Fraunhofer Institut
Software- und
Systemtechnik

1. Fachgespräch "Ortsbezogene Anwendungen und Dienste“

Service Roaming

Rüdiger Gartmann
Fraunhofer ISST, Dortmund
24. Juni 2004

1 Introduction

Mobile users may frequently change their position over time. As long as a service accessed by a user is globally valid, this should not cause any problem. However, often services are limited in their scope. For instance, a traffic information service typically covers a certain region. It may occur that the user leaves the region covered by the service. This leads to the following problems. First, the user has to realize that he or she left the covered region, then he or she has to search for another service covering the current position, and finally he or she has to access the service and request the information needed.

For applications using this kind of services, the problem is even worse. Thinking of a general navigation system (in contrast to a proprietary solution) requiring traffic information for dynamic routing, the navigation system has to search automatically for suitable services and switch between services when leaving and entering their scope. This means that any mobile application utilizing services of limited scope should implement a mechanism for controlling service scopes and for switching between services when necessary.

A more efficient approach would be to provide a service roaming solution for such services. This is motivated for instance in [RoHD03]. Roaming is well known in the area of mobile communications and cell phone networks, giving users connectivity independently of their current position, the reachable base stations, and the available cell phone network providers. A similar functionality would be desirable for services as well. The difference between these types of roaming is that a physical [ETSI99] and a network roaming [MaYO03] (such as for cell phone networks) is caused when a base station is no longer physically reachable, whereas services, given a constant network access, are usually worldwide available. Here logical reasons are triggering a roaming event. Logical reasons in this respect are validity constraints of services concerning any context dimension such as location or time. Whenever a user context, which is the current state of a user regarding the context dimensions, lies within these constraints, services are valid, otherwise they are not and there is a need for roaming to services fitting to the new user context.

From an economical point of view, service roaming allows application providers to build their applications on a distributed set of services with different scopes. Even if the application covers a large area (such as a whole country or even the world) this application can use services with limited scopes and roam between them whenever necessary. From a service provider's point of view it becomes possible to integrate services with a limited scope into applications of a broader scope by using the concept of service roaming.

2 Definitions

Context is defined in [DeAS99] as "any information that can be used to characterize the situation of an entity". For roaming, however, only a relevant subset of this situation information is needed [Dey00]. Regarding service roaming a context is always related to a certain service request. This is why we use the term *request context*. It refers to the context a request is dedicated to. This context usually consists of several dimensions such as location, time, temperature, and soon. For mobile services, the request context normally is identical with the user context. This means that when a user is at a certain place, he or she will request services which are valid at this place. Despite of this we distinguish between user context and request context as users may also want to request services for other contexts than their actual one, e.g., when they are traveling and when during their travel they want to access hotel information for their destination. Here, obviously, the user context dimension 'location' is somewhere on the trip whereas the request context dimension 'location' is at the destination.

Since context consists of several dimensions its range R can be seen as an n-dimensional space consisting of dimensions D_i.

$$R = D_1 \times D_2 \times ... \times D_n$$

A concrete context C_i accordingly is one point in this space, which is described by the context values for each dimension as a vector. Here it does not matter whether user or request context are regarded, both are described in the same way:

$$C_i = (d_1, d_2, ..., d_n); d_k \in D_k$$

On the hand we service scope is also of great importance. Given a service set S, many services s_i have a limited validity concerning one or more context dimensions. So for every service $s \in S$ a scope A_i has to be defined:

$$scope(s) = A_s \in R$$

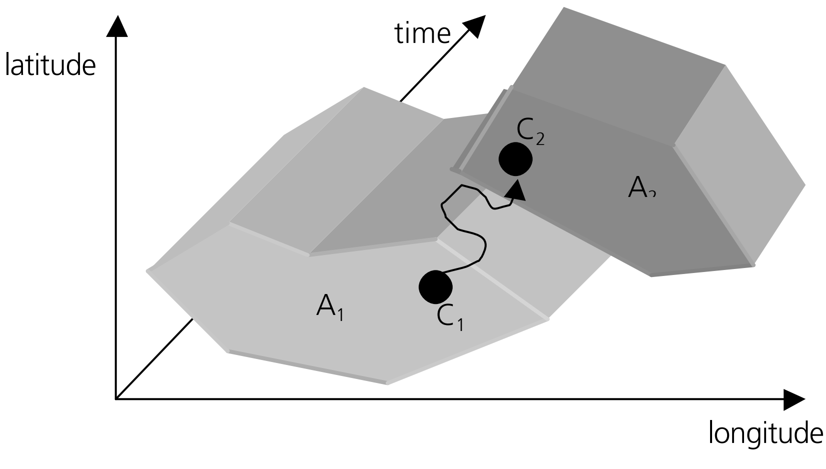
The service scope A_s defines all contexts in which this service is valid. This may be a region where the service is supposed to be used, a time range or any combination of intervals on context dimensions. Concerning the n-dimensional space of the whole context range, the service scope can be seen as one or more fields within the context space, being defined as limitations for this particular service along the context dimensions. A sample definition for a service scope A_1 in a three-dimensional context space consisting on geo-coordinates latitude (d_1) and longitude (d_2) as well as time (d_3) for service s_1 could look as follows:

$$A_1 = \{(d_1, d_2, d_3) \mid 51.2 \le d_1 \le 51.4 \wedge 7.3 \le d_2 \le 7.5 \wedge 08.00 \le d_3 \le 20.00\}$$

This could be a scope description for a service valid for the city of Dortmund and available during business hours. Whenever a request context lies within one of these fields, the according service is valid for this request. Hence, the relevant set of services S' for a certain context C_i is defined as:

$$S' = \{s_k \mid C_i \subseteq A_k\}$$

Figure 1 Visualization of service scopes and contexts in a context space



This concept is visualized in Figure1, which shows a three-dimensional context space consisting on two location dimensions and on a time dimension. There are two services s_1 and s_2, whose scopes A_1 and A_2 are displayed in the figure. Moreover there are two contexts C_1 and C_2 that are indicated. In context C_1 the appropriate service would be s_1, since the context lies within its scope. By changing the context from C_1 to C_2 the new context no longer lies within A_1 but in A_2. Thus a roaming from service s_1 to s_2 has to be performed.

3 Conclusion

With service roaming the top-level roaming functionality is provided to ensure a real roaming for mobile users. On a physical layer roaming is provided by many network types such as GSM, UMTS, or WLAN. Roaming on a network layer is solved, since when roaming on a physical layer the logical network connection usually stays unchanged, which means that devices keep their network addresses and connections.

What was still missing was the roaming ability on the application layer. Mobile users using somehow restricted services always had to care about monitoring the service validity for the current context and about searching for new services to replace the currently used ones for themselves. The use of service roaming disburdens the user from these efforts providing a certain functionality seamlessly, no matter which service instance is used to perform this functionality.

4 References

[RoHD03] Roos, A., Hartmann, M., Dutnall, S.: Critical Issues for Roaming in 3G. In: IEEE Wireless Communications, February 2003, 29-35.

[ETSI99] European Telecommunications Standards Institute: TS 100 521 V7.0.0. Digital cellular telecommunications system (Phase 2+); Network functions. (GSM 03.01 version 7.0.0 Release 1998), 1999.

[MaYO03] Matsuoka, H., Yoshimura, T., Ohya, T.: A Robust Method for Soft IP Handover. In: IEEE Internet Computing, March/April 2003, 18-24.

[DeAS99] Dey, A., Abowd, G., Salber, D.: A Context-Based Infrastructure for Smart Environment. In: Proceedings of the 1st International Workshop on Managing Interactions in Smart Environments (MANSE'99).

[Dey00] Dey, A.: Providing architectural support for building context-aware applications. PhD Thesis, College of Computing, Georgia Institute of Technology, 2000.