Mobile Home Automation – Merging Mobile Value Added Services and Home Automation Technologies

Andreas Rosendahl and J. Felix Hampe
Institute for IS Research
University of Koblenz-Landau, Germany
{rosi, hampe}@uni-koblenz.de

Goetz Botterweck
Lero – The Irish Software Engineering Research Centre
University of Limerick, Ireland
goetz.botterweck@lero.ie

Abstract

In this paper we study mobile home automation, a field that emerges from an integration of mobile application platforms and home automation technologies.

In a conceptual introduction we first illustrate the need for such applications by introducing a two-dimensional conceptual model of mobility. Subsequently we suggest an architecture and discuss different options of how a user might access a mobile home automation service and the controlled devices.

As another contribution we implemented a mobile home automation service, which we named REMOTILE. This prototype helps us to discuss typical components, such as modules that integrate various home automation devices.

1. Introduction

When people order a new car, they often invest a fair amount of money in electronic, comfort-enhancing accessories. Some extras, such as climate control, are already considered standard. For other extras, such as navigation or entertainment systems, there seem to be nearly no limits as to the money one can spend for additional functionality or gadgets.

If we compare this to the outfitting of a new house, it is rather unlikely that someone will pay a similar proportion of money to add comfort-enhancing electronics – especially if we see this in relation to the price of the whole building. Home automation systems are still only found in expensive, upscale real estate [1].

If we strive for an increased use of such facility automation and control systems, one important challenge lies in the way a user accesses their functionality. Besides the obvious goal of high usability, it is – at least in our opinion – also desirable to provide this access in an ubiquitous form that is available "everywhere".

In this paper we, therefore, study mobile home automation, a field that emerges from an integration of mobile application platforms and home automation technologies and deals with the ubiquitous access to home automation systems. We discuss some application scenarios which become feasible if we "connect" these two worlds, and elaborate on the challenges which arise when designing such applications. These include the following research questions:

- How can we deal with the inherent contradiction between (a) the complexity of the system that we want to control, i.e. the house with all its appliances and their function-sets, and (b) the **limited interaction capabilities** of mobile devices which we want to use as a user interface (UI) to control these systems?
- How should we structure our architecture, so that we can integrate a large **variety of back-end systems** and the mobile front-end can offer uniform and consistent access to their services?
- We cannot build *one* preconfigured standard application due to the variety of houses with varying building structures and facilities, etc. So how do we take into account that the application has to be **customized for each installation**?

To better understand these challenges, and as the source for a well-founded discussion, we designed and implemented a prototype application, REMOTILE, which we will demonstrate in the second half of this paper. This investigation and the discussion of application aspects are significant, because they allow us to better understand the related success factors and implications for potential providers of similar services.

First consider some of the electronic and electric appliances which can be installed in a house or flat, such as light, heating, home entertainment, motorized window blinds, telephone system or security devices. Now imagine that a user of our system could control all of this functionality remotely from her conventional mobile phone while she is at work or on the road.

Sixth International Conference on the Management of Mobile Business (ICMB 2007)
0-7695-2803-1/07 $25.00 © 2007 IEEE

For instance, if she is expecting an important private telephone call but has to work unexpected overtime in the office, she could **redirect phone calls** to her office number by remotely changing the configuration of the telephone installation at home. In addition, she could **program her digital video recorder (DVR)** to record her favourite TV series episode which she would miss otherwise.

Other application scenarios can be implemented based on **profiles**. For instance, imagine an "I'm leaving now" button at the door which turns off all the lights and lowers the heating. Before returning, the user could *remotely* initiate a "Coming home in 90 minutes" function which will switch heating back to comfort level, including the preparation of hot water necessary for a relaxing bath. Besides the obvious comfort enhancements, this promises significant energy savings by allowing remote adjustment of resource intensive appliances.

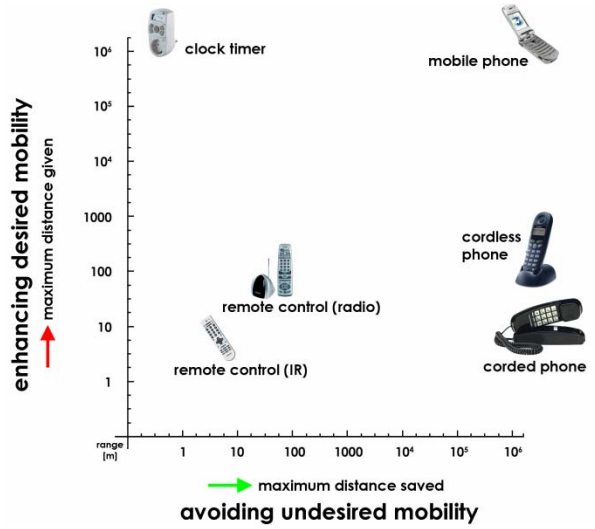

Figure 1. Different Forms of Mobility

2. Motivation

To illustrate the need for mobile home automation, we suggest a conceptual model that distinguishes two cases:

- In the case of **enforced undesired mobility** a person has to move somewhere else to perform a certain activity, for instance to talk to somebody who lives in another part of the city.
- In the case of **inhibited desired mobility** a person cannot go somewhere else because a task requires her to be at a certain location, for instance to switch the water sprinkler on and off.

Both cases can be improved by using appropriate technologies. We can illustrate this by a two-dimensional model (cf. Figure 1):

- **Undesired mobility** (horizontal dimension) **can be avoided** by using "far reaching" technologies instead of going "there", for example by using a remote control or by calling somebody by phone instead of visiting in person.
- **Desired mobility** (vertical dimension) **can be enhanced** if we use some mechanism to do the job "here" while we go somewhere else. An example is programming a timer to switch the water sprinkler on for 20 minutes or programming a DVR to record our favourite late night show. However, this only works if we carefully plan ahead. For instance, it is not possible to program the sprinkler or the DVR *after* we have already left home.

This situation can be overcome if we integrate home automation technologies and mobile networks with their (almost) global coverage.

The REMOTILE service we propose here should allow the mobile remote control of home appliances:

a. by every (legitimate) user
b. in every situation
c. from anywhere
d. at any time

To realize these requirements we first have to consider the whole communication chain from the controlled remote device to the user who wants to control it, and everything in between (cf. Figure 2).

2.1. User Interface

The proposed mobile service is designed to work with most of **today's standard mobile phones** and, hence, is *available for most users* (fulfilling requirement a.). However, this means that we have to pay careful attention to the **usability** [2]. Especially simplicity should be one of our main concerns [3].

Mobile phones are by design constructed to be used "on the road" *in almost any situation* (b.) due to their small size, low weight, and portable power source. Unfortunately, this also implies **limited interaction capabilities** as a result of the small screen and the low number of small keys.

2.2. Data Transfer

Mobile Networks enable access to our service *from almost anywhere* (c.). Although it would be tempting to use the higher bandwidth of **third generation mobile networks**, such as UMTS (Universal Mobile Telecommunication System), these advanced networks **do not have enough coverage yet**. Hence, we designed REMOTILE to work with the lower bandwidth offered

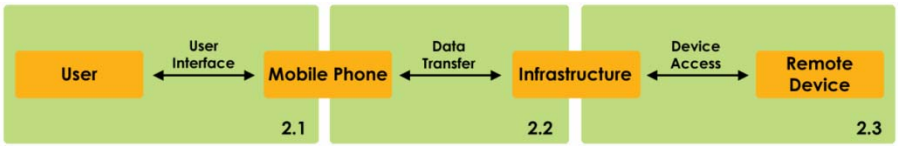

Figure 2. Data communication between the user and a remote device

by **GPRS** (General Packet Radio Service), which is available in most areas covered by GSM (Global System for Mobile Communications) and offers the advantages of packet-oriented data communication. As an implication we have to consider the **limited data transfer rate** and potential **latency** of several seconds.

Since we integrate our service with popular internet access technologies, such as DSL (Digital Subscriber Line) or broadband cable, which tend to come with flat rate contracts, the home server *can be reached at any time (d.)*.

2.3. Device Access

Even though mobile and fixed data networks fulfil all requirements for a remote administration, the higher hurdle is the access to the controlled devices. Here we can distinguish between several types of devices:

- Decentralised **facility busses**, such as KNX, EIB or LON [4] offer sophisticated control functions, but require large investments and are hard to install after a house or flat has been built. Their advantage lies in the large number of gateways which offer a considerable variety of access channels (cf. ③ in Figure 3). Besides the raw control, bidirectional systems allow access to the current state of a controlled device and feed this information back to the user.
- When considering the configuration and control of **telephony installations**, ISDN (Integrated Services Digital Network) seems to be a natural choice since it offers standardized protocols to directly and quickly manage services offered by the Intelligent Network (IN) [5]. The CAPI interface in combination with an ISDN controller offers a straightforward way to access the ISDN bus from a personal computer [6]. In the context of home automation, we are mainly interested in supplementary services such as various call forwarding services. Besides the configuration via the customer's telephone port, it is possible to access and configure these services from the side of the telephony service provider (cf. ④ in Figure 3).
- Many **standalone appliances** can be integrated, too. Some of them, such as standard entertainment electronics, can be integrated via their infrared remote functions [7]. Others, such as DVRs or security cameras, often come with an integrated network port and even a built-in mini web server.
- Many current **home theatre systems** are fully integrated PC systems themselves. On the one hand these computers can act as central hubs of a home automation service by controlling other appliances; on the other hand they contain software applications which should be controllable by our service. In this case we can avoid the use of external interfaces altogether (cf. ② in Figure 3). Currently key technologies, such as OSGi (Open Services Gateway Initiative) [8] are emerging, which enable and ease the integration and configuration of middleware and external devices.

3. Related Work

Applications which facilitate the remote control of home appliances are usually provided by the **manufacturers** of the related home control and automation hardware. In most cases these appear in the form of a small web server which is installed on the home server available for the particular system [9]. The user can observe and control the installed devices via these **web interfaces**. Some systems offer special **WAP** (Wireless Application Protocol) pages for remote control from mobile devices. If such systems offer dedicated control software, this is usually intended for use within the house, for instance via a Bluetooth-enabled PDA [10].

Similarly, software and telecom companies and research institutions are dealing with the topic of home automation and control. For research purposes, they have designed and built model houses where future living scenarios and their technological and social implications can be evaluated in practice. In addition, these installations serve as image-building and marketing platforms which enable the presentation of future products. Three major projects in Germany are mentioned as examples:

- **inHaus** (Innovation centre intelligent home) [11]
- **T-Com-Haus** (the T-Com home) [12]
- **Das Haus der Gegenwart** (Contemporary House), Bayerische Hausbau GmbH and Microsoft) [13]

A key area in such projects is the user interface. Besides several control interfaces via mouse, touch screen or IR remote control, such projects often include a PDA which allows the control of the house from within or externally.

4. Back-End Architecture and Concepts

Figure 3 shows an overview of the overall architecture of the proposed service, including a mobile client ①, a back-end installed in the user's house ②-⑤, and various options to integrate telephony or network service providers ⑥.

Based on this architecture we identified three scenarios, which are compared in Table 1.

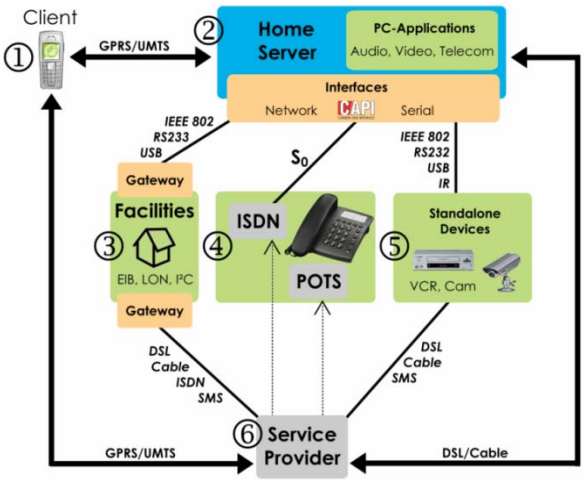

Figure 3. Integration of back-end systems

In the end, the type of realization largely depends on the prevailing conditions and the requirements of the users. If there is no home server or residential gateway, the provider has to find other channels to access and control the devices installed in the customer's house. For many telephony installations this is straightforward as they can often be configured directly by the telephone operator. Other devices come with a built-in network interface including a minimal web-server or other interfaces such as a GSM module.

With increasing complexity of features offered by the appliances, it makes sense to leave the installation and configuration to a specialist.

Similar to the communication scenarios shown in Table 1, there are different **locations** within the architecture **where we can implement intelligent control**, such as the processing of automation rules. For instance, such control logic could be placed in the home server, in the systems of a service provider, implemented in the controlled devices themselves or distributed over the whole system.

5. The Prototype from the Users Perspective

The mobile application which provides the UI for our proposed service was developed according to the **MIDP** (Mobile Information Device Profile) [14, 15], which is the most widespread open software platform for mobile devices.

When using the high-level API of MIDP, the form of presentation of interaction elements, such as menus and commands, is largely left to the device and its operating system. This means our custom-developed application is consistent with the native interface experience the user is accustomed to from applications which come built-in or preinstalled with the device. In addition, using the high-level API makes the application independent of particular devices [16].

	Via Home Server	Via Service Provider	Hybrid
Scenario characteristics	Client ① connects to the home server ② which communicates via the appropriate interfaces with the controllable devices ③ - ⑤ and returns information to the client.	Client ① connects to a service provider ⑥ which connects directly (not using the home server) to facility control busses ③, telephony services ④ and standalone appliances ⑤	Combination of the other two scenarios
Advantages	Home server can be used to provide multimedia functionality System remains under the user's control (privacy, security) Can integrate devices which can be connected to a PC, but have no network port	Can be offered as a professional service Bundling of efforts and devices (lower power consumption, maintenance more efficient)	Many options to integrate devices System provider can offer advice, installation and maintenance
Disadvantages	Home server is an additional device (power consumption, maintenance) More user know-how necessary	Devices require external interface, e.g. PBX that can remotely be configured over ISDN or integrated network port Higher complexity of the devices (costs, maintenance)	Complexity of the overall system

Table 1. Comparison of scenarios

0-7695-2803-1/07 $25.00

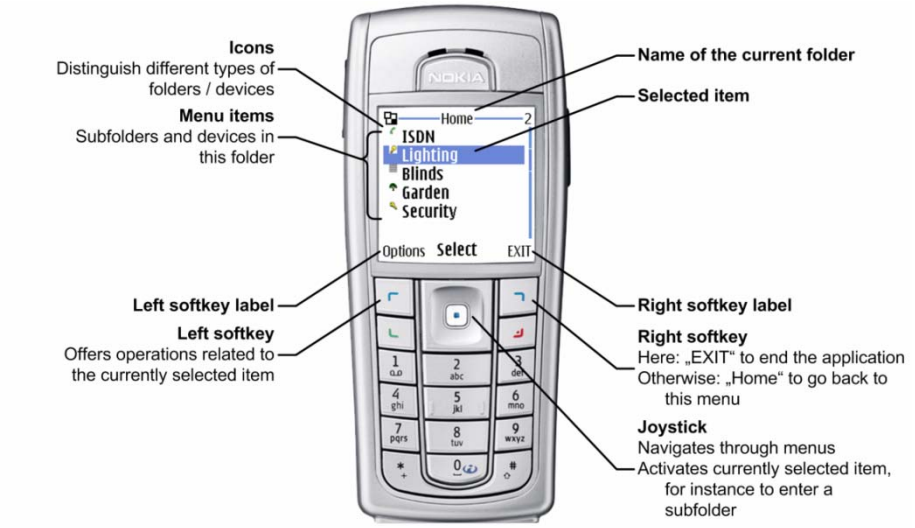

Figure 4. The user interface showing the main menu

Another principle followed during the design of our application was the precedence of **simplicity over functionality** [3].

In the following sections we give an overview of the application front-end by demonstrating it on a phone with the popular Nokia Series40 user interface [17].

5.1. Main Menu

The main menu (Figure 4) displays the first level of a **hierarchy** which represents the controllable services and devices structured according to functional groups or personal preferences of the user. This menu structure is defined by an XML-based configuration file which can be updated by OTA (over-the-air) mechanisms. This facilitates a user-friendly personalization or a later extension or update of the menu.

Because the personalized hierarchy reflects a well-known structure (e.g. the floor plan in the user's home), the navigation stays intuitive despite the limited interaction capabilities of the small device.

The concrete presentation of possible commands depends on the J2ME implementation on the particular device. For instance, on the Nokia phone shown in Figure 4 the commands are available via softkeys (labelled "Options" and "EXIT" or "Back").

Figure 5. Entering the "lighting" sub folder

If the user selects a subfolder item, the application descends into that subfolder (Figure 5). In contrast to the functional grouping of the main menu, these sub-menus are structured by locations in and around the house.

5.2. Device Modules

When the user navigates to a leaf of the tree hierarchy, then each menu item represents a device module. The UI of a device module consists of selection and input fields and, hence, is clearly distinguishable from the navigation screens.

A device module fulfils two functions. Firstly, it displays the **current state** of a device, showing for instance the current position of the window blinds in the living room. Secondly, it allows the specification of a **desired target state**, for example by entering that the blinds should be half closed.

For our prototype we developed some device modules for different tasks (cf. **Table 2**) which we will discuss in the following sections.

Function	Application Sample	Device Name
Switching (on/off)	Lighting, windows, doors, alarms, profiles	SwitchDevice
Range	Dimmer, window blinds	AnalogDevice, BlindDevice
Telephone	ISDN	ISDNDevice
Imaging	Security camera	ImageDevice

Table 2. Overview of device modules

0-7695-2803-1/07 $25.00

5.2.1. Switching: SwitchDevice. This can be used to control simple devices (Figure 6) or to activate predefined profiles.

Figure 6. A switching device, here lighting in the bedroom

The UI for a SwitchDevice could have been implemented with just one checkbox. However, we decided to use **two separate interaction elements** instead, to be able to label the two different states with icons and application-specific text, such as "profile activated".

We have chosen checkboxes over radio buttons, since this **allows us to indicate an unknown** state by leaving both checkboxes empty (if we did not request the current state from the device so far). When the user is entering a requested state, the interaction elements behave like radio buttons in order to avoid ambiguous input, i.e. they allow only one option to be checked.

5.2.2. Range: AnalogDevice. An AnalogDevice extends SwitchDevice to integrate devices which can be controlled within a certain range, for instance window blinds or a lamp that can be dimmed.

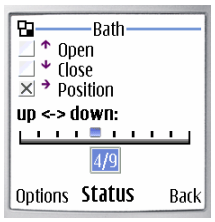

Figure 7. Controlling a range value with an AnalogDevice, here motorized blinds

For instance, in addition to completely opening and closing window blinds (see Figure 7) you may also partially close them.

5.2.3. Call forwarding: ISDNDevice. The configuration of call forwarding in modern phone networks, such as ISDN or GSM, is implemented by the ISDNDevice module (cf. Figure 8).

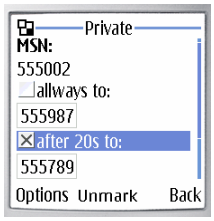

Figure 8. Configuring a telephone installation with an ISDNDevice

In the menu hierarchy there is one device for each logical telephone port, as identified by the MSN (Multiple Subscriber Number). In addition to this each telephone port is labelled with a name (e.g. "Private" in Figure 8) to avoid confusion.

There are three types of call forwarding: all-calls, no-answer, and on-busy. Each of these can be parameterized with a destination number the system should forward the call to.

5.2.4. Imaging: ImageDevice. The module ImageDevice offers access to still images taken by security cameras or other imaging devices (Figure 9).

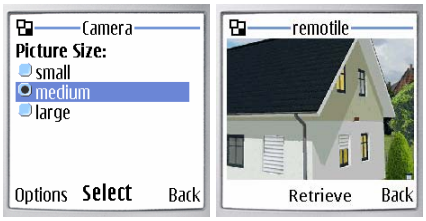

Figure 9. Accessing surveillance cameras with an ImageDevice

After the selection of the device, the user can specify whether she wants to retrieve a small, medium or large picture (left screenshot in Figure 9). A "small" image is equivalent to the native display size of the MIDP screen of the mobile device. In other cases the user has to scroll by using the cursor cross or joystick.

5.3. Execution of Selected Functions / Changes

When designing the front-end, we decided that interactions with the UI **should have no immediate effects on the physical devices** which are controlled. The appropriate commands are sent only when the user explicitly executes the intended functions:

Within each device module there are interaction elements to view the current state or to edit a desired state. In addition there are abstract commands to retrieve the current state of a physical device ("Status") or to execute the transmission of a desired state to the physical device ("Execute"). For a "read-only" device, such as the ImageDevice, the user can only retrieve the current status.

Figure 10. Retrieving the status and executing commands

Since the basic structure of this part of the UI is the same for all device modules it is easy to learn and understand. Furthermore separating the execution of commands is advantageous in another way: These are the operations that are causing traffic over the mobile network. Hence, it makes sense to keep them separated from other interactions to avoid unexpected costs for transmitted data.

To deal with the delay caused by typical mobile networks, the interface informs the user that the operation is in progress (Figure 11 left).

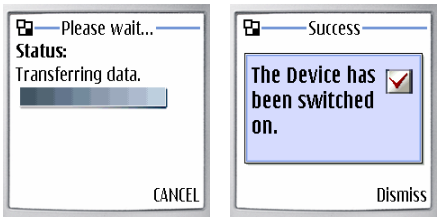

Figure 11. Transmission of commands and feedback

Additionally, this screen offers the option of cancelling the ongoing operation. After the transmission the interface either displays the result of the operation or an error message. Successful results and error conditions are presented with a strong visual difference, so that they can be easily distinguished (Figure 11 right).The result message has to be actively acknowledged by the user ("Dismiss" in Figure 11). An inattentive user may otherwise miss the result message which is displayed for just a few seconds.

6. Outlook on Possible Enhancements

During the design of the front-end presented in the preceding sections, we developed and applied a number of **UI principles** and patterns. Future versions and enhancements should follow these guidelines.

One option is to implement similar menus and device modules on **other mobile platforms**, such as Symbian or Windows Mobile. In that case the implementation should again aim at using abstract APIs to support a device-independent implementation.

In addition to the modules already implemented and presented above, one can think of many **additional device modules**, such as implementing an intelligent temperature control or the programming of a digital video recorder (DVR).

Another opportunity for enhancement is the configuration of the system. This could be handled by the user with a graphical front-end on a PC or the system integrator who installs the physical devices could configure the system either on-site or remotely from a service centre as an additional value-added service.

At present, the front-end application is designed for **remote usage**, i.e. controlling devices **from far away.** This scenario includes certain assumptions which imply design decisions incorporated in the front-end in its current state. We could extend the application scenarios to local (in-house) usage by integrating faster local communication technologies, such as Bluetooth and wireless LAN.

If we would switch to local usage (or offer it as an additional mode), i.e. controlling devices from within the house, we would clearly have to consider scenarios with visual contact between the user and the controlled physical device; for example, it could make sense to remove the additional step necessary to execute a command (as the status is then obvious). Hence, future versions of the mobile front-end **should be context-sensitive** and take the current **situation and location** of the user into account and adapt accordingly.

For a manufacturer who wants to sell his devices and home appliances, having (local or mobile) controllability could be the decisive difference to the competitor. Interestingly a lot of appliances already come with some interface which enables remote control. The main challenge is the technical integration and design of a consistent and usable interface across all devices.

Within our project we strive for an implementation of the server-component as an **add-in for common DSL or cable routers** which are already widely available and provide the interfaces and computing power necessary to realize home automation applications.

Besides the control of home appliances and telephony installations there are even some other services which might profit from the application platform presented in this paper. It would take another paper to present the ideas on m-health, m-government or m-business applications.

In any case the system has to offer usability as well as a high-level of **confidence** regarding the **reliability** of every executed operation. If the customer does not feel confident that the service is reliable ("Did I switch the sprinkler off now or do I have to press that button again?"), she will rapidly reject the system as either unusable or not trustworthy.

7. Summary and Conclusion

In the introduction we illustrated the need for mobile home automation services by introducing a two-dimensional conceptual model of mobility. Subsequently we discussed different architectural options for such services.

We then introduced REMOTILE, a prototype service, to demonstrate how an actual implementation of a mobile home automation system might look, especially from a user's perspective. The REMOTILE system addresses some of the challenges we identified in Section 1:

- Despite the limitations imposed by the limited interaction capabilities of mobile devices, REMOTILE offers a usable interface for a home automation system.
- The system's architecture allows the integration of a large variety of back-end systems which can be controlled over a uniform, consistent front-end.
- The system can easily be extended to consider more sophisticated types of devices or to provide a richer user experience.
- It can be configured to a customer-specific logical structure, for instance to reflect a building's floor plan.

In the near future another market push for such systems could come from the emergence of home automation systems which are based on wireless communication between a base station and any controlled devices (actors and sensors) [19], [20]. These wireless home automation systems are more suitable for legacy environments which do not provide any reasonable way of installing cable-based bus systems.

Preliminary evaluations with test users seem to confirm that the system is usually understood quickly and without prior instructions. Especially the hierarchical menu, which reflects a well-known logical structure (functional groups and rooms) seems to help here. Only at the point where they have to explicitly send the specified changes (see Figure 10) do the users often hesitate for a few seconds, since this interaction pattern seems to be unfamiliar to them. After an encouragement from the guiding researcher they usually go on to use the system without further difficulties.

Based on experience gained so far with this prototype, it is our opinion that **simplicity, and ease of use – but not feature richness – will determine the acceptance** of home automation services. Otherwise those services will be restricted to enter only the homes of a few early adopters and technology enthusiasts and, hence, never reach the critical mass necessary for a real market breakthrough.

References

[1] R. Staub and R. Senn, "Electronic-Home Report: Passive Ausrüstung von Wohnräumen – Heute vorbereiten," *HOMEelectronic*, vol. 2003, 2003

[2] E. Bergman, "Information Appliances and Beyond – Interaction Design for Consumer Products." London, UK: Academic Press, 2000

[3] C. Bloch and A. Wagner, *MIDP 2.0 Style Guide for the Java 2 Platform, Micro Edition*. Reading, MA, USA: Addison-Wesley, 2003

[4] A. Kell and P. Colebrook, "Offene Systeme für die Gebäudeautomation: LonWorks und KNX im Vergleich," i&I limited, Watford, UK 2004

[5] P. Bocker, *ISDN – Digitale Netze für Sprach-, Text-, Daten-, Video- und Multimediakommunikation*, 4 ed. Berlin: Springer-Verlag, Germany, 1997

[6] A. Badach, K. Merz, and S. Müller, *ISDN und CAPI – Grundlagen der Programmierung von ISDN-Anwendungen auf dem PC*. Berlin: vde verlag, 1994

[7] C. Bartelmus and K. Scheibler, "LIRC – Linux Infrared Remote Control," 2006, [cited 2006-12-12]

[8] R. Sietmann, "APIs für das intelligente Heim," *c't*, vol. 2003, pp. 64, 2003

[9] M. Fromm-Wittenberg, "EIB-Userclub Jahrestagung 2004 – Workshop," Giersiepen GmbH & Co. KG, Radevormwald, Germany 2004

[10] H. Jung, "KNX/EIB Bluetooth-Gateway." Schalksmühle, Germany: Albrecht Jung GmbH, 2004

[11] K. Scherer, "inHaus: Innovationszentrum für intelligente Haussysteme Duisburg," Fraunhofer IMS, 2004

[12] Deutsche Telekom, "Das T-Com Haus Berlin." Bonn: Deutsche Telekom AG, 2005

[13] H. Lösch, "Das Haus der Gegenwart." München: Schörghuber Unternehmensgruppe, München, 2005

[14] JSR 37 Expert Group, "Mobile Information Device Profile (JSR-37) – JCP Specification 1.0a, Sun Microsystems Inc. and Motorola Inc," 2000

[15] JSR 118 Expert Group, "Mobile Information Device Profile for JavaTM 2 Micro Edition – Version 2.0, Sun Microsystems Inc. and Motorola Inc.," 2002

[16] K. Topley, *J2ME in a nutshell : a desktop quick reference*. Cambridge, MA, USA: O'Reilly, 2002

[17] Nokia, "Nokia Series 40 Developer Platform," 2006