

Service Offer and Request Descriptions in Mobile Environments^{*}

– A POSITION PAPER –

Johannes Grünbauer¹ and Michael Klein²

¹Institut für Informatik, Technische Universität München,
85748 Garching, Germany, gruenbau@in.tum.de

²Institute for Program Structures and Data Organization, Universität Karlsruhe,
76128 Karlsruhe, Germany, kleinm@ipd.uni-karlsruhe.de

Abstract. In order to obtain robust and context-aware applications in mobile environments, service oriented computing can be a very promising programming paradigm. Here, the participants cooperate in a loosely coupled manner by invoking context-dependent services on demand. However, the approach demands for an expressive description of the offered and requested services. This paper aims at giving a basis for discussion on the requirements and approaches when describing services for mobile environments.

1 Introduction

Service oriented computing is a new paradigm that is especially interesting in mobile environments. As a characteristic, functionality is hidden behind an interface and described as a black box with the help of a service description language. This enables participants of the network to enlarge the limited capabilities of their devices by using services provided by others. As service requestors and providers are not fixedly tied together but are dynamically matched and bound, this architecture is especially advantageous in mobile environments and their constantly changing situation.

Typically a service usage follows the so called *service triangle* (see Figure 1): The service provider (which can be mobile) wants to offer a certain functionality. He describes it as service using the service description language (Step 1) and publishes this description at the service repository (Step 2). This repository can be central or distributed. If a requestor (which can be mobile, too) want to use a certain functionality, he described his requirements as service request (Step 3) and sends it to the repository (Step 4). Here, a matchmaking of the request and the offers takes place (Step 5). Matching offers are returned to the requestor, who selects one of them (Step 6), configures it according to his requirements

^{*} The ideas for this paper have been developed at the Dagstuhl Seminar 04441 on *Mobile Information Management* in October 2004 together with Georgia Koloniari, George Samaras, and Can Türker.

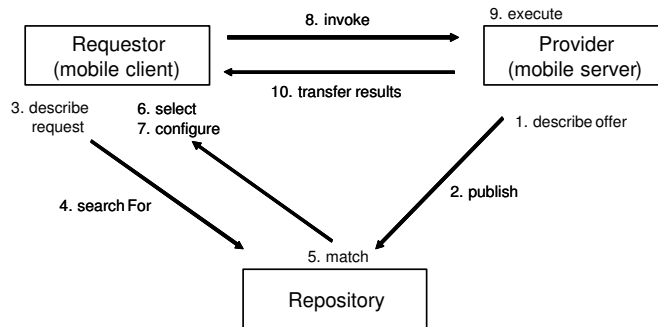


Fig. 1. The service triangle.

(Step 7), and invokes the corresponding service provider directly (Step 8). The provider executes the service with the given configuration (Step 9) and – if there are any results – returns them to the requestor (Step 10).

In the following, we want to give a basis for a discussion for two major topics from this process with regard to their characteristics with regard to mobile environments: Service offer descriptions (Section 2) and service request descriptions (Section 3). We conclude the paper by taking a look at some challenges in this area.

2 Service Offer Descriptions in Mobile Environments

First, we give a classification of services in mobile environments. This can be done by splitting the space along two dimensions (see Figure 2):

- *Mobility.* Whether the service itself is mobile or not.

	mobile	non-mobile
location dependent	Car Taxi (location affects functionality, context - related)	Restaurant
not location dependent	Tools on mobile devices (functionality is independent from the location)	Flight booking

Fig. 2. Classification of services in mobile environments.

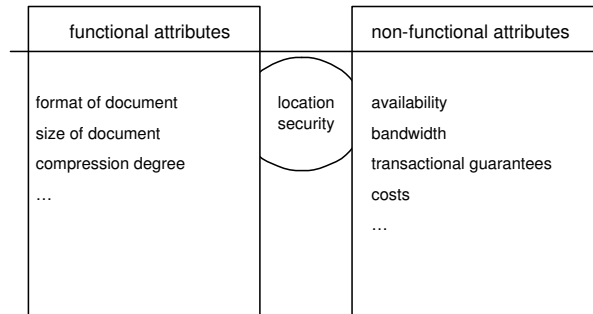


Fig. 3. Important functional and non-functional attributes for services in mobile environments.

- *Location Dependency.* Whether the content of the service is dependent from the location where it is used.

The simplest services are non-mobile, non location dependent services like a flight booking service in the internet. It runs on a central server and provides a functionality that is not bound to a specific location. In contrast, a restaurant booking service could enable a client to reserve a place in a restaurant in a certain area. Therefore, this is a location dependent service. Examples for service where the providing device itself is mobile could be a taxi reservation service (location dependent) or tools like a dictionary on a mobile device (not location dependent).

When describing these services, it is necessary to give information about certain attributes that are of special importance in mobile environments. They can be divided in functional and non-functional attributes (see Figure 3). Important functional attributes could be the format, size, or compression degree of a file as mobile devices typically have limited capabilities in dealing with all file types. With regard to non-functional attributes, availability, bandwidth requirements, transactional guarantees, and costs could be relevant when checking whether a given service is appropriate in a mobile situation.

However, filling the attributes of a service description leads to problems in a mobile environment: Changes in the environment can also lead to changes in the description, so frequent updates of the description would be necessary. To avoid this, the description could be split up into two parts: a static part containing the regular service description and a dynamic part, which captures the current context of the service provider like his current location, the times of availability and so on. This division could be done logically only or lead to two different documents which could be stored in two different repositories. With our classification from above, mobile services would have a large dynamic part whereas non-mobile service would only have a small one.

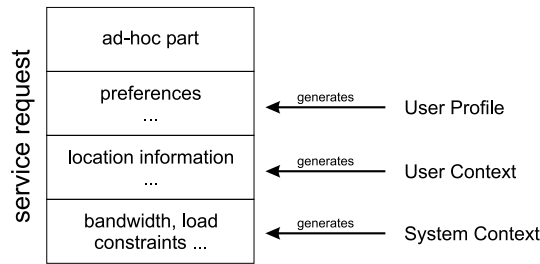


Fig. 4. Generating requests.

3 Service Request Descriptions in Mobile Environments

Requesting services is quite similar to performing a database query. The user or a program needs to have a certain query language to get information from a service. However, we neither create a new query language nor relate to an existing one in this section, but we present some ideas for request descriptions in mobile environments.

In mobile environments there exists additional information which has to be given to the service—the context information (cf. Sect.2). The context adds additional constraints to the request. We call these constraints *implicit information*.

As shown in Fig.4, a mobile request is assembled by different parts: The *ad-hoc part*, the *User Profile*, the *User Context* and the *System Context*.

The implicit information is generated by different aspects:

- The *User Profile* generates information like preferences of the user. This profile can be set up manually or be generated by the user’s behaviour over a period using an AI-system. The preferences could contain information like “user prefers Chinese food” or “user likes musical shows”.
- The *User Context* generates information like location information. This makes sense when ordering a taxi or in case of using a PDA with a travel guide system which can e.g. offer information about sights nearby the user’s current position.
- Furthermore, the *System Context* generates information like bandwidth or load constraints. To continue the travel guide example, we assume that the user wants to watch an information movie on his PDA about a sight he is currently standing in front of. If the bandwidth is low, the system should provide the user a movie with a low resolution or sound with a low quality.

The *ad-hoc-Part* is the request generated by the user. This is the explicit information the user gives to the service (“I need a taxi in 20 minutes” or “I want to reserve a table in a restaurant.”). It must be possible to overwrite the generated implicit information by the *ad-hoc part*.

Example 1 (Taxi Service). A person queries a service to get a taxi (“I need a taxi in 20 minutes.”). Here we have to distinguish between implicit and explicit

information. The explicit information is, that a taxi is needed in 20 minutes. But this information is completely worthless if the service doesn't know *where* the taxi should be in 20 minutes. So, here the implicit information (user context) is the location of the user.

If the query contains explicit location information (“I need a taxi in 20 minutes *at the Plaza Hotel*”), the information generated by the user context will be overwritten. \square

Example 2 (Reservation Service). Another example is a service, which can be used to reserve a table in a restaurant (“I want to reserve a table in a restaurant.”). The implicit information for the service request could be:

- the restaurant should be nearby (user context)
- the user prefers Chinese over Italian food (user profile)
- advertising movie about the restaurant in a certain quality available (system context)

Again, the implicit information can be overridden with queries like: “I want to reserve a table in a Greek restaurant in Munich”. \square

4 Summary and Further Challenges

In this paper, we started the discussion on requirements and approaches for describing and matching services especially in mobile environments. We presented a classification of services in these environments and showed which attributes in the description are necessary here. Moreover, a first approach to complete ad-hoc queries with profile and context information was sketched.

However, as this is a rather new research topic, a number of open questions remain. On the one hand, it has to be analyzed how detailed service descriptions should be, i.e. which constructs are needed to build and process them. On the other hand, it should be researched how such descriptions can be compared. Moreover, questions about appropriate similarity measures arise.