The Simplicity Device: Your Personal Mobile Representative

Giovanni Bartolomeo[1], Francesca Martire[1], Enrico Rukzio[2], Stefano Salsano[1], Nicola Blefari Melazzi[1], Chie Noda[3], John Hamard[3] and Alexander De Luca[3]

[1] DIE – Università di Roma "Tor Vergata", Italy
{giovanni.bartolomeo, francesca.martire, stefano.salsano, blefari}@uniroma2.it
[2] Media Informatics Group, University of Munich, Germany
enrico.rukzio@ifi.lmu.de
[3] NTT DoCoMo Euro-Labs, Germany
{noda, hamard, deluca}@docomolab-euro.com

Abstract. In this paper we present the concept of a mobile phone acting as the user representative in the digital world. Such an enhanced mobile phone, which we call the Simplicity Device, can store relevant user data and communicate with other devices. Therefore, context-aware services and applications provided by the Simplicity Device or other devices can be tailored to the users needs. We designed a corresponding system architecture considering existing standards and privacy issues. Then we implemented a prototype of the Simplicity Device supporting several applications; three of them are discussed in this paper.

1 Introduction

The growing complexity of using computing devices, services and applications is a universally accepted problem. So many of us use different devices such as mobile phones, PDAs, Laptops, PCs or terminals to access services and applications via different networks. Within the research field of *context-aware services,* researchers from industry and academia have been working on solutions for this problem for the last decade. The basic idea is to have relevant context information, for instance about the user, her preferences, her devices or the location, which is used for the adaptation of services and applications. One common problem is the question of where information about the user is stored, how this information is defined and who can use it under which circumstances.

In this paper we present the concept of the Simplicity Device which is an enhanced mobile phone that stores and handles personal information about the user. The Simplicity Device can be connected (e.g. via Bluetooth) to several other devices thus allowing personalization of services and applications running on them. One important advantage of this approach is that personal and sensitive data are not stored on a server but carried by the user who therefore keeps control over it.

This paper presents the usage, architecture and implementation of the Simplicity Device. We also present the relationships between the Simplicity Device and the Terminal Broker, which is a software that must be installed on every device the Simplicity Device would interact with. More information about this software and the corresponding architecture can be found on the web page of the Simplicity project [1] and in [2]. We already used the Simplicity Device within several prototypes or scenarios. Three of them, *My PC, Tour Guide* and *Automatic Form Filling,* are discussed in this paper. To conclude, we relate our work to existing approaches and give an outlook on further work.

2 Usage of the Simplicity Device

Figures 1 and 2 illustrates how the Simplicity Device can be used in an enhanced mobile phone:

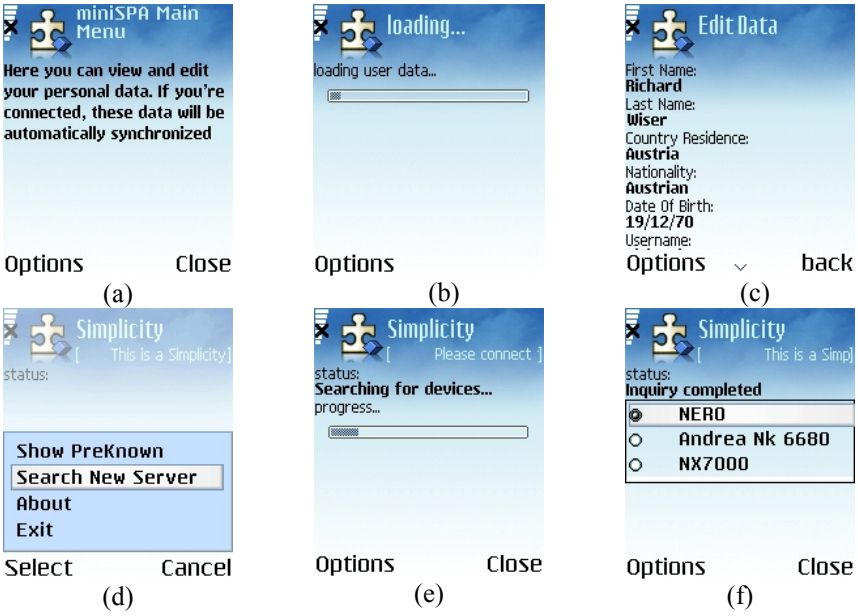
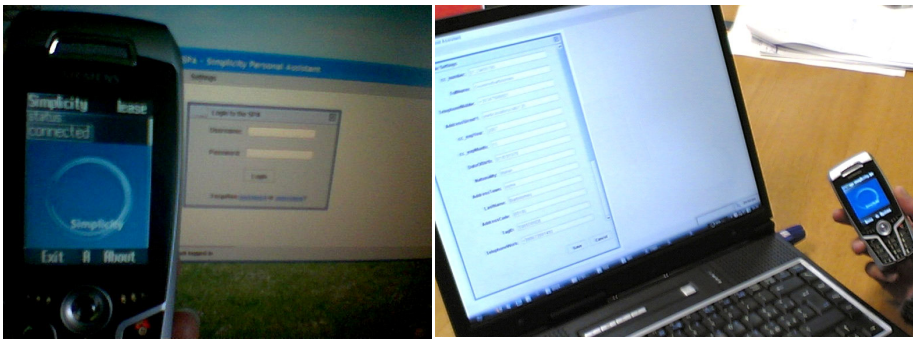


Fig. 1. Screenshots of the Simplicity Device

At first, the Simplicity Personal Assistant application is started on the phone. Personal data can be viewed and edited directly on the phone (figure 1a, 1b, 1c). Anyway, the Bluetooth interface of the Simplicity Device may be used to scan the environment for nearby terminals to connect to (figure 1d, 1e, 1f). After selecting a suitable terminal and connecting to it, the personal data stored on the Simplicity Device is used to adapt services, applications and networks (figure 2a, 2b).



(a) (b)

Fig. 2. Interaction of the Simplicity Device with a terminal.

3 Architecture and Implementation

Our work has been performed in the context of the Simplicity project [1]. Therefore, here we'll take a look at the Simplicity architecture and the Simplicity User Profile (SUP).

The Simplicity System encompasses a set of software and hardware components for providing user services. The main components of the Simplicity architecture are the Terminal Brokers, the Network Brokers and the Simplicity Device. A Terminal Broker is a software component implemented in the user terminal. It manages the access to personal information stored in the Simplicity Device, network services and user interfaces via a user agent software called Simplicity Personal Assistant (SPA). The Network Broker is as well a software component residing on the network and providing support for service advertisement, discovery and adaptation. The design of the Terminal Broker and Network Broker is based on subsystems, stand-alone components that provide specific functionalities either to the final user or to other subsystems. Finally, the Simplicity Device holds user information such as user preferences and policies that constitute the so called Simplicity User Profile (SUP). Some implementations of the Simplicity Device (noticeably on mobile phones) can also run an optimized version of the SPA and make it possible to view and edit user's data without connecting to the Terminal Broker.

The SUP has been designed considering inspiration by the 3GPP Generic User Profile (GUP) and more particularly by its hierarchical structure, described using the Data Description Method [20] and based on abstract components. Unlike the GUP, however, the SUP implements (but is not limited to) five concrete components:

- user profile (mostly based on Liberty Alliance Project Personal Profile [11]),
- device profile (based on a UAProf Schema provided by the WAP Forum [12]),
- network profile,
- service profile and
- Simplicity Device profile.

Thus the SUP is intended as a user level representation of the user herself, her context, device, network access and the services to which she is subscribed. The SUP is expressed in XML format.

Figures 3 shows the static model of the software application which enables a phone to act as a Simplicity Device and connect to remote devices. There are two main classes: the Controller and the mini Simplicity Personal Assistant (miniSPA). The Controller manages the connection when the Simplicity Device is connected to remote terminals using asynchronous messages exchanged over a Bluetooth link. The miniSPA allows the user to view and edit her profile directly on the phone by using a graphical interface without establishing any connection with remote devices. To speed up performances, the functionalities handling the user profile have been encapsulated in just a few classes, the most important ones of them are the DomManager (for handling XML data), and the MemoryManager.

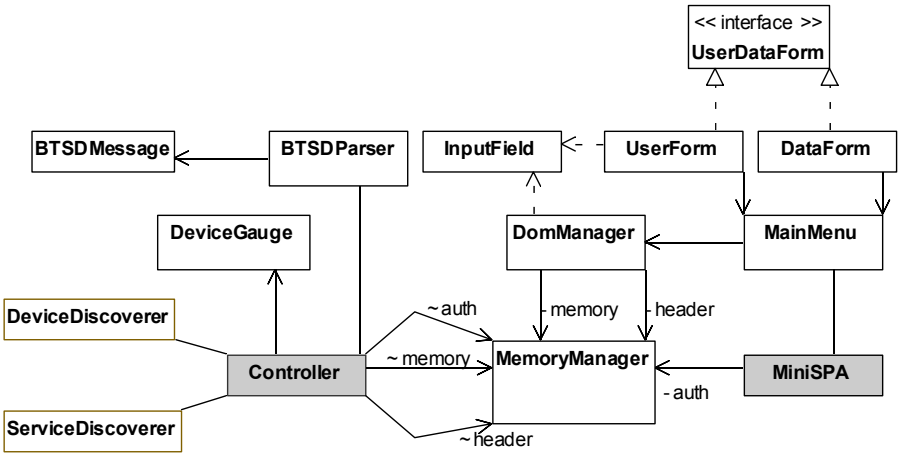


Fig. 3. Architecture of the Simplicity Device: UML class diagram

Figure 4 shows the dynamic model of the Simplicity Device. After an initialization procedure to set up its memory areas, the Simplicity Device is ready to use. In this state, the Simplicity Device may connect to other Bluetooth devices and after the user logs in and authenticates it is ready to interact with them. The user may log out without disconnecting, or also disconnect the Simplicity Device after or without logging out.

For the development of the Simplicity Device we used Bluetooth phones running Java 2 Micro Edition (J2ME). We tested our application with real phones including Nokia 6600/6630, Motorola A1000, Sony Ericsson P900 and Siemens S65. The developed Java 2 Micro Edition Midlet Suite uses the additional APIs Java APIs for Bluetooth Wireless Technology [14] and kXML [16].

The terminal software runs on a Windows machine running Java and commercial Bluetooth libraries [17,18], but we're investigating other possible supports [19].

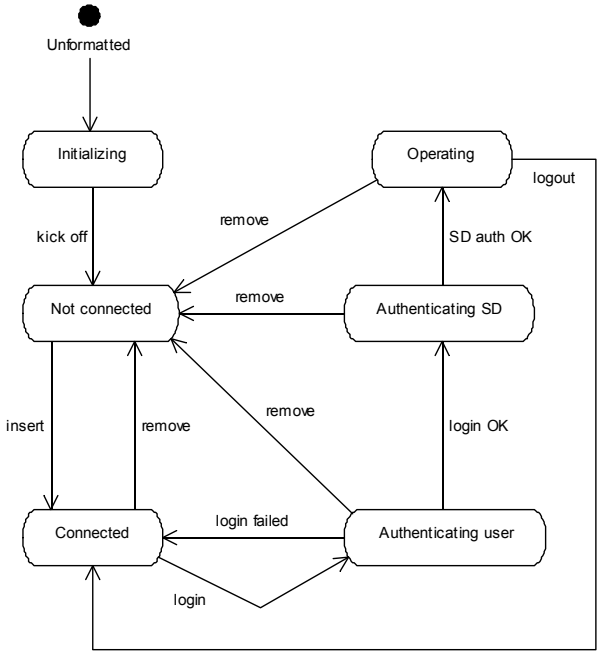


Fig. 4. The state machine of the Simplicity Device: UML state diagram

4 Applications

We used the Simplicity Device within several demonstrators of the Simplicity project [1]. In this section we will briefly discuss three of them.

The basic idea of *MyPC* (Figure 5a) is that personal information is used for adapting the operating system and the applications on a normal PC. There are several solutions in which the user profile and preferences are used to adapt the desktop of the PC. However, these are mostly business solutions which are not available for private users and which have the disadvantage of having to store private information on a central server. Modern mobile phones like the Nokia N91 with a storage capacity of 4 GByte show that it is already possible to store a user profile on a mobile phone. This idea was realized with *MyPC* where the profile is securely stored on the Simplicity Device. When users try to connect their Simplicity Device to a terminal, a screen is displayed asking if they want to connect to a pre-known terminal (e.g. “my laptop”, ”PC at office”, etc.) or whether they wish to look for new terminals. In the first case, the phone connects directly to the target terminal; in the second case, the phone searches for new suitable devices and lets the user decide which one to connect with. The user then authenticates himself (using a password; alternative mechanisms based on recognition of displayed contents, physical token possession or gesture [10] might be employed). Finally, information from the user profile is used to customize the target PC.

The hardware of the *Tour Guide* demonstrator which is depicted in Figure 5b consists of a Tablet PC that is connected to an external GPS device. The idea is that such hardware could be borrowed at the tourist office of a city and that the corresponding software on it is Simplicity enabled. The user can connect to the Tablet PC via Bluetooth and based on the information on the Simplicity Device the tour guide application is adapted. For instance the user profile includes layout preferences like "I prefer big font sizes" or "I prefer images instead of extensive textual descriptions". In addition to that the profile also includes information about the interests of the user like whether he is interested in history, shopping or restaurants. The tour guide could also be used for things like reserving a car or buying a ticket for a museum. Therefore the personal data of the user is required which can be requested from the Simplicity device.

A further prototype is the *Automatic Form Filling* (Figure 5c). Here profile information like name, address, bank account or credit card is used to automatically fill information into forms for buying or reserving something. There are three different version of the automatic form filling function: one for a PC or a Laptop, one for a PDA and one for the mobile phone. When looking at the desktop version, our implementation can be compared with the *AutoFill* function of the *Google Toolbar* or the *Auto form fill* function of the *MSN Search Toolbar*. However, our solution works also on mobile phones because of a special architecture and algorithms which are optimized for such devices; further it can be used with any web browser because the form filling function is provided by a proxy.

(a) (b) (c)

Fig. 5. Screenshots of the MyPC (a), Tour Guide (b) and Automatic Form Filling (c) prototpyes.

5 Related Works

In our work we focus on a subset of the research field of *context aware services* in which methods for gathering, describing, structuring and using context information for the adaptation of services are discussed [6, 7]. Our focus lies on the description of personal data, the storage and management of this data within the user's mobile devices, the interaction of it with several terminals and the usage of the personal information for the adaptation of applications, terminals and services.

There are several approaches and standards for the description of user data like address information, preferences, policies, bank account or credit card information. The 3GPP Generic User Profile [8] is an emerging standard with a grammar framework for the definition of the syntax of user profiles. 3GPP is an open framework which makes no assumptions about the content of the profile, limiting itself to the definition of its structure. The W3C Working *Draft Client Side Automated Form Entry* [9] defines data elements for describing the user data which are grouped into identity information, contact information, postal information, postal information for billing purposes and organizational information. The idea of a personal server where the user carries all his personal data and programs as well as a full PC in the size of a PDA with him or her is investigated by Intel's Ubiquity Personal Server [21, 22]. The Mobile Personal Server from Realm Systems [23] follows a similar approach. The advantage of our approach is to extend the mobile phone which everybody already uses instead of introducing another device.

6 Conclusion and outlook

Within this paper we presented the idea of using a mobile phone, which we call the Simplicity Device, as the user's representative when interacting with the digital world. The advantage of this approach is that the user has her personal data with her which does not have to be stored on a server in the internet. This solves an often discussed privacy issue of the research field *context aware services* in which information about the user, such as address information, preferences or settings, is used for the adaptation of applications, services, networks or terminals.

We also showed an architecture and a corresponding implementation of this concept which takes existing devices, standardized data handling and widespread communication protocols into account. Based on this we presented three different applications which use the Simplicity Devices for the adaptation of services, applications and terminals. The advantage of our approach compared to others is the usage of widespread Bluetooth enabled mobile phones, an implementation based on the platform independent standard J2ME and an XML-based representation of the personal data of the user. Thus, it is easily possible to convert existing mobile phones into Simplicity Devices.

We currently plan to run user studies to evaluate the overall idea of our concept, the user interface and the adaptations based on the user profile. In parallel we work on the further development of the Simplicity Device. In particular, waiting for the release of the first real phones implementing API compliant with the *Security and Trust Services API for J2ME* [3], we are developing a prototype of a secure Simplicity Device able to mutual authenticate with a second Bluetooth device using certificates compliant with the X.509 standard and set up a secure communication channel to exchange user information. In addition to that we plan to develop a Simplicity Device with Near Field Communication (NFC) [4] functionality taking NFC/RFID phones such as the Nokia NFC Shell [5] into account. Through this it is possible to touch terminals with the Simplicity Device to establish a connection. This shortens the time for connecting the Simplicity Device with a terminal remarkably.

Acknowledgements

This work was performed in the framework of the Simplicity project [1] and of the SMS project, partly funded by the European Union. The authors wish to express their gratitude to the members of the Simplicity and SMS consortium for their valuable contributions.

References

1. The Simplicity project, http://www.ist-simplicity.org/
2. N. Blefari Melazzi, S. Salsano, G. Bartolomeo, F. Martire, E.Fischer, C. Meyer, C. Niedermeier, R. Seidl, E. Rukzio, E. Koutsoloukas, J. Papanis, I. S. Venieris: "The Simplicity System Architecture", 14th IST Summit, 19-23 June 2005, Dresden, Germany.
3. Security and Trust Services API for J2ME, http://www.jcp.org/aboutJava/communityprocess/final/jsr177/
4. Near Field Communication (NFC) Forum, http://www.nfc-forum.org
5. Nokia NFC Shell, http://europe.nokia.com/nokia/0,,76314,00.html
6. Chen, G.; Kotz, D. A Survey of Context-Aware Mobile Computing Research. Technical Report: TR2000-381Dartmouth College, 2000.
7. Abowd, G.; Dey, A.; Brown, P.; Davies, N.; Smith, M.; Steggles, P. Towards a Better Understanding of Context and Context-Awareness. In Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing, pp. 304-307, 1999.
8. 3rd Generation Partnership Project. Generic User Profile – architecture. Technical specification of Technical Specification Group Services and System Aspects, Version 6.3.0. 2004.
9. Hallam-Baker P. Client Side Automated Form Entry. W3C Working Draft, WD-form-filling-960416.
10. Shwetak N. Patel et al. A Gesture-based Authentication Scheme for Untrusted Public Terminals, http://www-static.cc.gatech.edu/~jpierce/papers/shake-uist04-camera-final.pdf
11. The Liberty Alliance Project, specifications https://www.projectliberty.org/resources/specifications.php
12. UAProf Schema, XXX
13. W3C XML Path technology (XPath), http://www.w3.org/TR/xpath
14. W3C XML Query Language (XQuery), http://www.w3.org/XML/Query/
15. JSR 82: Java APIs for Bluetooth, http://www.jcp.org/en/jsr/detail?id=82
16. The kXML Project home page, http://kxml.objectweb.org/
17. Atinav aveLink Technologies, http://www.avelink.com/bluetooth/index.htm
18. Avetana JSR-82 implementation, http://www.avetana-gmbh.de/avetana-gmbh/produkte/jsr82.eng.xml
19. BlueCove at SourgeForge, http://sourceforge.net/projects/bluecove/
20. 3rd Generation Partnership Project. Data Description Method (DDM) - 3GPP Generic User Profile (GUP). Technical specification of Technical Specification Group Terminals, Version 6.1.0. 2004.
21. Want, R.; Pering, T.; Danneels, G.; Kumar, M.; Sundar, M.; Light, J. The Personal Server: Changing the Way We Think about Ubiquitous Computing. In: Proceedings of the 4th international conference on Ubiquitous Computing (UbiComp '02), Göteborg, Sweden, 2002.
22. Ubiquitous Personal Server, http://www.intel.com/research/exploratory/personal_server.htm
23. Realm Systems, http://www.realmsys.com/