Illustrating some issues raised when designing context-aware personalized services for mobile users

Pathathai Na Lumpoon, Mu Lei, Teerawat Kamnardsiri, Ahmed Lbath, Marie-Christine Fauvet

Illustrating some issues raised when designing context-aware personalized services for mobile users

Pathathai Na Lumpoon, Mu Lei, Teerawat Kamnardsiri, Ahmed Lbath, Marie-Christine Fauvet
Joseph Fourier University of Grenoble, LIG-MRIM,
41 Rue des Mathématiques BP53X - 38041 Grenoble Cedex 9 France
Emails: Pathathai.Na-Lumpoon@imag.fr, Lei.Mou@imag.fr, Teerawat.Kamnardsiri@imag.fr
Ahmed.Lbath@imag.fr, Marie-Christine.Fauvet@imag.fr

***Abstract*—When travelling people might seek for help or any sorts of information. For example, travelers might need suggestions about accommodation, transportation, activities, food, etc. while they are travelling. Moreover, they expect to get suggestions which are personalized according to some specific criteria such as preferences, age, location, etc. This paper sketches a framework named Context-Aware Recommender for Mobile Users that is responsible for providing users with personalized recommendations in order to deliver them the right service to the right user at the right time with the respect of their privacy.**

Index Terms **Mobile Computing, Recommendation Systems, Context Aware and Personalized Services, Mobile Tourism, Users' Privacy.**

I. Introduction and motivating example

With the advent of pervasive computing, several research fields and directions have emerged in computer science. One of them is about context-aware and personalized applications or softwares that are meant to fulfill mobile users' needs. These applications and softwares are challenged to locate and deliver the right service to the right person, at the right time and location, with the appropriate rendering. We adopt the definition of service given by the W3C [1]: *A service is an abstract resource that represents a capability of performing tasks that form a coherent functionality from the point of view of providers entities and requesters entities. To be used, a service must be realized by a program acting on behalf of a person or organization (i.e. the provider).*

The study reported in this paper is a part of a broader project which aims at designing and implementing a framework which provides context-aware personalized services for mobile users. To achieve such a goal, issues to be addressed are related to system design, software architecture, distributed and heterogeneous resource access and integration, quality of service requirements, and information synchronization when switching between connected and disconnected mode.

As an illustrating example, we consider the following scenario: *Peter is traveling in Chiang Mai Province, Thailand. He is standing near by the monument of Three Kings. Using his smart-phone and logged on our recommendation system, he enters the query: I would like to know whether there are any places of interest located around here. I am interested also in booking a 2 day-tour starting tomorrow*. The result returned to Peter is: i) a list of points of interests (museums within Three King monument area) and ii) a two day-tour booking in the bush around Chiang Mai City, resulting from some interactions with the appropriate tour operator company.

Processing Peter's query involves the selection, integration and orchestration of web services that meet the most Peter's needs according to his context and profile. Here we consider the following definitions for context, profile and web service concepts. A *context* includes spatial, temporal, physical and environmental properties that could be collected by sensors embedded in the devices used to submit the queries. Such properties are for example: GPS location, timestamp, external temperature, screen size, etc. A *profile* captures users' personal details, preferences and centers of interest. For instance, a profile could contain information about users' age, citizenship, gender, occupation, favorite recreational activities, etc.

Back to our motivating example, the context of Peter's query is [city: Chiang Mai, location: the monument of Three King, coordinates: 18°50'14"N 98°58'14"E, device: a smart phone, time: June 25th 2012 12pm, temperature: 38C]. Peter's profile is [gender: male, citizenship: UK, age: unknown, favorite activities: hiking, museum]. According to the respect of Peter's privacy, his age is unknown because he doesn't want to disclose it.

In this paper, we consider the concept of *web service* as defined by the W3C [1]: *A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL[1]). Other systems interact with the Web service in a manner prescribed by its description using SOAP[2] messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.*

The remainder of this paper is organized as follows. First we give an overview of the system functionalities and architecture in Section II. Next, in Section III, we introduce our approach to recommendation. Section IV discusses service integration and orchestration. Finally, section V gives a brief overview of related work and provides some concluding remarks.

II. Architecture

As shown in Figure 1, the system we propose is composed of two main components: (1) MS-S Mobile Server which is

[1]Web Service Description Language
[2]Simple Object Access Protocol

responsible for accepting requests from users, processing requests (e.g. finding a composition of services that fits the most the request), and binding the resulting composition with the requester; (2) M-App an application running on users' mobile device which is responsible for collecting and maintaining information users accept to store and disclose to MS-S Mobile Server.

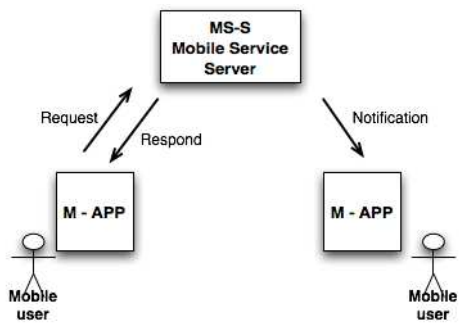

Fig. 1. Push and pull interaction modes between users and MS-S Server

We assume that users' mobile devices are capable of running a browser for accessing MS-S Server and a mailer for receiving notifications. Using the browser users can access the MS-S Mobile server in order to submit requests. This communication mode between users and the server is called "pull mode". The server can also contact users by proposing services that might be useful. This latter communication mode is called "push mode". Figure 2 illustrates the architecture we propose for our context aware recommendation system. Components are described below.

The component Portal is a web application providing the needed functionality to support the proper method to interact with users. The Portal builds the query Q (the requested query from users), and analyses Q with User and Context Management component to generate Q' (the detailed query with user information).

The Recommender is a matching component which offers the prediction techniques to provide a list of recommended services discovered in Service Description matched on users' profile and context. However, the system keeps recommendation patterns to be used for machine learning in Recommendation Management component. Consequently, the actual output of Recommender is a list of service, service control flow and recipient noted as R in the architecture.

The Binder is a component which uses semantic technique to R to generate a list of service orchestration models noted as O. O is a group of codes later sent to M-App to be executed.

The recommendation management component handles a set of recommendation and orchestration models. Service Description handles a set of business services such as hotel, restaurant, shopping and etc.

As already said, there are two delivery service approaches which are pull and push services. The workflow of pull and push services is displayed in the architecture. To achieve the first one, mobile user submits query Q through his browser to

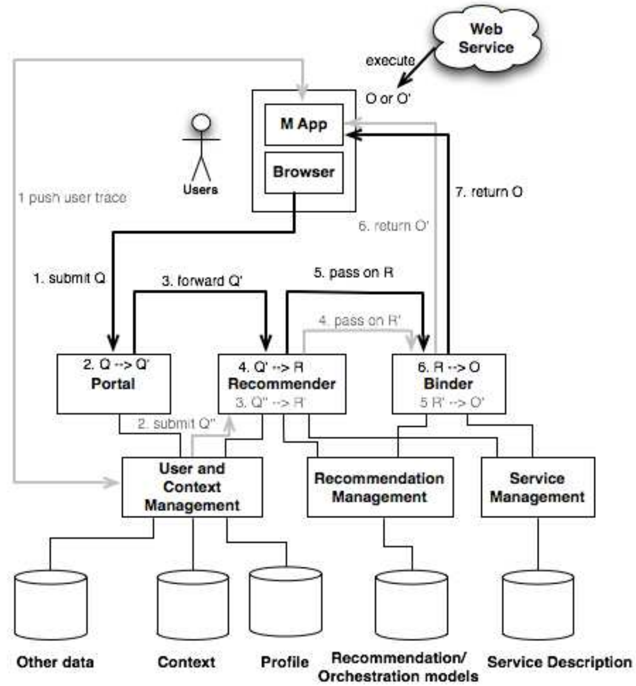

Fig. 2. Context aware recommendation system logical architecture

Portal web application. The Portal generates Q' to be input of Recommender. Recommender produces R and pass R to Binder. Later the output O generated from Binder is delivered to Mobile user. The later approach works similar to the first one that deliver right services to users except the users do not send any request query. The recommender trigger resides in User and Context Management component monitoring user context change, notifying the change and sending the predicted query Q" to Recommender component.

The Recommender processes Q" to conduct a list of services R' used to deliver to mobile users later on. The more information regarding to Recommender is discussed in the next section.

III. Recommendation

The component recommender in the architecture we proposed in this paper works as a matching component, with the help of the component Service Management, it's responsible for searching web services to answer queries submitted by users, their context and users' profile.

In the recommender, we aim to provide recommendation selected from existing Service Descriptions (denoted by S), according to the standardized user's query enriched with the context and user's profile (denoted Q'), the group profile (noted P) and group context (noted C) from the cluster which the user belongs to. The output of this component (noted R) contains Q' and a list of services. Finally, R is sent to the component "Binder" which is, in turn, 0responsible for integrating the returned services and orchestrating them (see Section IV).

The process of generating R from Q' is carried out respecting users' privacy.

In order to predict the user's potential interests properly, the components Profile Management and Context Management classify or cluster profiles and contexts stored in the database, dividing them into different groups by calculating the similarity. The recommender learns from Q' the user's profile and context, finds the group which is the most similar to the user, picks up the group profile (P) and group context (C). Q'+P+C works as a filter, to find the highly customized recommendation of services in the Services Description, by Services Description Management.

Back to our motivating example, Peter is after places of his interests in the area of the Three King Monument as well as a 2 day-tour package starting June 26. Among all web services whose descriptions are managed by the component Service Management (see Figure 2), the recommender has selected 2 web services: (i) http://www.chiangmaitourism.org which exposes, among others, one operation that returns a list of points of interests located in a given area in Chiang Mai city; (ii) http://www.adventure.tour.th which exposes 2 operations: the first one returns a list of packages offered for a given duration and an area around Chiang Mai and the second one for booking a selected package, charged on a credit card given by its number.

While users are interacting with the system, their privacy has to be considered. Privacy may has different meanings to different users. For example, Peter does not want the others to know his current location, while Mary may consider that this is not an issue. Privacy requirements, specific to each class of users, have to be captured and stored by the system.

Also, user's private information may be leaking during the processes of information collection, storage, and clustering/classification.

Conducting the storage and packaging of user profiles on the client side, and developing the privacy sensitive algorithm for users' profile clustering or classification to prevent users' privacy from disclosing will be explored in future work.

IV. Integration and orchestration

This section intends at introducing issues raised by integration, composition and execution of heterogeneous services that have been recommended. The services we consider could be either stateless or stateful and might offer more than only one operation. Among others, these issues are:

- Specification of an orchestration model according to the output of the recommender (e.g. descriptions of the selected services).

- Syntactic and semantic integration of these services.

The component Binder generates an orchestration model that specifies both control flow and data flow such as the services returned by the Recommender can be executed. The control flow is captured by an oriented graph whose nodes are web service operations and oriented edges from one node O1 to another O2 means that the operation associated with O2 has to be executed before the one associated with O1.

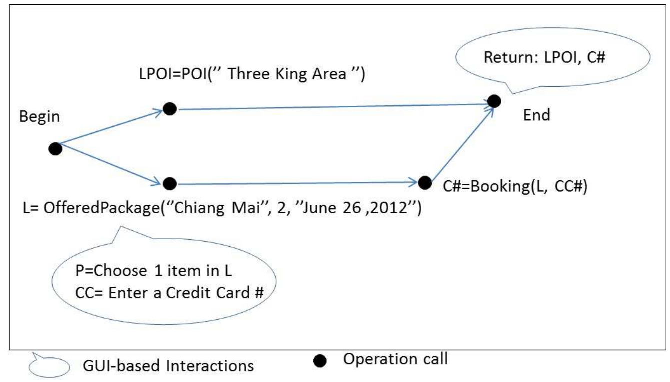

Fig. 3. Orchestration model with user interactions

Figure 3 illustrates the orchestration model related to our example, as it has to be generated by the component Binder. This orchestration model is completed by some interactions with users. First, the operation POI provided by the service http://www.chiangmaitourism.org is executed with the parameter Three King Area; it returns a list of POIs. In parallel the operation OfferedPackage provided by http://www.adventure.tour.th is executed with parameters Chiang Mai city, 2 (e.g. 2 days), and June 26, 2012. It returns a list of 2 day-tour packages starting June 26, 2012 and around Chiang Mai city. In this orchestration model, some parameters such as Chiang Mai city, June 26, 2012 come from the context.

The choice of the web service http://www.adventure.tour.th is driven by Peter's profile (Peter likes hiking and adventure). The duration and the choice of a list of POIs are deduced from the query. All others have to be given by the user via some interactions (see callouts in Figure 3). These later interactions are generated by the component Binder according to the descriptions of the participating services.

V. Conclusion

Many of previous works are related to context recommendation services applied in travel and tourism domain. Murshid provides guidance using context model of user location, user profile, date, event information and user interaction to user visiting a country for tourism or business [2]. LiveCities similar to Murshid focused on the push service notification to mobile device according to the visitors' context on a defined area [3]. However, [2] and [3] are designed specifically for mobile service model in tourism domain. AIDAS is a middleware to support user-centric semantic service discovery based on user/device/service profile (meta data), which uses Semantic-based discovery to support mobile context-aware service access [4]. Nevertheless, AIDAS middleware does not support automatic workflow generation. The last related work [5] is the closet to our work. The system has workflow process using semantic matching with evaluation of QoS and syntactic mapping to satisfy user query. However, [5] does not provide push service mode and focus less on user profile.

In this paper we have presented a framework dedicated to the design and development of Context-Aware Recommender for Mobile Users that aims to deliver the right service to the right user at the right time with respect of the users privacy.

Currently, we are working on the definition of the formal model supporting the framework including standardized architecture with a recommender system that takes into account the respect of users privacy. For experimental validation purpose, we plan to implement the framework presented here, and apply it to mobile tourism application domain.

ACKNOWLEDGMENT

This presented work is supported by European Commission in Sustainable e-Tourism: Erasmus Mundus Action 2 Program.

REFERENCES

[1] W3C, "Web services glossary," http://www.w3c.org/TR/ws-glossary.

[2] A. Echtibi, M.-J. Zemerly, and J. Berri, "A servicebased mobile tourist advisor," *International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM)*, vol. 1, pp. 177–187, 2009.

[3] D. Martin, A. Alzua, and C. Lamsfus, "A contextual geofencing mobile tourism service," in *Proceedings of the International Conference Information and Communication Technologies in Tourism*, Innsbruck, Austria, 2011.

[4] A. Toninelli, A. Corradi, and R. Montanari, "Semantic-based discovery to support mobile contextaware service access," *Computer Communications*, vol. 31, no. 5, pp. 935–949, 25 March 2008.

[5] A. Bhuvaneswari and G. Karapagam, "Reengineering semantic web service composition in a mobile environment," in *Proceedings of the International Conference on Recent Trends in Information, Telecommunication and Computing*, 2010.