Using Meta User Interfaces to Control Multimodal Interaction in Smart Environments

Dirk Roscher, Marco Blumendorf, Sahin Albayrak

DAI-Labor, TU-Berlin
Ernst-Reuter-Platz 7
10587 Berlin

Firstname.Lastname@DAI-Labor.de

ABSTRACT

Smart environments bring together multiple users, (interaction) resources and services. This creates complex and unpredictable interactive computing environments that are hard to understand. Users thus have difficulties to build up their mental model of such interactive systems. To address this issue users need possibilities to evaluate the state of these systems and to adapt them according to their needs. In this work we describe the requirements and functionalities for evaluating and controlling interactive spaces in smart environments from the system and the user perspective. Furthermore we present a model-based implementation of these capabilities which is accessible for the user in form of a meta user interface.

Categories and Subject Descriptors

H.5 [Information Interfaces and Presentation]: User interfaces; H.1.2 [Models and Principles]: User/Machine Systems-Human factors; H.5.2 [Information Interfaces and Presentation]: User Interfaces-graphical user interfaces, interaction styles, input devices and strategies, voice I/O.

General Terms

Management, Design, Human Factors,

Keywords

Meta user interfaces, human-computer interaction, smart environments, model-based user interfaces

1. INTRODUCTION

The ongoing realization of the ubiquitous computing paradigm and the creation of environments holding multiple networked (interaction) resources lead to new forms of human-computer interaction. While current systems support multiple applications through multi-tasking and multiple users one after the other or via web-based applications, their interfaces are usually build for one user using one service with one limited and fixed set of interaction resources. Future interaction in smart environments however brings together multiple users, multiple interaction resources and multiple services (applications). This raises the need to manage and control the assignment of resources, users and services and leads to the complex problem of considering the multiplicity in three dimensions (Figure 1).

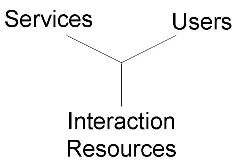

Figure 1: The problem is characterized by multiple users using multiple services via multiple interaction resources, which leads to a highly complex scenario with different dimensions.

Considering multiple services simultaneously (1) e.g. requires the distribution of screen space among them, the shared usage of interaction resources like microphones or loudspeakers as well as the exchange of semantics and information between the services to reach a useful level of interconnection. Multiple simultaneous users (2) require e.g. the shared or alternating usage of interaction resources, the resolution of conflicts, the collaborative usage of resources and services, the possibility to exchange information between multiple users as well as the consideration of privacy issues. Finally the multiple available interaction resources (3) drive new forms of interaction, but this also requires e.g. the possibility to directly select and address resources according to the needs of users and services, the management of resources (occupied resources), the distribution of information across multiple resources or the adaptation to the resource properties. As different resources can also support different modalities this involves the utilization of multimodal interaction capabilities. In this paper we mainly focus on the latter aspect (3), being a facilitator for the former two. Without the possibility to manage the utilization of interaction resources, it is very unlikely that multi-user and multi-application scenarios can benefit from the availability of multiple interaction resources.

In the remainder of the paper, we first introduce the user perspective by explaining the functionalities users need to manage the utilization of interaction resources. Thereby they can determine which services or parts of the services are presented on or controlled through which interaction resources. Following this we elaborate on the system perspective by explaining how the system manages services and interaction resources and provides the functionalities to establish connections between both entities. In section 4 we introduce our implementation. Based on a runtime system using a user interface model with multiple levels of abstraction to describe such multimodal, distributed user interfaces for smart environments, we present how the user can manage the utilization of interaction resources. A comparison to the related work and a summary and outlook complete the paper.

2. THE USER PERSPECTIVE

The utilization of multiple interaction resources (IRs) at the same time poses new demands on users. The user needs the possibility to keep track of the user interfaces from the different services (service UIs) spread across different IRs and should be provided functionalities to alter the configuration according to her needs. We refer to this as the configuration of the (personal) interactive space of the user. An ambient interactive space has been defined as a dynamic assembly of physical entities coupled with computational and communicational entities to support human activities [4]. According to this definition we define the (personal) interactive space as the set of currently used services and interaction resources as well as the connections between them for the remainder of this paper (see also Figure 2). The interactive space thus defines which services or parts of services the user currently accesses and the way she accesses the different services (through which interaction resources).

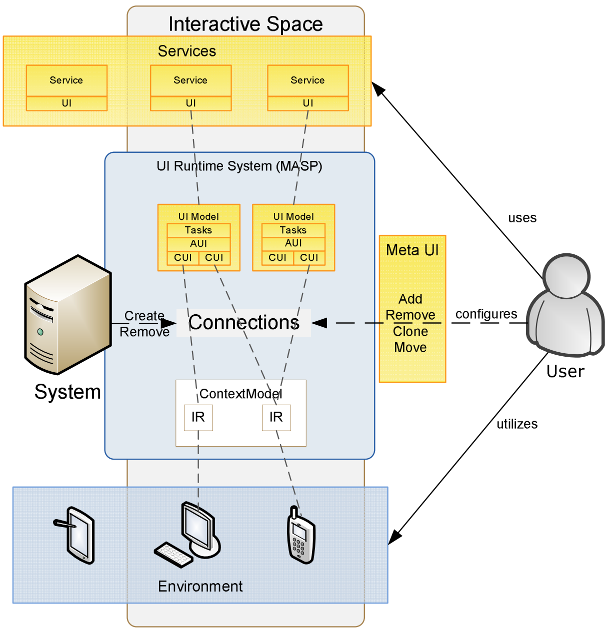

Figure 2: Distributed interaction in smart environments via the personal interactive space. A runtime system manages the user interfaces interfaces (in form of UI models), the interaction resources (in form of context model) and the connections between the two. The user controls the interaction via a meta-UI.

From the user perspective the utilization of one or several services currently available in a smart environment, thus requires the configuration of her interactive space to determine the IRs she wants to utilize the services. Two possibilities can be addressed to configure the personal interactive space: (1) the configuration of a single IR and (2) the configuration of a set of multiple IRs. In the first case, the user uses a given IR to control the utilization of this very interaction resource. This means the IR provides access to a meta-level of the user interface, allowing the alteration of its presentation. For a specific IR this includes adding or removing parts of the service UI to/from the IR. In the second case, the user again uses an IR to access a meta-level of the user interface. In this case however, the configuration via the IR also affects other IRs. The user can move or clone part of the service UI between IRs or add and remove elements to an IR, different from the one currently used. This second configuration requires access to the complete environment and the available services and provides a freely configurable interactive space.

To make these functionalities available for the user, a configuration interface is required that has to be provided independently from the services. Meta user interfaces (meta-UIs) have been proposed to provide such common facilities for user interfaces and thus a generic control on a meta-level [5]. As illustrated in Figure 2 this meta-UI provides the possibility to configure connections between interaction resources and services, allowing defining which service UIs (or parts of service UIs) are utilized through which interaction resources. As IRs provide different capabilities and support different modalities this also requires the (multimodal) support of the different resources by service UIs.

To address this issue we utilize a runtime system providing distributed multimodal user interfaces. As described in the next section, this runtime system is aware of the context-of-use and manages the service UIs. It also controls the connections between service UI parts and interaction resources. The meta-UI is provided as the control interface to configure the runtime system.

3. THE SYSTEM PERSPECTIVE

From the perspective of the system, the configuration of a users' interactive space, involves the management of the service UIs and their current states as well as the available IRs. To support the distribution of (multimodal) user interfaces, the underlying runtime system is also responsible for the assignment of IRs to service user interfaces, so that the user can interact with the service. In our work, we assume a server side system that is aware of the available services (in form of a UI model for each service) and of the available IRs (represented in a context model). Figure 2 shows the elements of this runtime system. The interaction with a service is defined by a UI model that is combined of different levels of abstraction. Similar to the CAMELEON Reference Framework [1] we distinguish task, abstract UI (AUI) and concrete UI (CUI) level, which allow the modality independent definition of the interaction and the provisioning of different concrete modality-specific representations. In our approach the UI model provides a state at runtime, which describes the currently possible interaction at all times [4].

The runtime system also continuously senses the environment for new IRs and manages them in a context model. The model comprises information about users, environment and IRs, where representations of the IRs define the available resources internally for the system. The runtime system uses the IR representations to push CUI elements to these resources as we described in [3]. Thereby the system selects the CUI element matching the constraints of the IR. Before it pushes this element to the IR, it performs the necessary adaptation steps to ensure an optimal presentation. Based on these functionalities the system can provide capabilities to establish connections between the service UI elements defined on the task level and the interaction resources making these elements accessible for users.

In the simplest case, this leads to UI elements connected to a single IR, e.g. the presentation of a user interface on a screen. The ability of the system to maintain and alter this connection and the possibility to push the UI elements to any connected IR now also allows changing the target IR, leading to the migration of the UI e.g. to another screen. Redirecting the elements to an IR of another modality could e.g. also lead to starting a voice dialog. However, to realize multimodal interaction we aim at the simultaneous utilization of multiple interaction resources. This in turn requires the distribution of the available UI elements to multiple IRs simultaneously (see also [12, 7]). Multimodal interaction can be created, if these IRs support different modalities. Redundancy in the interaction can be created by connecting the same UI element to multiple IRs.

The different configuration scenarios described above can technically be brought down to the atomic operations of creating a connection between a CUI element and an IR or removing such connections. For example changing the interaction modality of a task from graphical to vocal includes the removal of connections between IRs and graphical CUI elements of that task and the creation of new connection between the voice CUI elements and the appropriate IR (or IRs). When a "CUI to IR" connection is established our runtime system sends the element to the IR, which then creates the final user interface and delivers it to the user. If a CUI element should no longer be accessible through an IR, the appropriate connection between both is destroyed, which results in the removal of the corresponding FUI from the IR. It must be said, that the association between the CUI elements and the elements at higher levels of abstraction (task and AUI) are always preserved. This is necessary for the state synchronization of all elements as described in [2]. For example, if a task becomes no longer available to the user, the associations assure that all connections between the CUI elements belonging to the task and the interaction resources are removed. As the result the user cannot access the user interface of the task and has no possibility to perform it.

In the next section, we describe our implementation of the described system. The Multi-Access Service Platform (MASP), a modal based runtime system, provides the basis to provide users a meta-UI allowing them to control multimodal interaction.

4. THE MASP & THE META-UI

To evaluate the described approach for the control of multimodal interaction we have implemented a first version of a meta-UI with the MASP, our implementation of the UI runtime system described above. Providing a model-based framework for the development and execution of multimodal multi-device user interfaces, the MASP provides the means to develop interactive services for smart environments. Combined with the capability of the MASP to automatically discover interaction resources in the environment the prerequisites are fulfilled to implement a meta-UI service allowing the user to evaluate and control multimodal interaction.

Figure 3 shows a screenshot of the current implementation of the meta-UI. In the upper, left corner the user can requests the currently available services. In the upcoming list the user can choose which one she wants to connect to the currently used screen. Once the user selects a service the UI of the selected service is shown in the centre and the configuration options at the bottom of the screenshot can be used to configure the current service UI. Here we distinguish four features the user can utilize to configure her interactive space. (1) The migration feature provides possibilities to migrate a service UI from one interaction resource to another to e.g. transfer the UI to another screen better viewable from the users' current position. Through the distribution feature (2) the user can distribute parts of the user interface to other IRs. Thereby the user can also specify if the selected parts should be cloned or moved to the target IR. The third configuration feature is called multimodality (3) and provides possibilities to configure the utilized modalities within the interaction. This allows users to e.g. switch off audio output of the MASP if it is currently disturbing the user. The adaptation feature (4) allows the user to configure further functions of the MASP. E.g. the MASP supports a so called "FollowMe" modus which can be configured through the adaptation feature. The activation of the "FollowMe" modus leads to an automatic configuration of the interactive space by the MASP over time. The MASP senses for changes in the available interaction resources for the user (resources made available or are no more available to the user) and reconfigure the interactive space according to the new resource combination by trying to support a broad range of interaction possibilities.

Figure 3: Screenshot of our implementation of the meta-UI.

These configuration options allow adapting the interactive space according to the possible changes defined e.g. by Coutaz [6]. The user can redistribute the UI elements to different interaction resources (at the interactor level) by moving or cloning elements, migrate parts or the complete user interface to another IR and can also remould the existing user interface on one IR by adding or removing UI elements.

Moreover the status symbols in the upper centre of the screen allow the user to observe which modalities are currently enabled. In the bottom right corner the user can "release" her interactive space, which results in removing all service UIs from all interaction resources.

5. STATE OF THE ART

Several approaches exists which enable the configuration of the relationship between services and IRs by some means or other. Most of them also provide some kind of meta-UI to allow the user to access the configuration possibilities Molina et al. [9] describe a system for the rapid prototyping of user interfaces distributed over several graphical IRs. They shortly mention a meta-UI allowing to distributed user interface elements to other graphical IRs. In [8] a similar approach is presented with a focus on a development framework to design user interfaces distributed over several graphical IRs. The system supports the attachment and detachment of user interface element from/to graphical IRs. An approach which supports the migration of complete user interfaces is presented in [10]. However, none of the solutions we are aware of support the configuration as flexible and broad as described in this paper: the distribution of user interface elements to arbitrary interaction resource(s) to allow free configurable multimodal interaction.

In the mentioned approaches the meta-UIs are not the focus but are developed to give access to exactly the specific described configuration possibilities. They do not consider other functionalities which could improve the possibilities of users to simplify the control of their interactive space. The work described by Vanderhulst [11] is very interesting as it focuses on the meta-UI and not the system side to "put the user in control". However the approach focuses on the handling of services by e.g. start/stop or suspend/resume them. The issue of how to utilize a service is only considered aside. Thus the work should be a good addition to the one described here.

6. CONCLUSION

We presented our approach to control multimodal interaction in smart environments. The functionalities to keep the user in control of the interaction by configuring her interactive space as well as the prerequisites from the system perspective were described. Furthermore a first implementation of the described concept allowing the user to access these capabilities through a meta-UI was presented. However, there are still some aspects which deserve further investigation.

At the moment the user has to configure the interactive space based on the provided information by herself. But the system with its knowledge about the available IRs and services as well as the user and further environment information can at least help the user by providing useful configuration possibilities. Furthermore the system can automatically configure the interactive space of the user (as we started to implement with the "FollowMe"-feature). However, the automatic configuration can also reduce the satisfaction of the user if it does not exactly match her preferences and requirements and should therefore be used careful.

Another aspect that arises with the configurability of the interactive space is the persistence of the user configuration. When the user (re)configures its interactive space, she adapts it to her preferences and needs in the current situation. It thus appears to be suitable that the system utilizes this knowledge by providing the same configuration to the user in the same situation, so the user does not need to do the same configuration over and over again. However, the automatic analyzing of the current situation and the detection of the relevant context parameters is a difficult task which needs further investigation. A first solution could be to let the user specify the relevant situation parts if she wants the system to be able to restore a given interactive space. Another direction for future work are the problems occurring when considering multiple users and multiple services.

7. REFERENCES

[1] Balme, L., Demeure, A., Barralon, N., Coutaz, J., Calvary, G. Cameleon-rt: A software architecture reference model for distributed, migratable, and plastic user interfaces. In *EUSAI*, 2004.

[2] Blumendorf, M., Feuerstack, F., Albayrak, S. Event-based synchronization of model-based multimodal user interfaces. *In Proceedings of the MoDELS'06 Workshop on Model Driven Development of Advanced User Interfaces (MDDAUI)*, 2006.

[3] Blumendorf, M., Feuerstack, F., Albayrak, S. Multimodal user interaction in smart environments: Delivering distributed user interfaces. *European Conference on Ambient Intelligence: Workshop proceedings*, 2007.

[4] Blumendorf, M., Lehmann, G., Feuerstack, F., Albayrak, S. Executable models for human-computer interaction. *15th International Workshop on the Design, Verification and Specification of Interactive System*, 2008.

[5] Coutaz, J., Meta-user interfaces for ambient spaces. In *TAMODIA '06: Proceedings of the 5th international workshop on Task models and diagrams*, 2006.

[6] Coutaz, J., Calvary, G. HCI and Software Engineering: Designing for User Interface Plasticity. In *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications*, 2008

[7] Demeure, A., Calvary, G., Sottet, J.-S., Vanderdonkt, J., A reference model for distributed user interfaces. In *TAMODIA '05: Proceedings of the 4th international workshop on Task models and diagrams*, 2005.

[8] Grolaux, D., Vanderdonckt, J., Van Roy, P. Attach me, detach me, assemble me like you work. *INTERACT'2005: Int. Conf. on Human-Computer Interaction*, 2005.

[9] Vanderdonckt, J., González, P., Fernández-Caballero, A., Lozano, M.D., Molina, J.P., Rapid prototying of distributed user interfaces. In *Proceedings of 6th Int, Conf. on Computer-Aided Design of User Interfaces (CADUI'2006)*, 2006.

[10] Paternò, F., Santoro, C., Scorcia, A., Bandelloni, R., Mori, G., Web user interface migration through different modalities with dynamic device discovery. In *AEWSE'07, 2nd International Workshop on Adaptation and Evolution in Web Systems Engineering*, 2007.

[11] Vanderhulst, G., Luyten, K., Coninx, K., Put the user in control: Ontology-driven meta-level interaction for pervasive environments. *First International Workshop on Ontologies in Interactive Systems (Ontoract 2008)*, 2008.

[12] Vandervelpen, C., Coninx, K., Towards model-based design support for distributed user interfaces. In *NordiCHI '04: Proceedings of the third Nordic conference on Human-computer interaction*, 2004.