] SU X., KHOSHGOFTAAR T.M., "A survey of collaborative filtering techniques", *Advances in Artificial Intelligence*, vol. 19, Article ID 421425, 2009.

[TRO 99] TROUSSE B., JACZYNSKI M., KANAWATI R., "Using user behavior similarity for recommandation computation: The broadway approach", in: *Proceedings of the 8th International Conference on Human Computer Interaction (HCI'99)*, BULLINGER H.J. and ZIEGLER J. (eds), pp. 85-101, Munich, Germany, 1999.

[VIR 02] VIRVOU M., KABASSI K., "IFM: An intelligent graphical user interface offering advice", *2nd Hellenic Conf. on AI, SETN-2002,* Thessaloniki, Greece, pp. 155-164, April 2002.