User-Centred Tool Support — Developing User Interfaces for Persons with Special Needs

Jochen Frey
DFKI GmbH
Stuhlsatzenhausweg 3
Saarbrücken, Germany
jochen.frey@dfki.de

Thomas Gard
DFKI GmbH
Stuhlsatzenhausweg 3
Saarbrücken, Germany
thomas.gard@dfki.de

Jan Alexandersson
DFKI GmbH
Stuhlsatzenhausweg 3
Saarbrücken, Germany
jan.alexandersson@dfki.de

ABSTRACT

Developing user interfaces is always a challenging task, particularly if the users are elderly or people with special needs. In this paper we summarize the experiences we made during the i2home project regarding the design and development of accessible user interfaces. On the basis of the industry standard URC a middleware platform has been implemented and the concept of pluggable user interfaces has been introduced. We give an overview of this approach and point to existing tools and future directions that can support developers and end users of smart home applications.

Keywords

ambient assisted living, universal remote console, instrumented environments, middleware, tool support

1. INTRODUCTION

Developing user interfaces (UIs) that are both advanced enough to satisfy the needs of experts, while remaining simple for other users is not an easy task. Worldwide, considerable efforts are invested in research and development for accessible and user-friendly technology for the sake of coping with the demographic change. In Europe, the European Union alone or possibly in combination with domestic initiatives, e.g., the Ambient Assisted Living (AAL) Joint Programme, fund projects that target different aspect on living or being as a person with special needs. A big part of these efforts include providing different users with intuitive and accessible user interfaces for interacting with appliances and service in the smart home. While there are many approaches that are concerned with providing sophisticated methods for discovery, controlling and communicating in smart environments, user interfaces still have to be developed separately for each controller platform or are neither intuitive nor easy to understand [13].

In the i2home project[1] we have developed user interfaces for persons with special needs within a smart home scenario. Here, four heterogeneous demanding user groups were involved: elderly, persons with Alzheimer's disease, partially-sighted and blind persons and finally, young persons with mild cognitive impairments. We used a combination of persona [3] and user-centred design (UCD) [1] methodology for modelling users and their needs. In fact, the user's requirement was the driving force for the development of scenarios and finally the user interfaces.

Standard UCD includes typical design activities including producing and testing with paper prototypes and other non-functional mockups which have the disadvantage that the user has to imagine the effect and the interaction concepts behind the envisioned UIs. This becomes particularly problematic when the users are not able to comprehend the functioning and effects of their actions. Furthermore, since we make a clear split between tasks—technical developers are not working directly with the end users. Therefore, the prototypes cannot even be adjusted during the early user tests since the user experts lack the ability and/or tools.

In this paper we describe our efforts in providing the different participants within such projects with tools that overcome some of these problems. We illustrate our arguments with two such design concepts. The first is a tool for rapid prototyping which support participatory design, that is, the user expert and the user can jointly implement and modify a functional UI on the fly. The second is the concept of virtual reality which grounds the actions of users via their user interfaces in a virtual yet realistic environment.

This paper is structured as follows: The next section presents our research methodology, user-centred design and points to some of the problems with its implementation, particularly for our end users. In section 3 we introduce the technical framework by means of the Universal Remote Console (URC) [6] in the combination with Task-Based UIs [8] providing us with a versatile platform for developing smart environments based on appliances and task models. We explain how an abstract UI description can be plugged by means of rendering a concrete UI. Furthermore we show that providing a standard model is not sufficient for achieving a broad acceptance of the technology. One important aspect in disseminating technology is creating an ecosystem that offers a continuous support at all levels of the development process. In section 4 we present the two design concepts based on virtual reality and a tool for rapid prototyping. Finally, we conclude the paper in section 5 and lay out some of our future research agenda.

2. USER-CENTRED DESIGN

By pursuing a user-centred design (UCD) approach, e.g., ISO 13407, the interaction is completely based on real user needs meaning that different user types are the driving force for the development of technology. A key ingredient at this point is Cooper's persona approach [3] describing different

[1] www.i2home.org

MobileHCI 2010 September 7-10, 2010, Lisboa, Portugal.
ACM 978-1-60558-835-3/10/09.

users in various stages of the development process. On the basis of this approach several general descriptions and scenarios have been constructed bearing the typical characteristics of interviewed participants. The UIs implemented for the users are evaluated and the results and experiences are fed into the requirements phase again. In i2home we divided the parties concerned into technical developers and user experts, which are responsible for defining the user requirements. A commonly used approach for capturing usability issues early in the design process is providing paper, mockup and finally functional prototypes, see figure 1.

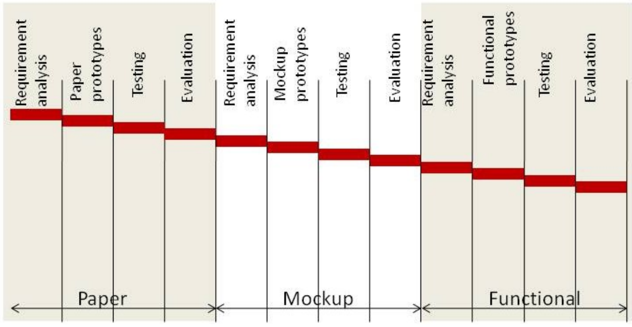

Figure 1: User-Centred Design with its traditional phases: paper, mockup and functional prototypes.

Applying this approach in i2home, it turned out that the use of paper-prototypes is hard for persons with special needs, e.g., cognitive or sensory impairments, since they often lack the ability to comprehend non-functional interfaces. Furthermore, the possibility to flexibly adopt actual results of user tests directly within the prototype is missing. Persons working with the users could benefit from being able to adjust the user interface interactively on the fly.

3. UNIVERSAL REMOTE CONSOLE

The Universal Remote Console (URC) framework [6], see figure 2, connects any networked device or service with any controller. The first project in Europe using URC technology was i2home, which had the ambitious effort to inject an ecosystem around the industrial URC standard and to introduce URC technology in the field of AAL. Currently more than 100 organizations and companies in Europe are using or working with i2home and/or URC technology. The URC standard provides an architecture called *pluggable user interfaces* that allows for interfacing arbitrary networked appliances or services with—in principle—any user interfaces. This allows for personalized and, perhaps most important in the context of AAL, accessible UIs. The architecture is based on the concept of a Universal Control Hub (UCH) [12] which is a gateway-based architecture implementing the URC standard thus managing the communication between controllers and targets:

- i) a *Controller* is any device for rendering the abstract UI, e.g., TV, touch screen or smartphone
- ii) a *Target* is any networked device or service intended to be controlled or monitored, e.g., kitchen appliance, home entertainment or eHealth devices
- iii) a *Resource Server* is a global service for sharing UIs and various resources necessary for interacting with the targets

The benefit of this approach is that it is possible to deploy consistent and, particularly, accessible interfaces which are tailored to the users and their specific needs. Currently, a pilot resource server is being operated by dot UI[2]. The UCH architecture mainly consists of three layers:

- i) the UI Protocol Layer is responsible for communicating with controllers
- ii) the UI Socket Layer defines the standardized contract between controllers and targets
- iii) the Target Adapter Layer manages the grounding and communication with actual targets

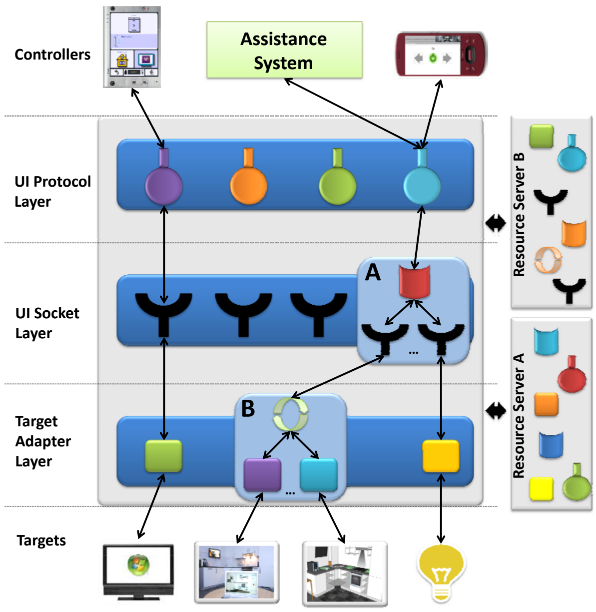

Figure 2: The layered URC framework allows any target device to be accessed and manipulated by any controller device. The figure shows two additional plugin extensions: A) Task-Model Engine based on CEA-2018; B) Synchronization Module

Contract between User Interface and Target

One of the key features of the URC framework is the standardized definition of the UI socket layer between the back-end devices (targets) and the front-end UIs (controllers), see Fig. 2. This *user interface socket* describes the input and output behavior of the appliance on an abstract level. The socket is then rendered on some controller thus giving the abstract UI a concrete implementation, or, in other words: plugging the socket [11, 5]. The concrete UI would connect to one or multiple sockets in two directions: first, getting the values that reflect the current state of the target, and second, requesting changes in the target's state through variable changes and function calls. On the back-end side each

[2] www.dotui.com

target appliance is represented by a dedicated target adapter that is responsible for the grounding of abstract socket elements with any specific network protocol. Clearly, this architecture offers a flexible way of connecting different UIs with any user interface socket and therefore with any connected target application. Multiple controllers and targets can be attached, detached and exchanged at runtime.

Task-based UIs

In the implementation developed within the i2home project, a runtime task-model engine based on [2] has been integrated, see [8]. This allows for extending the functionality of the platform with task-based UIs. Such a UI is also represented as a user interface socket in the platform (see Module A in Fig. 2). The Task-Model Engine allows for the implementation of home automation and task-based scenarios, e.g., leaving home or preparing a meal which includes interaction with targets integrated into the platform. Several home automation and task based scenarios have been implemented, tested and evaluated with real users [7].

4. TOOL SUPPORT FOR DEVELOPING UIS FOR USERS WITH SPECIAL NEEDS

While using URC technology during the i2home project it has become clear that an appropriate tool support is indispensable for developers of intelligent environments and assistant systems. Starting with the integration of devices and services into the middleware platform the developers first have to model the corresponding functionalities of appliances, i.e., the intended input and output signals. As mentioned in section 3 in the URC context this is done in an abstract way by means of writing UI socket descriptions. Furthermore there is the possibility to define additional resource descriptions, e.g., labels, icons and semantic roles. Writing all these resources manually is both tedious and error-prone. Secondly, by following a user-centred design approach in i2home we found out that particularly for people with cognitive disabilities it is more difficult to contribute to typical participatory design activities, such as sketching and paper mockups. A third problem is the setting of the testing environment. It is not possible to have always a fully instrumented environment when evaluating intelligent systems with the appropriate user groups. For the users it is important to visualize the effects of their interaction with the environment in a way that the need of abstraction is reduced to a minimum.

4.1 Interactive Rapid Prototyping

TESA is a graphical user interface builder that enables rapid prototyping of graphical user interfaces (GUI). It allows for rendering sockets on the fly. Figure 3 shows the workbench. The left side depicts the abstract socket elements which can be combined with the graphical objects from the lower tool bar via drag-and-drop. The content of the lower tool bar contains an extendable library of GUI objects. TESA immediately generates a fully functional user interface which can be parameterized by the settings part on the right side. TESA is very easy to use since it does not require any or only very shallow technical knowledge. For developers, TESA serves as a reference tool for testing their URC components and as an interactive mockup tool for the initial design of the graphical user interface.

True Participatory Design

Participatory design [9] involves the end users directly in the design and decision-making process. Understanding the needs of people with cognitive disabilities has the potential to provide unique insights [4] and is an essential ingredient when designing user interfaces. Rather than following the typical design activities, including producing and testing with paper prototypes and other non-functional mockups, with TESA, it is also possible for the user experts to modify and design the functional GUI directly together with the end users thus allowing for true participatory design.

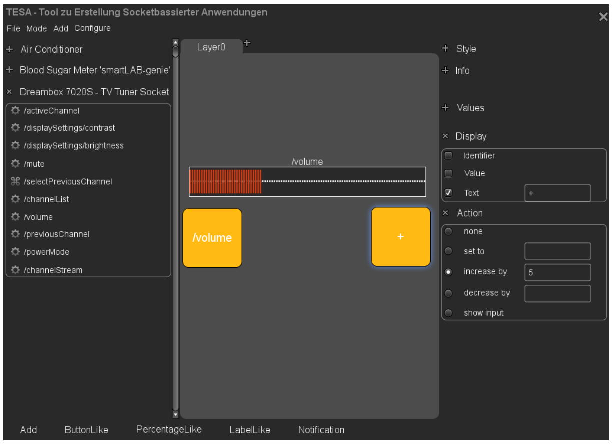

Figure 3: An example of a prototypical graphical user interface for controlling the volume settings of a television device. The left column depicts the abstract socket elements for a television device. The user can combine these elements with the predefined set of graphical objects on the bottom line via drag and drop, thus creating a fully functional user interface.

Given the functional prototype in figure 3 the normal behaviour would be that the volume is increased after pressing the + button. Working with cognitively impaired people during the i2home project it turned out that for instance some people want to press a button twice or hold it for a certain period of time before the corresponding functionality is triggered. Such preferences can only be found out if the users work with a functional user interface and thus have to possibility to experience the effects of their actions directly.

4.2 Virtual Reality

In early development phases a fully instrumented testing environment is often missing. This becomes particularly problematic when the users are not able to comprehend the functioning and effects of their interaction with a UI for controlling the intelligent environment. Virtual reality provides developers and end users an appropriate testing and debugging environment since it offers the possibility to visualize the effects of the user actions and therefore reduces the need of abstraction. Furthermore, the concept of spatial modeling can be used to address typical design issues, such as the choice and the placement of sensors and actors for human environment interaction [10]. Figure 4 depicts two visualization toolkits which have been integrated into the URC ecosystem, i.e., graphical models can be combined with ab-

Figure 4: Two existing visualization tools integrated into the URC platform. The graphical editor LONG on the left side shows a model of a completely furnished flat whereas the map modeling toolkit YAMAMOTO on the right side shows a virtual model of DFKI's intelligent kitchen.

stract socket elements and thus are capable to visualize the effects of any user interaction.

5. CONCLUSION

We have presented a middleware approach based on the industry standard URC. The platform itself provides a model-based approach in the sense that it represents target devices and service as abstract socket descriptions. We introduced the concept of pluggable user interfaces, i.e., the sockets can be plugged and thereby rendered to concrete user interfaces. Furthermore we have demonstrated tools and future directions for supporting participatory design, development and testing of URC technology. While the proposed tooling landscape will be finalized and extended in the future, further work will also be dedicated to investigate the concept of automatic UI generation, for instance by integration semantic, user models and context information into the URC ecosystem.

6. ACKNOWLEDGMENTS

This work has been partially funded by the European Commission under the grant FP6-033502 (i2home). The opinions herein are those of the author and not necessarily those of the funding agency.

7. ADDITIONAL AUTHORS

Additional authors: Christian H. Schulz (DFKI GmbH, email: `chschulz@dfki.de`).

8. REFERENCES

[1] Being human: Human-computer interaction in the year 2020. http://research.microsoft.com/hci2020/, 2007.

[2] CEA. *CEA 2018 (ANSI): Task Model Description.* Consumer Electronic Assiciation (CEA), 2008.

[3] A. Cooper. *The Inmates are running the Asylum : [why high-tech products driveus crazy and how to restore the sanity].* SAMS, Indianapolis, Ind., 1. print. edition, 1999.

[4] G. Fischer and J. F. Sullivan. Human-centered public transportation systems for persons with cognitive disabilities - challenges and insights for participatory design. In *in 7th Participatory Design Conference*, pages 194–198, 2002.

[5] J. Frey, C. Schulz, R. Nesselrath, V. Stein, and J. Alexandersson. Towards Pluggable User Interfaces for People with Cognitive Disabilities. In *Proceedings of the 3rd International Conference on Health Informatics (HEALTHINF)*, January 2010.

[6] ISO. *ISO/IEC 24752: Information Technology — User Interfaces — Universal remote console — 5 parts.* "International Organization for Standardization", 2008.

[7] M. Klíma, U. Diaz, A. Etxaniz, E. Urdaneta, J. Görlich, M. Dubielzig, and M. Lindemann. i2home Deliverable D8.4—Final Evaluation Report, 2010.

[8] C. Rich. Building task-based user interfaces with ansi/cea-2018. *Computer*, 42(8):20–27, 2009.

[9] D. Schuler and A. Namioka. *Participatory Design: Principles and Practices.* Lawrence Erlbaum Associates, 1993.

[10] C. Stahl. *Spatial Modeling of Activity and User Assistance in Instrumented Environments.* PhD thesis, Universität des Saarlandes, Department of Computer Science, 2009.

[11] G. Vanderheiden and G. Zimmermann. Use of user interface sockets to create naturally evolving intelligent environments. In *Proceedings of the 11th International Conference on Human-Computer Interaction (HCI 2005)*, Las Vegas, Nevada USA., 2005.

[12] G. Zimmermann and G. Vanderheiden. The universal control hub: An open platform for remote user interfaces in the digital home. In J. A. Jacko, editor, *Human-Computer Interaction*, volume 4551 of *LNCS*, pages 1040–1049. Springer, 2007.

[13] G. Zimmermann and G. Vanderheiden. A dream... the universal remote console. *ISO Focus+*, 2010.