Notification Services for Mobile Scenarios

Michael Decker
University of Karlsruhe, Germany

INTRODUCTION

In this article we introduce the concept of generalized notification services (NSs). NSs are a simple class of services for mobile and wireless terminals, but nevertheless there are many useful services which can be modeled as NSs. Mobile and wireless terminals (MWTs) in our sense are handheld computers with a wireless interface for data communication (e.g., GPRS, UMTS, or WiFi); examples of MWTs are cellular phones, personal digital assistants (PDA), and smart phones. Mobile services are functionalities offered by one or more remote computers (server or back-end systems) to a MWT (client). Because of the mobility of the MWT, at least the first part of the route for the necessary data communication with the server is realized using wireless standards like GPRS, UMTS, or WiFi.

The motivation for giving an exact description of a class of mobile services comes from the fact that in literature there are many descriptions of platforms or technical frameworks for mobile services, but often there is no precise definition of what kind of services are supported by these platforms; these descriptions are more concerned with architectural aspects or how to deal with context information. Also the prevailing paradigm for the realization of mobile services is the Web-like delivery of pull-documents (e.g., iMode) which is not ideal for mobile scenarios, because poor connection quality impairs the user experience (necessity of sending requests again, long waiting times) and push messages are more appropriate for many mobile scenarios.

The remainder of this article is structured as follows: in the next section we cover our understanding of context-awareness. We then explain the basic principle of NSs and argue why they are suitable for mobile scenarios. To show that NSs are a versatile class of mobile services, we mention several examples from different areas of application. Before we summarize, we sketch a protocol for the implementation of NSs.

CONTEXT-AWARENESS OF MOBILE SERVICES

Context-awareness of mobile services (and applications) is an important concept of mobile computing. Context is defined as: "information that can be used to characterize the situation of an entity" and "is considered relevant to the interaction between a user and application" (Dey, 2001). Since almost everything can be considered as an entity at some level of abstraction, we focus more on the purpose of context: to support a user when interacting with a service, for example, displaying information relevant to his or her current situation, or reducing the amount of data to enter manually. The context information has to be available in an explicit form during the runtime of the system. As discussed below in more detail, mobile services have several restrictions with regard to ergonomic aspects, so context information is crucial for the user experience of mobile services.

For the description of NSs, we need to distinguish how critical a given type of context information is with regard to data protection. Context information that is critical for the user's privacy is denoted as "personal context," otherwise as "public context." An example for personal context would be the current location of a user, because people do not like the idea of having their position tracked. "Time" or "weather" (e.g., a mobile service should not recommend outdoor activities when it is raining all day long) are public context parameters because they are not person-related data (but to interpret them, it could be necessary to have personal context information, e.g., the user's location to find out the correct time zone or weather).

BASIC PRINCIPLE

The basic idea behind notification services is that users in mobile scenarios do not want to browse for the information they need in longish sessions and thus push mode for information delivery is preferable. A notification service sends push messages to users based upon an initial configuration and public as well as private context information. Munson and Gupta (2002) already mentioned the idea of generalized notification services, but they considered only location information as a context parameter and their discussion of details concentrated on how to provide location information for a large number of clients.

Our definition of NS demands a special client application on the MWT and a machine-readable service description file for each type of NS. Using the service description the client application can guide the user through the configuration of the NS without any network interaction. The description file also specifies which personal context parameters the client application has to provide (e.g., current location of

the user retrieved from a GPS module, profile information, battery level).

A configured instance of an NS is denoted as "order." The order is submitted to the server where the actual business logic of the NS is running. If certain events occur, the server will send push messages (notification) to the MWT (e.g., SMS/MMS, e-mail, or a client-specific channel based on TCP/IP connection). The content of the push message might lead to further interaction (e.g., the message might contain a link to a WAP-document with further information), but this is beyond the scope of the NS. Depending on the type of order, it might be allowed to reconfigure the order using the client application; also the client application might submit updates of the personal context information (e.g., new position of user) to the server.

To illustrate this concept we consider the example of location-based advertising (Kölmel & Alexakis, 2002). In this scenario a mobile service sends advertisement messages to a user's MWT if the user approaches stores with offers matching his wish list (configuration) and his profile. If implemented as NS, the configuration would describe which offers the user is interested in (e.g., clothes, entertainment, restaurants, etc.); according to his or her current position and profile information (personal context information), the user could receive notifications (advertising messages), for example, about a restaurant not far away from his or her current position.

In Figure 1 we illustrate the messages exchanged between the client on the MWT and the back-end systems of a generic order as a UML sequence diagram (Fowler & Scott, 2000). Steps 2-6 might occur several times, the order may vary:

1. **Submission of Initial Configuration and Personal Context Information:** This step is the only mandatory one. If the NS needs personal context parameters, these are automatically filled in by the client application (e.g., current location of the user or profile information).
2. **Communication with Third-Party Systems:** The back-end systems of the NSs may communicate with third-party systems (e.g., databases to query data or public context information). If the third-party systems proactively send information to the back-end systems (which do not query if they need information), the NS is following the publish/subscribe paradigm which is considered suitable for mobile services (Huang & Garcia-Molina, 2004).
3. **Automatic Update of Personal Context:** If during the lifetime of an order a relevant personal context parameter should change (e.g., location of user, battery level), the client application automatically sends an update to the back-end system.
4. **Update of Configuration:** The user can change the configuration of the order at any time; this might not be reasonable for all kinds of NSs.
5. **Notification:** If the business logic of the NS on the back-end systems detects an event, it dispatches a push message to the MWT. A push message is a message sent to a user without being perceived as being directly requested. There might be multiple notifications or not a single one depending on the type of service.
6. **Termination of Order:** Orders might terminate themselves (e.g., based on configuration parameter "expiry date") or be terminated by the user manually.

Figure 1. Sequence diagram for a generic NS