qWixP, a Qt-based, Wireless Information Exchange Platform

Ralf Salomon, Holger Harms, Frank Reichenbach, and Thomas Kirste
Faculty of Computer Science and Electrical Engineering
University of Rostock, 18051 Rostock, Germany
{ralf.salomon,holger.harms,frank.reichenbach,thomas.kirste}@uni-rostock.de

Abstract: Due to technological advances, wireless personal digital assistants (PDAs) are becoming quite popular. However, it seems as if none of the existing applications fully exploits their inherent potential. As a contribution to this (marketing) problem, this paper presents a Qt-based, wireless information exchange platform (QWIXP) that can be used as an integrated conference information system, for which wireless PDAs are very natural and cost-effective access tools. In addition to a brief overview of the application's architecture and functionality, this paper describes also the realized communication platform as well as parts of the graphical user interface. The communication platform allows for the usage of both wireless LAN and Bluetooth devices. With this tool, the user has full wireless access to all conference-relevant data.

1 Introduction

Personal digital assistants (PDAs) have been gaining increasing popularity during the last few years. Furthermore, widely available wireless technologies, such as wireless LAN (WLAN) [5] and Bluetooth [6, 4] add a new capability to mobile devices, for which the term *true mobility* seems appropriate. This gives rise to a growing number of application ideas that revolve around the concept of *context-aware guides*.

This paper presents a *mobile conference assistant* based on these technologies. Previous work on mobile visitor guides or conference assistants (such as Lancaster's Guide [1] or the Conference Assistant described by Dey et al. [2]) focuses on the fundamental *application concept* of mobile and context-aware guides, while relying on rather powerful devices, such as Notebooks and high-end PDAs, for the field trials. In contrast to this, the present system focuses on the *system- and network-level* issues that need to be considered when deploying such a system on ressource limited, moderately priced PDA devices in heterogenoeus wireless network environments. The authors believe that the following two aspects are essential for a wide-acceptance of such a system: good usability and an operating-system-independent implementation. The first aspect is important in that only rare opportunities for user training exist. The second aspect allows for both a cost-efficient distribution and an easy installation.

Section 2 argues that conferences inherently provide a setup in which an information sys-

tem might greatly support the attendees' needs for which wireless PDAs are the ideal tool for communication. Starting off from the conference scenario, Section 3 describes the *qWix* plattform also called *BlueWan* network.

Major components of the information system mentioned above have been designed and implemented in student projects. Section 4 presents the communication platform, which supports wireless access by means of both WLAN access points and Bluetooth gateways. Section 5 then describes parts of the graphical user interface. Section 6 reports on the current implementation status, and Section 7 finally concludes with a brief discussion. It also sketches future research projects including stability, performance, and security issues.

2 The Conference Scenario

Probably almost all readers of this paper have ever attended a conference. But it seems worth reviewing some of the conference peculiarities, in order to allow for a better understanding of some of the design decisions.

2.1 The Situation

In these days, most users register by sending an email or completing a WWW form. At the conference site, the attendee checks-in at the conference desk and gets the complete conference material, such as the conference program, the time table, meal tickets, banquet dinner tickets, a bag, some flyers, etc.

During the actual conference, then, the user has to coordinate a large number of different activities for which the relevant information can be divided into three different groups. The first group, *conference or organizational* information, subsumes everything that is relevant to the organization of the attendee's stay, such as getting room numbers and actually finding them in order to attend and/or to chair selected sessions and talks, information about when to give a talk where, which session to chair, and checking with the conference desk for getting up-to-date information on various changes in rooms, schedules, and the like.

The second group consists off all *content*-related information, such as author information, paper key words, related research, paper background material, and so forth. The third group collects all *personal* information "needs", such as checking with the conference desk for personal notes, meeting friends for lunch, dinner, etc, reading and writing email, ordering books and papers, downloading presentations, organizing discussion groups, using a means to distribute short messages, etc.

Depending on the size of the conference, these activities can consume a significant amount of time. Things might get even worse at huge conferences, which are typically distributed over several buildings, since it might be hard to meet friends, to find the right room etc. Thus, one might sometimes get the impression that unproductivly cruising around is the most important activity of everyone.

2.2 Users' "Needs"

Most conference organizers, of course, try to accommodate the users' "needs" by providing written documentation both before and during the conference. Most of them also provide bulletin boards and a small number of PCs with internet connection. Some might even be willing to run an information retrieval system.

Even though good in intention, the provided support is somewhat limited for obvious reasons. For example, printed material is often out-dated, bulletin boards are rather rare and small, it is still hard to meet someone in crowded areas (e.g., finding someone among 1000 attendees during a banquet dinner), and the number of PCs is small and thus constitute a bottleneck for internet access. Furthermore, the attendees have to consult their written material quite often, and they have to spend a significant amount of time for just polling some information system (polling vs. pushing information). Most attendees would also appreciate a personal, automatically update personal conference schedule.

Another issue concerns the given presentations. An attendee might find it desirable to browse keyword lists, to download a presentation, to obtain background material, or to receive links to related work (publications and demos). Hardly any conference organizer provides full only access to an integrated information plattform; Other systems [2] focus on context awareness, and, for example, provide access to additional multi-media content. This paper would thus like to argue that this paper addresses some additional attendees' "needs" that have not been in focus of other systems.

In such a "chaotic" conference situation, the attendees might be advised to use cellular phones in order to accommodate their needs. Unfortunately, this advise is of limited utility due to practical reasons. Mainly, international roaming service costs are generally high, bandwidth is rather small, the network reliability might be low, and incompatible radio schemes (GSM vs. CDMA) prohibit any communication.

As this paper argues, using mobile devices, such as PDAs and small laptops, in an installed wireless infrastructure might be another option. The infrastructure would consist of at least a server and some wireless LAN access points as well as Bluetooth gateways. This paper furthermore argues that the usability of PDAs is significantly better, since they are much smaller and lighter than laptops. On the other side, however, standard PDA software is currently not available for such applications.

2.3 Design Considerations

Targeting at currently available PDA technology imposes some severe design constraints. They normally have small memories in the range of about 32-128 MB. It is thus not possible, in contrast to laptops, to download all possible or potentially relevant information. In other words, PDAs are required to download the users' data on demand.

A straight forward solution would be to set up a web server and to access the data by means of HTML-pages, CGI-scripts, Java-scripts, and the like. Preliminary usability tests

(see also [7]), however, have clearly indicated that most test users did not like a web-based user interface, because the PDAs' displays are simply too small to display enough content, too many clicks were required, and the response time was too long. The first usability tests suggested that for a success, an integrated graphical user interface (GUI) with high usability is essential. Another reason for not using a web server is that the PDAs' batteries are of relatively small capacity. With respect to a long operation time, the wireless-transmitted data portions should be as small as possible; and web pages contain a lot of rendering data in addition to the actual content.

Because of the technological resources, the situation, and requirements presented above, the system employs a *central* server that stores all conference-relevant data in a relational data base. It is furthermore assumed that (1) the user first registers by sending an email or submitting a completed HTML-form, (2) the submitted data is then fed into a data base system, and (3) the attendee finally checks-in at the conference desk, where the custom-made application including some personal identification data as well as encryption and decryption keys are downloaded by means of some media, such as flash memory or IrDA. Since this paper presents a software application, the terms "attendee" and "user" are interchangeably used.

3 The BlueWan Network Architecture

As can be seen in Figure 1, the BlueWan network architecture consists of a central (mini) server and a local area network (Ethernet), which connects the server as well as some WLAN access points and Bluetooth gateways. By means of the latter, the users' PDAs (as well as notebooks, of course) can communicate with the central server.

As described earlier, the server's data base stores all conference as well as user-specific data, such as schedules, paper details, keyword lists, news data, attendees' affiliations, the PDAs' ethernet addresses, etc. In addition, the data base also hosts administrational information, such as the IP numbers of the gateways and access points. Furthermore, the central server also provides Internet access (not shown) to allow for normal Internet usage (primarily email and WWW).

In addition to the functionality discussed above, the system provides a chat function with which the attendees can be exchanging short messages. Since it might happen that a recipient is suddenly offline, the server has to temporarily store such messages. These messages would be forwarded, if the recipient goes online again (detected by a WLAN access point or Bluetooth gateway). Furthermore, any conference changes and announcements can be distributed with the very same mechanism.

The server and all clients access the database by means of graphical user interfaces. These interfaces allow the users to easily navigate the information provided. Because in this application, all user interfaces are implemented in Qt [10, 11], it can be installed on various operating systems, such as Windows, Linux, OS/2, etc, and also on embedded devices.

The network infrastructure as well as the server software is based on Linux, because Linux and most of its applications are free of charge and thus allows for development under GPL.

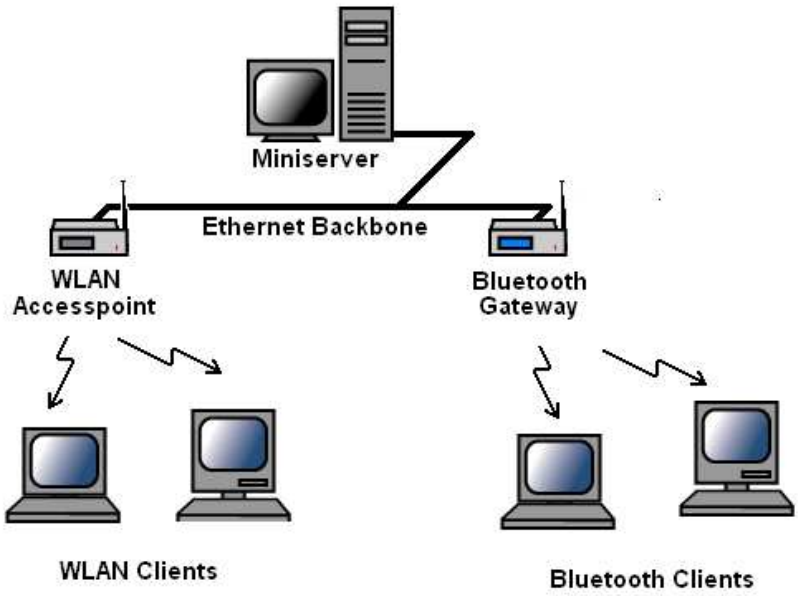

Figure 1: The infrastructure consists of at least one central server and several access points (wireless LAN access) as well as Bluetooth gateways.

4 Network Infrastructure

This section describes some important design and implementation issues. With these descriptions, it should be possible to either re-implement the system or contribute to the project. In so doing, this section presents some selected topics; for further details see [3].

4.1 Network Communication

It is quite obvious that the communication patterns are highly dynamic in the described conference scenario. For this setup, the communication backbone is based on Ethernet, and with respect to the communication protocol, the choice was on connectionless UDP, as opposed to connection-oriented TCP, for the following reasons:

- It might happen that one of the communication partners, i.e., an attendee, is offline or becomes unreachable for any reason during a data transmission.
- It might happen that the user travels between access points (or Bluetooth gateways, respectively) during a data transfer.
- Even though IPv6 [12] is able to realize hand-over (horizontal roaming), not all devices support IPv6 and is thus not further considered.

- It might be that messages cannot be delivered, which are thus temporarily stored on the central server for re-transmission at a later point in time. This approach is useful in this application, since the sender might wish to go offline after sending some information.
- If a connection-oriented transmission would be used, one Bluetooth gateway could only handle up to seven clients at a time, due to Bloothoth's piconet limitations.
- The application should support both WLAN and Bluetooth (see, also, the following subsection).

The choice of using UDP as the network protocol implies that all error handling, time outs, re-transmissions, etc. have to be done by actual application. In case a receiver goes offline during a data transmission, the server stores the data that have not been successfully transmitted are stored in the server's data base until the client goes online again.

4.2 Bluetooth Integration

Since the official (Linux) Bluetooth stack[1], BlueZ, does not support IP via the logical link and adaptation protocol (L2CAP) without installing a particular software package, dedicated gateways perform the protocol translation between UDP/IP and Bluetooth. In contrast to WLAN, where the 802.11 network interface card might be automatically sending to the access point with the best bit error rate (BER), Bluetooth requires the knowledge of the Bluetooth device address of the communicating gateway. For this purpose, the clients maintain a list of all system gateways and perform an inquiry scan from time to time. Since the Bluetooth client sends a `ping` to the gateway, the central server can use this information to appropriately route all packages to the client.

4.3 Network Administration

In addition to storing all conference-relevant data, the central server also administrates the network. To this end, the server distinguishes between three different types of data packages. By means of an internal `system` commands, the server is able to initialize, reset, configure, and query the system's WLAN access points and Bluetooth gateways.

Since the system features both WLAN and Bluetooth and relies on the UDP network protocol, "Bluetooth-routing" has to be done by the server. To this end, the server maintains the devices addresses of all known Bluetooth devices (i.e., PDAs and notebooks that have registered at the conference desk). It furthermore maintains the device's state as well as the Bluetooth gateway to which it is currently connected. For maintenance purposes, the server exchanges internal `control` messages with the wireless devices, and internally stores its state, i.e., `online`, `offline` and `unreachable`.

[1] kernel version $>$ 2.4.10, bluez kernel version $>$ 2.2, bluez libraries versions $>$ 2.3.

By means of these control commands, the communication partners also negotiate the MTU (maximal transfer unit), since it depends on several system characteristics. The system consequently splits larger data units into several smaller packages. These packages are sent one after the other, and upon arrival, receipts are sent to the sender. Data packages are re-send, if no receipt is received after a specified amount of time (positive acknowledgement with retransmission (PAR)).

By means of the third class of commands, the server exchanges data packages of various sorts with the users' wireless devices. These packages are either transparently routed via the WLAN access points or translated by the Bluetooth gateways.

The system furthermore provides a chat function that virtually appears as a peer-to-peer operation. For the various reasons discussed above, the communication has to be coordinated by the server. Because a user might remember a recipient's nickname easier than the hardware address of the recipient's wireless terminal, the server provides an address resolution protocol that maps users' nicknames onto the hardware address of users' wireless devices. To this end, the server also stores the latter two data items in the data base.

Finally, as has been mentioned above, the server also stores a limited number of data packets that could not be delivered to the recipient. This mechanism is also used to post up-to-date information and general announcements. The server simply adds this data to the list of not-delivered data packages.

5 Graphical User Interfaces

5.1 Data Types

From the user's point of view, the numerous data present in the system can be grouped as follows:

System: IP addresses of all access points and Bluetooth gateways, room and building locations, maps, etc.

Schedule: conference session schedule, breaks, lunch details, banquet dinner, etc.

Attendee: name, address, affiliation, nick name, PDA's MAC address, encryption/decryption key pairs, shopping cart (books, papers, etc.), location (for online users only), session chairs, and so forth.

Papers: presenter, proceeding details, key words, category, session, etc. background information, presentations, etc.

News: general announcements, changes of rooms and times, etc.

Chat: short messages and pings between online users.

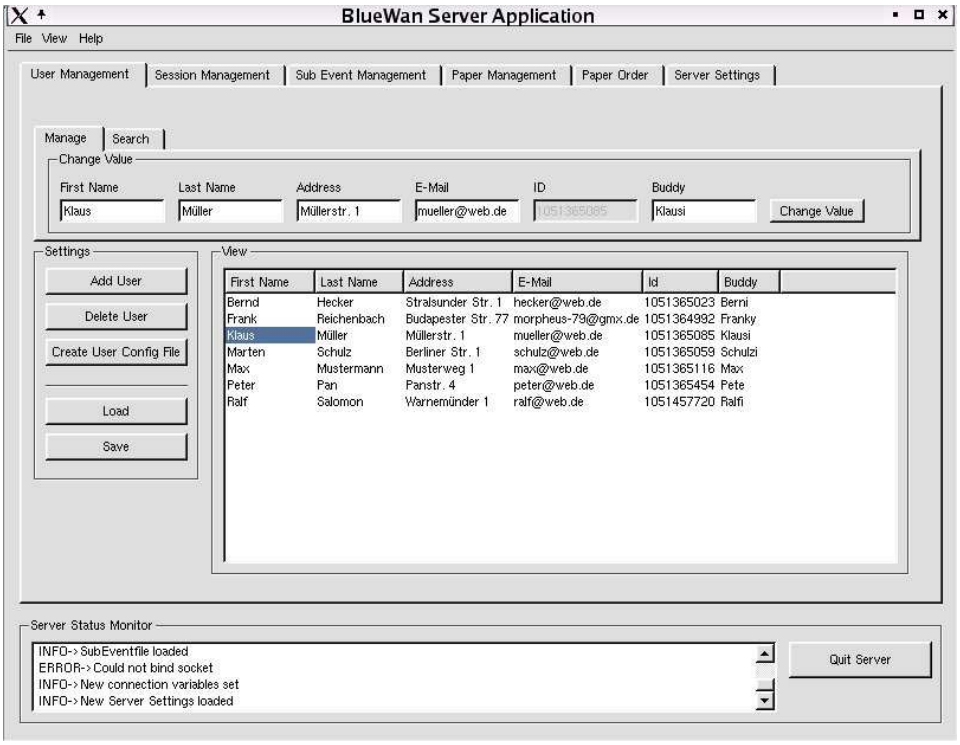

Figure 2: The system provides an extended GUI to maintain and update all data items of the server's data base. The access is organized into six different tabs.

5.2 General GUI Remarks

Since the system should be coherent and platform independent, it was implemented in Qt [10, 11]. Qt is a library that supports graphical user interfaces, internetworking, data base integration, and is running on various platforms, such as Windows, Linux, OS/2, etc. Furthermore, *embedded* Qt also supports the implementation on devices, such as PDAs and embedded systems. This approach allows for using modern terminals, such as HP's Compaqs and Sharp's Zauruses. The following two subsections present some of the server and client GUIs.

5.3 The Server

The server's data base stores *all* data present in the entire system (see also Subsection 5.1). The system provides a GUI for maintaining and updating all data items. As Figure 2 shows, access to the individual topics is ordered by six different tabs. Figure 3 shows a day view with several "sessions". The system also provides an interface to automati-

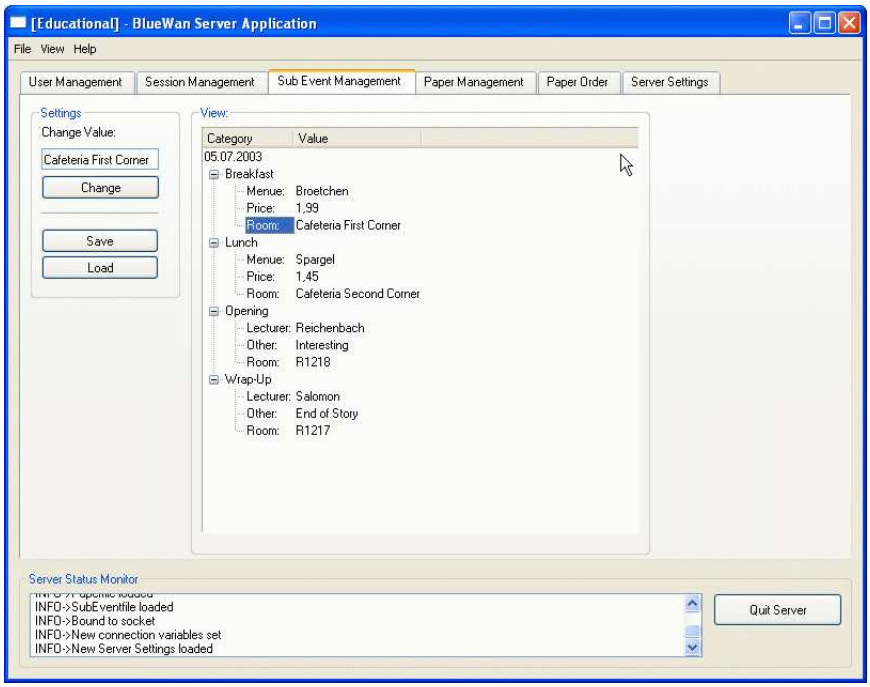

Figure 3: This figure shows how events/sessions are edited.

cally import and export external data, e.g., attendee's registrations, conference schedule, paper information, etc. Due to space limitations, further screenshots are not shown. The interested reader is referred to [8].

5.4 The Clients

As has already been mentioned, many PDAs feature only a very small display with typically 320 $\times$ 240 pixels. It is evident that presenting all information, which is present in the system and potentially relevant to the user, on such a small display is rather challenging, especially when usability engineering is of high importance. It should also be kept in mind that most of the data does not have a strong hierarchical nature; rather, most data items have many links to others, it appears as a web with a high connectivity. The system design thus also emphasizes on navigating the information offered.

The top client window provides the following six different tabs: `Events`, `Chat`, `Search Person`, `Order Papers`, `Client Settings`, and `Status Monitor`. At this level (see, also, Figure 4), each tab provides rather coarse information. To get more details, the user might be clicking into the various fields. For example, when clicking into a session (session 8:00-9:00 on the 5th of July in Figure 4), the user is presented with a session view

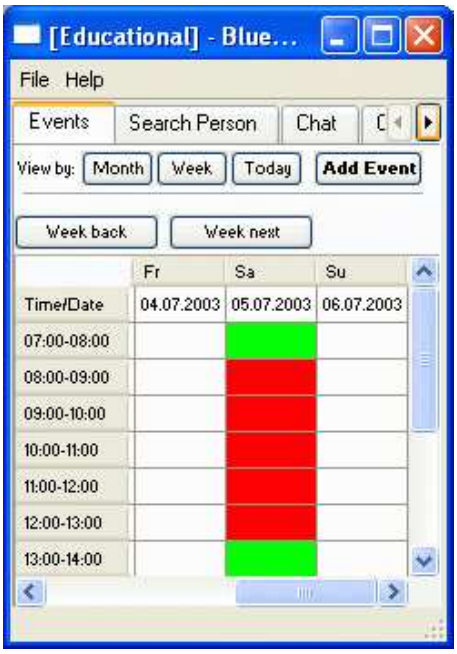

Figure 4: A conference week view.

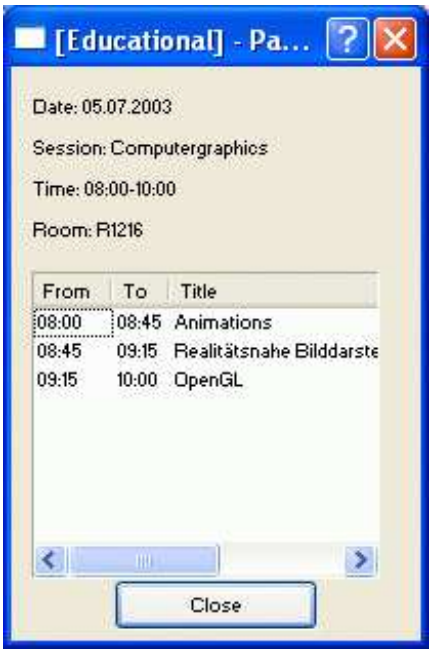

Figure 5: A particular session.

(Figure 5), which presents all papers of that particular session. By clicking into one of the papers, the user receives all paper details, such as author, title, author's affiliation, etc. (see, also, Figure 6). From there, the user can switch to the author, can get more paper details, such as the abstract, page numbers, keywords, and so on. In addition, Figure 7 shows the presentation of a compact day view. Further details can be found in [8].

6 Current Implementation Status

Major parts of the system have been implemented as parts of some student theses. The server has been realized on a Linux system, and for the data base, mySQL has been used. The current server GUIs allow for the administration of the system. Furthermore, the client GUIs have been implemented on PCs (for easier (usability) testing) as well as embedded Qt for the Compac and Zaurus PDAs. Finally, the network communication allows for accessing both WLAN and Bluetooth, which include the required Bluetooth gateways.

Figure 6: All details of a selected paper.

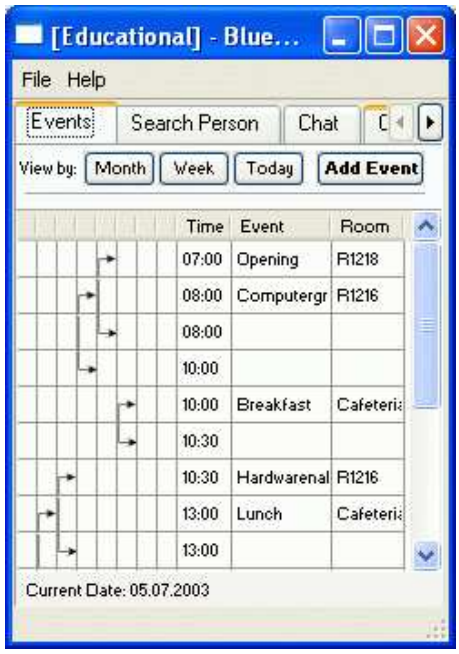

Figure 7: This figure shows a compact day view.

7 Future Research

This paper has presented an integrated conference information system that might be accessed by wireless personal digital assistants. Some main advantages of using wireless PDAs is their small size, light weight, and that the entire data transmission is free of charge (no telephone costs).

Future research will be devoted to the following three main aspects. The first aspect concerns performance issues (with respect to network transfers). The main aim is to implement a cache memory on the PDA in order to keep the required network transfers at a minimum. The second aspect concerns extending the systems availability by porting it to other operating systems; the PDA client software is implemented in Qt and currently runs on Pocket PC and Linux. The third issue concerns security. As has been explained, the user may also order books, demos, papers etc. The current systems tackles security issues by using a public key method [9]. As a side effect, encrypting messages with an asymmetric procedure requires significantly more computational power. Even though the usage of a public key method, such as PGP or RSA [9], seems reasonable, it should be proven that no security problem exists in this particular application. Therefore, this research avenue will be devoted to these two problems. It is furthermore planned, to conduct extensive usability studies for all of the aforementioned developments.

References

[1] Keith Cheverst, Nigel Davies, Keith Mitchell, Adrian Friday, 2000, *Experiences of Developing and Deploying a Context- Aware Tourist Guide: The GUIDE Project*, Proc. MOBICOM 2000.

[2] Anind K. Dey, Grogory Abowd, Daniel Salber, 2001, *A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications*. Human Computer Interaction 12(2):97–166.

[3] Holger Harms, 2003, *A Wireless Communication Platform for Personal Digital Assistants in Closed Networks,* student thesis, student's university, Microelectronics and Computer Science.

[4] David Kramer, Gordon McNutt, and Brian Sense, 2002 *Bluetooth Application Developer's Guide,* Syngress Media.

[5] Daniel Minoli, 2002, *Hotspot Networks: WiFi for Public Access Locations,* McGraw-Hill.

[6] Robert Morrow, 2002, *Bluetooth: Operation and Use,* McGraw-Hill Professional.

[7] Jakob Nielsen, 1993, *Usability Engineering*, Morgan Kaufmann, San Francisco.

[8] Frank Reichenbach, 2003, *Design studies and Usability Tests for a Client-Server Tools for Wireless Digital Personal Assistants,* student thesis, student's university. Microelectronics and Computer Science.

[9] William Stallings, 1995, *Network and Internetwork Security: Principles and Practice,* Prentice Hall, New Jersey.

[10] Trolltech, `www.trolltech.com`.

[11] Patrick Ward, 2000, *Qt 2 Programming for Linux and Windows 2000,* Prentice Hall, New Jersey.

[12] Herbert Wiese, 2002 *Das neue Internetprotokoll IPv6,* Hanser Verlag.