DECOMPOSITION AND REUSE OF MOBILE SERVICES

Elisabeth Dörr

E-Commerce Competence Center – EC3
Donau City Straße 1, A-1220 Vienna, Austria
elisabeth.doerr@ec3.at

Werner Winiwarter

Faculty of Computer Science, University of Vienna
Liebiggasse 4/3-4, A-1010 Vienna, Austria
werner.winiwarter@univie.ac.at

Abstract: *Today, the development of new mobile services is associated with high costs and risks. The main aim of our research is to make this development process faster and more efficient by making available a modular service development platform. We decompose existing services into generic components of a mobile service design space to reuse them as building blocks for the construction of new mobile services. This paper reports on the first steps towards this final goal.*

1. Introduction

There has been a lot of discussion about the future of m-commerce [7, 15] and expectations were high for this new technology. Today, disillusionment prevails. Often the actual product experience of mobile services does not fulfill the expectations, which causes frustration for the user.

However, mobile phones and other mobile devices are quickly becoming indispensable in our modern society. According to [9] the penetration rate of mobile phones in Austria is about 80%. The drive towards mobile devices has led to the integration and convergence of various technologies into a wide range of innovative mobile and multimodal applications. Novel approaches are therefore needed for the development and evolution of services on and across different mobile platforms.

Currently, it is largely unclear, which are the crucial factors for successful services [9, 20]. Technology in itself is not enough to drive the success of the business if useful and convenient contents are not displayed. Creating end user customized services is critical for successful services. Another factor, often overlooked, are technical problems like network failures, insufficient devices, etc., which prevent success. A common problem is still a lack of user convenience. As mobile devices get smaller, users do not want to type long text or click through large menu trees. Services get more complex and the number of available services is increasing. The big challenge is to reduce complexity for the user, so that he can use the service intuitively [22]. An important factor for easy usage is that there is not only one way the service can act but that there exist many possible alternatives. Attractive, intuitive, and easy-to-use services, personalized applications, and ubiquitous access to location-based services dominate the users' expectations [7].

The future success of 3G wireless networks depends on the widespread availability of innovative services [13]. Such services will increase the network usage for data and voice services. However, so far, services are provided only by individual operators and have been strongly tied to the particular capabilities of the operators' networks.

By providing components for using the new technology, services can be created easier and faster. This has advantages not only for the developers, but also for the users. It is a great benefit if interaction with users is harmonized and oriented towards their needs. All this can be achieved through the reuse of generic service components derived from the decomposition of existing mobile services within a modular service development platform.

The remainder of the paper is structured as follows. In Sect. 2 we present a classification scheme for mobile services. We focus on 3G networks because the new emerging services need a higher data rate and performance than conventional services. Mobile multimedia services introduce new challenges, such as traffic asymmetry, driven by the wide variety of multimedia-based activities available to the user. Therefore, the development of such new services is more complex than the development of traditional services. In Sect. 3 we describe the characteristics of service development platforms for mobile services and provide a short overview of commercially available tools. Finally, Sect. 4 discusses existing approaches for service composition and identifies candidates for mobile service components.

2. Service Classification

In the literature one can find many, partly controversial, definitions of the terms *application* and *service*. For the purposes of our research we follow [11] in that we define an *application* as "a task that requires communication of one or more information streams, between two or more parties that are geographically separated, being characterized by service characteristics, and also by traffic and communications characteristics". Again, according to [11] a *service* is then defined as "a set of applications with similar characteristics, or a single application, if they have a common set of characteristics".

There are many different kinds of services, ranging from simple SMS messages to formalized transactions. In the literature many different service classifications can be found. A popular approach is to divide services into: *information*, *communication*, *entertainment*, and *transaction* services [11, 14]. Simple entertainment services, for example, can be easily classified with this scheme. However, when classifying more complex services this method is often insufficient. The classification shown in Fig. 1 is based on the ITU recommendation [11, 21].

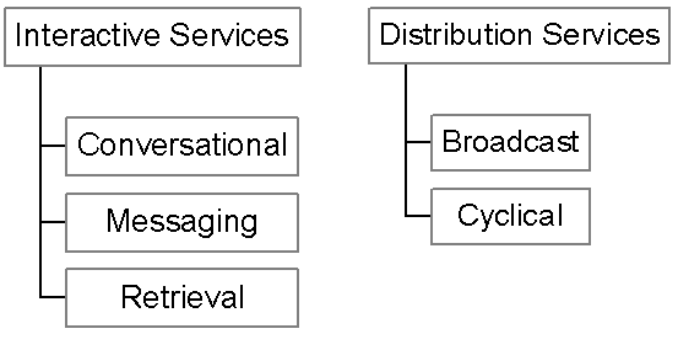

Figure 1: Classification of services and applications according to ITU-T I.211 recommendation

Interactive Services are those in which there is a two-way exchange of information between two users or a user and a service provider. Three different subcategories can be distinguished:

- *Conversational Services* provide bidirectional communication between two users or between a user and a service provider host. The flow of information may be bidirectional symmetric, bidirectional asymmetric, or in specific cases, unidirectional. Conversational services are real time (no store and forward). Applications are chat, videotelephony, or videoconferencing.

- *Messaging Services* offer user-to-user communication between individual users with store and forward units (mailbox or message handling devices). They are not in real time. Messaging services might typically provide combined voice and text, audio and high-resolution images. Possible applications are video messages or voice mail.

- *Retrieval Services* provide the user with a capability to retrieve information stored in information centers. The user can control the start of an information sequence, which is sent by an information center to the user. Each information center accessed may provide a different media component, e.g. high-resolution images, audio, and general archival information. Applications are broadband retrieval services for film or audio information.

Distribution Services are those whose information transfer is primarily one-way, from service provider to subscriber, including broadcast services, where the user often has no control over the presentation of the information, and cyclical services, which allow the user more presentation control:

- *Broadcast Services* provide a continuous flow of information, distributed from a central source to an unlimited number of authorized receivers connected to the network. Each user can access this flow of information but has no control over it. A possible application is the broadcast of an electronic newspaper broadcast service.

- *Cyclical Services* allow the distribution of information from a central source to a large number of users. The information is provided as a sequence of information entities with cyclical repetition. The user can control start and order of presentation. Applications are teletext or an electronic newspaper using public networks.

We use this classification in our research because, in our opinion, most services can be assigned to exactly one class. This is important for the reuse of mobile services, as with this classification we can identify common attributes of services in one class. Recurring attributes can be used as components for further development of new mobile services.

3. Service Development Platforms

The interest in developing new mobile and wireless technologies is high. Complex functions, novel services, and the handling of enormous amounts of information poses huge challenges at the technical as well as at the business level [18]. The efficient and fast development of complex mobile services is still the greatest barrier.

For handling these challenges, an innovative architecture is necessary, which meets certain requirements. Key attributes of such an architecture are [12]:

- *Modular* – the platform consists of software components, which can be reused in different services.

- *Flexible* – new service groups require new components, therefore it is important that the platform is flexible with respect to the introduction of new components. In addition, the adoption of existing services can be done easily.

- *Evolutionary* – services operate in traditional telecommunication systems as well as in New Generation Networks.

- *Content-independent* – there is no dependability on certain content providers, and it does not provide any content itself. Therefore it is easy to integrate new content.

- *Open* – open for all developers who want to create new services. Today, often the infrastructure vendors also develop services; a development platform must be also open for content providers and application developers.

At the moment many development platforms are available, however, it is very difficult to get detailed information about certain products. For this reason, it remains unclear to a large extent, which platforms meet all the above-mentioned criteria. Many platforms do not support the usage of service components or offer only limited support.

In the following two subsections we first describe the characteristics of a generic mobile application platform and then present three commercially available platforms.

3.1 Mobile Application Platforms

Mobile application platforms support the development and management of mobile applications and services. They mostly consist of a three-tier architecture, including a client-tier, residing on the mobile device, a middle-tier, consisting of business applications and a set of middleware and network services, and a backend-tier consisting of databases and applications [12].

Figure 2 shows the architecture of a mobile application platform. The individual components are characterized as follows:

Mobile Application

The purpose of such a platform is to support the development of mobile services. The system is specially developed for supporting mobile users and offers special functions. Therefore, it handles business functions involved in a particular organization (e.g. how to handle information requests) and provides features for the integration with backend applications (ERP, CRM, etc.) or databases.

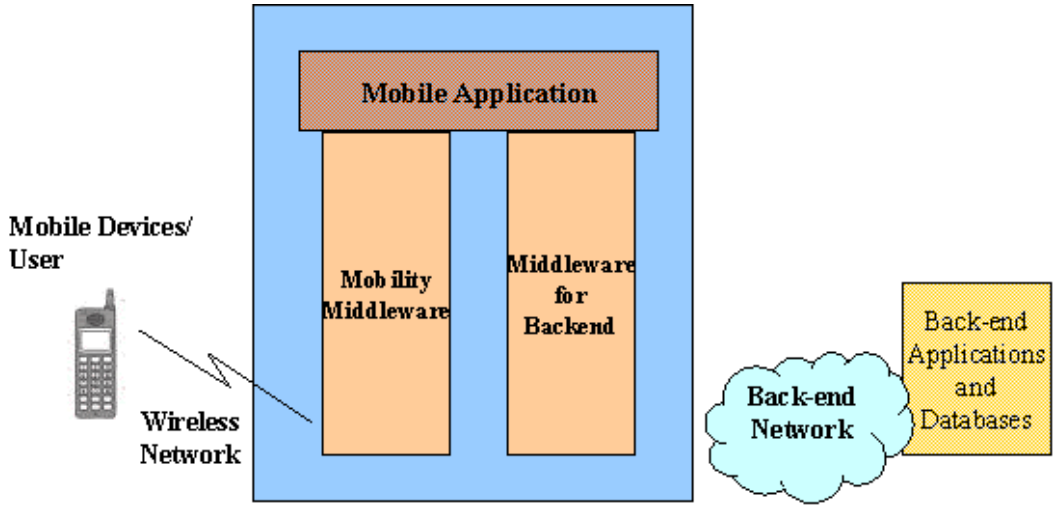

Figure 2: Mobile Application Platform Architecture, according to [1]

Mobile Devices

These could be mobile phones as well as notebooks, PDAs, Pocket PCs, etc. These devices use different operating systems that reside in the mobile device (WinXP/NT, WinCE, Palm OS, etc.). Services have to be able to adapt to all kinds of operating systems of the devices. Furthermore, the platform has to provide interfaces for the integration of new devices.

Wireless Network

This could be a wireless LAN, WAN, or MAN. Network providers are, for example, T-Mobile, H3G, AT&T (Canada), or NTT DoCoMo (Japan). Regarding the technical basis, some differentiations are possible. Depending on the used network, different features are possible for the realization of services. Therefore, services have to be able to adapt to different networks, both traditional networks as well as New Generation Networks.

Mobile Middleware

Responsible for handling mobile devices, the mobile middleware is an important component of the platform. This middleware takes raw data of database applications or queries and transforms it for the clients. Information has to be adapted to the limitations and peculiarities of mobile devices. This includes adaptation to different operating systems, content adaptation because of different display sizes, colored displays, not supported data formats, etc. This middleware also presents mobility specific services such as location.

Middleware for Backend

A backend application or program serves indirectly in support of the front-end services, usually by being closer to the required resource or having the capability to communicate with the required resource. A common application of middleware is to allow programs to access databases.

3.2 Commercially Available Development Platforms

Actual implementations of mobile application platforms vary among different vendors. Some application servers are generic Web servers with an SDK (Software Development Kit) or API (Application Programming Interface) capability, some offer customization capability in a horizontal application, and some are based on the WAP protocol.

3.2.1 BREW

Qualcomm's BREW wireless platform offers a method for development, configuration, and execution of software applications on mobile devices. It is a complete end-to-end solution that supports multiple programming languages, including Java, C/C++, and XML. In the BREW solution, a software development kit (BREW SDK), the BREW client software, and the BREW Distribution System (BDS) is included [19]. For application developers the BREW SDK provides an easy to use interface for rapid service development.

The BREW client software supports all types and tiers of devices, from low-end, mass-market phones to high-end smart phones. By means of extensions all kinds of browsers, video players, and Java Applets can be integrated in services. The BREW Distribution System supports wireless operators in managing Java-based applications, which are downloaded and executed on BREW devices [19].

BREW is an easy to use development platform, which provides components for small services. As the BREW solution provides only limited functionality it is not well suited for the development of complex services.

3.2.2 IBM WebSphere

The foundation of the platform is IBM WebSphere Application Server, which provides specialized configurations designed for advanced business needs [10]. Every Application Server configuration is powered by a single Java engine so that applications can be easily ported across editions when needs change. For development needs, IBM WebSphere Studio provides a suite of tools in configurations that span development for the Web, the enterprise, and wireless devices. It provides J2EE 1.3 support to simplify enterprise applications by basing them on standardized, modular components. The WebSphere Studio development environment is based on the Eclipse Platform, an open universal platform for tools integration. Application integration provides interfaces using standards like Simple Object Access Protocol (SOAP) and Web Services Description Language (WSDL).

3.2.3 OmniPlatform

Aligo's Omni Platform enables extending existing business applications to a mobile workforce, or to customize mobile applications [3]. It is a network-independent and standards-based platform, supporting Java and .NET clients. It supports virtually every type of wireless handheld device. The OmniStudio provides development capabilities to build a data access layer for integration with back-office applications and databases. It consists of Java 2 Platform Enterprise Edition (J2EE) components and is a comprehensive Enterprise Java Beans application server environment. A

special feature of this platform is that all applications are available in real-time, connected mode as well as in synchronized, disconnected, and occasionally connected modes [3].

The three above mentioned products show the diversity of available development platforms. It is very difficult to exactly determine all relevant functions of the systems because detailed information about the systems is not easy to get. Nevertheless, we can say that none of the platforms described above meets all requirements for using it as modular service development platform within our research. Therefore, we decided to create a new platform by taking parts of these development platforms into consideration.

4. Service Components

4.1 Service Composition

There already exists some work in the area of Web services composition (e.g. see Benatallah's work on semi-automated service composition [6, 23]). Composition is the merge of two or more services (service components). Many compositions are created manually by taking simple Web-accessible programs, such as form-validation programs or database lookup programs, and composing those components using typical procedural programming constructs such as if-then-else, sequence, or while-loop. Most recently, technologies have been proposed that use some form of semantic markup of Web services in order to automatically compose Web services to perform some desired task [12, 16, 17]. In order to raise the flexibility in designing the service compositions, one should apply small service units for dedicated tasks and compose them to advanced services.

The *Web Services Flow Language* (WSFL) is often used for Web Service compositions [16]. WSFL considers two types of Web Services compositions:

- The first type specifies the appropriate *usage pattern* of a collection of Web Services in such a way that the resulting composition describes how to achieve a particular business goal; typically, the result is a description of a business process.

- The second type specifies the *interaction pattern* of a collection of Web Services; in this case, the result is a description of the execution sequence of services.

WSFL also supports recursive composition, which means that the result of the composition is another Web Service.

A second approach is the *Business Process Execution Language* (BPEL) [5]. BPEL is an XML-based language that provides an abstract way to compose Web services, where the developer can specify which services will participate, and the sequence of execution. The BPEL server then takes care of the actual orchestration, data flow, persistence, and error handling. However, it is the developer who has to identify which service components can cooperate.

Both languages are XML-based and are used for the description of Web services compositions. It may be possible to use one of these approaches also for mobile services compositions. Currently, we are in the process of taking a closer look at both languages regarding the decision, which of the two languages will be better suited for the task of composing complex mobile services.

4.2 Mobile Services Components

Some development platforms for mobile services offer pre-built components, which can be integrated into new services easily. This saves time during development and standardizes services.

In [12] a platform for building applications using predefined components is described. This Mobile Application Server (MAS) is a Java middleware prototype platform for mobile services and provides some basic building blocks. The following subsections briefly describe the basic components that are provided [4, 12].

4.2.1 Authentication

Authentication is the process in which a network user establishes a right to an identity. Service customization requires strong authentication rules. This means that before using any service, the user (or the service) has to identify himself through simple login or password procedures, smart cards, or biometric methods. Authentication merely ensures that the individual is who he or she claims to be, but says nothing about the access rights of the individual.

Devices will have different ways of offering identification services. Depending on the service, varying authentication methods are useful. By offering services requiring authentication, devices provide not only personalization but also privacy to their customers.

4.2.2 Message Management

Notification works at several levels and with several objectives. Basically, it allows users to exchange data over a network. Users are notified of any communication as soon as it arrives at a certain device or when a certain user is logged on. Furthermore, message management assists in installation or configuration. It informs the user about upgrades of software or increased functionality.

4.2.3 User Profile Management

After the identification of a user it is possible to create preferences that implicitly and explicitly define how notifications and messages should be handled or how multiple devices should be coordinated. From the user's perspective user profiles enable to use services without entering personal data each time. In a personal profile, information about the user (what the user likes, dislikes, etc.) can be used to personalize services, so that users get only information they are interested in. Moreover, information about used devices, authorized services, data for privacy control, and mobile network data (e.g. WAP or GPRS parameters) can be stored [2].

The other side of the coin is that users want to keep their privacy. For handling this trust problem – when users do not know which data is stored, for what purposes it is used, to whom it is handed on, etc. – user profiles have to be available to the user.

4.2.4 Directory

The service directory includes all available services on a local network and interfaces with other directories. Thus, users can easily find services, people, or other information. It is useless to develop loads of services that the user does not know how to find. A good service directory is an important factor in increasing user convenience.

4.2.5 Location Information Management

The location component manages actual location information, continuously or on demand. The necessary location determination can be done in two different ways. As most available network technologies (e.g. GSM, GPRS, or HSCSD) have a cellular character, it is possible to use the wave propagation times to base stations of several cells to calculate the approximate position of a device. A more exact solution for location determination is realized through the interaction with a GPS daemon [8].

These basic components represent first steps towards mobile service composition. However, they only provide some building blocks of services, no complete mobile service can be created by just using those basic components. Since our goal is to compose services mainly by combining existing components, we have to create a comprehensive collection of generic services to enable the fast and efficient development of new mobile services.

5. Conclusion and Future Work

In this paper we have presented the first steps towards the development of a component-based mobile service development platform. By decomposing and reusing existing services we can speed up the development of new innovative services. As a first step we classified services to find common attributes of service classes, which can be used for the definition of generic components. We also took a closer look at mobile service development platforms and analyzed some commercially available platforms with the result that none of them are suitable for our research aim. Finally, we compared two approaches for service composition, which could serve as foundation for our composition framework, and presented several candidates for basic components as starting point for our collection of service components.

Future work will concentrate on the decomposition of existing services into generic components. Moreover, we will develop a formal representation of components as functions in a design space. The combination of function using operators will again result in new functions. Besides functional correctness, we also have to consider additional aspects such as execution time, price, or security options.

References

[1] 3GPP ORGANIZATIONAL PARTNERS, 3GPP TS 22.101. V 6.7.0. Service Aspects, Service Principles, 2004.

[2] 3GPP, 3GPP Generic User Profile – architecture; Stage 2; Release 6; Technical Specification TS 23.240 V6.3.0. http://www.3gpp.org/ftp/Specs/archive/23_series/23.240/, 2004.

[3] ALIGO, Inc., OmniPlatform, http://www.aligo.com/products/omniPlatform/, 2004.

[4] AMOR, D., Internet Future Strategies. How Pervasive Computing Services Will Change the Future. Bonn: Galileo Press, 2001.

[5] ANDREWS, T. et al, Specification: Business Process Execution Language for Web Services. ftp://www6.software.ibm.com/software/developer/library/ws-bpel.pdf, 2003.

[6] BENATALLAH, B., DUMAS, M, FAUVET, M-C, RABHI, F., Towards Patterns of Web Services Composition. In: Patterns and Skeletons for Parallel and Distributed Computing , Springer, 2002.

[7] CARLSSON, C. and WALDEN, P., Mobile Commerce – Empirical results on value-added products and services. In: Proceedings of the Second International Conference on Mobile Business, 2003.

[8] ETSI, Digital cellular telecommunications system (Phase 2+); Mobile Application Part (MAP) specification. ETSI TS 100 974 V7.15.0., 2004.

[9] FORUM MOBILKOMMUNIKATION, Studie zur volkswirtschaftlichen Bedeutung des Mobilfunks in Österreich mit ausgewählter Betrachtung der UMTS-Einführung, 2004.

[10] IBM, WebSphere Application Server, http://www-306.ibm.com/software/webservers/appserv/was/.

[11] ITU-T Recommendation I.211, B-ISDN Service Aspects. 1993.

[12] JAIN, R. et al, The Mobile Application Server (MAS): An Infrastructure Platform for Mobile Wireless Services. In: Information Systems Frontiers 6:1, 23-34, 2004.

[13] KROGSTIE, J., Applications and Service Platforms for the Mobile User. In: ERCIM NEWS, 54, July 2003.

[14] LAURIDSEN, O. M. and PRASAD, A. R., User Needs for Services in UMTS. In: Wireless Personal Communications 22(2), 2002.

[15] LEHR, W. and MCKNIGHT, L., Wireless Internet Access: 3G vs. WiFi?. eBusiness@MIT working paper #166. http://ebusiness.mit.edu/research/papers-author.html, 2002.

[16] LEYMANN, F., The Web Services Flow Language Guide, http://www-3.ibm.com/software/solutions/webservices/pdf/WSFL.pdf, 2001.

[17] MCILRAITH, S. and SON, T., Adapting Golog for Composition of Semantic Web Services, Proceedings of the 8th International Conference on Principles of Knowledge Representation and Reasoning, 2002.

[18] MILANOVIC, N. et al, Towards Adaptive and Composable Services. Proceedings of VIP Forum of the International IPSI Conference (IPSI 2003), Montenegro, 2003.

[19] QUALCOMM, BREW and J2ME. A Complete Wireless Solution for Operators Committed to Java. White Paper, 2003.

[20] SADEH, N., M-Commerce: Technologies, services, and business models. Wiley Computer Publishing. New York. 2002.

[21] VELEZ, F. J. and CORREIA, L. M., Mobile Broadband Services: Classification, Characterisation and Deployment Scenarios. In: IEEE Communications Magazine 40:4, 2002.

[22] WÖRTHER, G., Mobile Business: Erfolgsfaktoren aus Kundennutzensicht für 2,5G- und 3G-Services im B2C-Bereich unter Berücksichtigung der aktuellen Lage am österreichischen Mobilfunkmarkt. Diplomarbeit WU-Wien, 2004.

[23] ZENG, L., BENATALLAH, B. et al , Flexible Composition of Enterprise Web Services. In: Electronic Markets - The International Journal of Electronic Commerce and Business Media , Vol. 13(2), 2003.