Offering Context-Aware Personalised Services for Mobile Users

Marie-Christine Fauvet[1](✉), Sanjay Kamath[1],
Isaac-Bernardo Caicedo-Castro[2], Pathathai Na-Lumpoon[3],
Ahmed Lbath[1], and Lorraine Goeuriot[1]

[1] LIG (MRIM), University of Grenoble Alpes, 38000 Grenoble, France
Marie-Christine.Fauvet@imag.fr
[2] University of Córdoba, Córdoba, Colombia
[3] University of Chang-Mai, Chiang Mai, Thailand

1 Introduction

This paper presents a system that provides mobile users, context-aware personalized services. Users might need any sort of services: information about the weather, places to visit, accommodation booking, etc. The eTourism system is based on a semantic composition of recommended services into a composite service. A typical use case is as following:

Alice is an American tourist visiting Paris in France. At 4 PM, she wants to book a table at the finest restaurant in the city, and the direction to get there. So, she picks up her smartphone and accesses the system whose architecture is discussed in the next section (see Sect. 2), and issues the query: *I want to book a table for 2 people at the finest restaurant in the city, and I need the directions to the restaurant.*

The eTourism system will assist Alice by (1) analyzing her query; (2) capturing her context information; (3) processing the query, by identifying the right services, and composing them. (4) enacting the resulting composite services thus fulfilling Alice's needs.

Our main contribution is threefold: first, we have addressed service retrieval issues in order to build a module which discovers services queried by users; then, we have dealt with issues related with service operation automated composition to execute business processes, and finally, our system supports interactions with users to discover the missing parameters (not included in users' queries or their context information) in order to execute the business process. The expected result of the executable business process is able to fulfill the user's needs.

The demonstration introduced in this paper is accessible via the link http://lig-membres.imag.fr/etourism/.

2 System Architecture

The system we describe in this paper provides mobile users with context-aware personalized services according to their needs [5] when they are travelling. Figure 1

A. Barros et al. (Eds.): ICSOC 2015, LNCS 9435, pp. 495–498, 2015.
DOI: 10.1007/978-3-662-48616-0_37

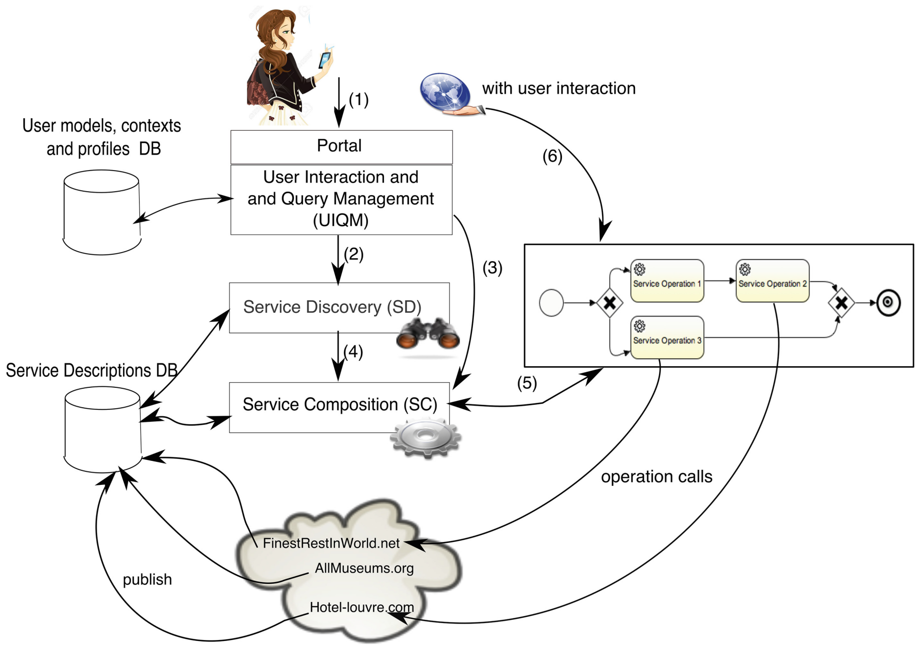

Fig. 1. A context-aware discovering system for mobile users

sketches the overall architecture of this system. The role of each module and the flow of information are detailed further in this section.

We consider the following definitions for context, profile and service concepts. A *context* includes spatial, temporal, physical, and environmental properties that could be collected by sensors embedded on 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 centres of interest.

We adopt the definition of service given by the W3C [6]: *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 system whose architecture is sketched in Fig. 1, is accessible to registered users via a website on their mobile device (see Fig. 1, the data flow 1). This system is built on top of three components, each of which has been implemented as a service (the Portal is a web application).

[1] Web Service Description Language, http://www.w3.org/TR/wsdl.

[2] Simple Object Access Protocol.

User Interaction and Query Management (UIQM). This module aims at managing user connections and handling queries submitted by users. From data flow 1, this module receives the query and identification, and extracts from it the information needed to choose and compose service components. With data flow 2 the module sends queries to the Service Discovery module: if the query contains multiple requirements, it is split according to individual requirements and sent as single topic queries. With data flow 3, the users' requirements are sent to the Service Composition module. For more detailed description see [2].

Service Discovery (SD). Given the user's query received in the data flow 2, her profile and context, this module is responsible for retrieving services among a repository of service descriptions, that once composed can potentially meet the user's needs expressed in her query (for details see [1]). The retrieved services are then sent, following the data flow 4, to the next module.

Service Composition (SC). Eventually this module is in charge of automatically composing and executing services returned by the discovery phase. This module results in a BPMN[3] model (see data flow 5) whose tasks refer to a service operation (for details see [3,4]). During the execution of the BPMN model some interactions with the user might be necessary to give value to parameters not found in the original query (see data flow 6).

3 Use Case

In order to illustrate the use case presented in Sect. 1, we will detail the process on the following query: *I want to book a table for 2 people at the finest restaurant in the city, and I need the directions to the restaurant.*

The User Interaction and Query Management (UIQM) module analyses the query to retrieve Alice's context and profile, along with the parameters needed for the requirements of the query.

The UIQM first captures Alice's context information: coordinates="48.2167 N, 2.3332 E", current date="1/06/2014". Besides, the system has the following information about Alice's profile: name="Alice", citizenship="USA", travelPurpose="tourism", gender="female".

The query considered in the use case has two requirements (restaurant booking, and directions), then the UIQM module splits this query into two subqueries and submits them to the SD. The first subquery is: *I want to book a table at the finest restaurant in the city.* The second subquery is *I need the directions to the restaurant.* The SD sends to the SC two ranked lists of candidate services, one for each subquery. From the list of candidate services that may fulfill the first subquery, the one with the highest rank contains the following operations:

1. FindFinestRestaurant: cityName ⟶ name, address which receives as a parameter the name of the city where Alice is looking for the finest restaurant. This operation returns the name and the address of the restaurant.
2. BookRestaurant: name, people, phone ⟶ bookingConfirmation which receives as parameters the restaurant name, the number of persons, and the user's telephone number. As a result, the operation returns a confirmation whether the table has been booked or not.

[3] Business Process Management Notation (http://www.bpmn.org).

In the ranked list of candidate services that fulfill the second subquery, the highest ranked service contains the following operations:

1. FromCoordinatesToCity: longitude, latitude ⟶ cityName which, given geographical coordinates, returns the name of the city whose location includes these coordinates.
2. CoordinatesFromAddress: address ⟶ longitude, latitude which returns the geographical coordinates of the giver address.
3. GetDirection: origin, destination ⟶ direction which provides instructions on how to reach a destination. This operation receives two parameters, the coordinates of the starting point, and the coordinates of the destination.

The module SC takes as an input the operations of both services and compose them. The resulting composite is transformed into a BPMN model which in turn is executed on top of BPMN engine taken from the shelf. The resulting composite service fulfills both Alice's needs (i.e., booking a table at the finest restaurant of the city, and getting the directions to go there).

4 Conclusion and Further Work

In this work, we contribute with a context-aware system which provides mobile users with services, which are requested through queries in natural language.

For further work: (1) we will address service recommendation issues, aiming at provide services in push-mode, (2) we will tackle issues related with user privacy protection, (3) we will handle stateful services in business process model and (4) we will improve the user interaction by including into the system speech recognition to support spoken queries.

References

1. Caicedo-Castro, I.-B.: S^{3}niffer: A text description-based service search system. Ph.D. Dissertation, University of Grenoble (2015)
2. Kamath, S.: Discovering context information and parameters from a natural language query. Masters MoSIG (2015).https://hal.archives-ouvertes.fr/hal-01172050
3. Na-Lumpoon, P.: Toward a framework for automated service composition and execution: E_tourism applications. Ph.D. Dissertation, University of Grenoble (2015)
4. Na-Lumpoon, P., Fauvet, M.-C., Lbath, A.: Toward a framework for automated service composition and execution. In: Proceedings of the 8th International Conference on Software, Knowledge, Information Management and Applications, Dhaka (2014)
5. Na-Lumpoon, P., Lei, M., Caicedo-Castro, I., Fauvet, M.-C., Lbath, A.: Context-aware service discovering system for nomad users. In: Proceedings of the 7th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), Chiang Mai, Thailand (2013)
6. W3C. Web services glossary (2004). www.w3c.org/TR/ws-gloss