Design and Implementation of Meta User Interfaces for Interaction in Smart Environments

Dirk Roscher, Grzegorz Lehmann, Marco Blumendorf, Sahin Albayrak
DAI-Labor, TU-Berlin
Ernst-Reuter-Platz 7, 10587 Berlin, Germany
firstname.lastname@DAI-Labor.de

ABSTRACT
Interaction in smart environments encompasses multiple input and output devices, different modalities, and involves multiple applications. Each of these aspects is subject to changes and thus high adaptation requirements are posed on user interfaces in smart environments. One of the challenges in this context is the assuring of the usability of highly-adaptive user interfaces. In this paper, we describe the design and implementation of a Meta User Interface that enables the user to observe, understand, manage and control ubiquitous user interfaces. Our major contribution is a functional model and system architecture for Meta-User Interfaces for smart environments.

Keywords
Supportive UIs, meta-UI, smart environments

INTRODUCTION
Smart environments comprehend networks of (interaction) devices and sensors that influence the interaction between humans and computers. In contrast to the traditional usage of applications with one PC, the interaction in smart environments comprehends a dynamic set of multiple devices supporting different modalities and involves multiple applications and users. Based on an analysis of multimodal interaction in smart environments, the notion of ubiquitous user interfaces (UUIs) with five distinguished features has been defined in [1]:

1. Shapeability: Identifies the capability of a UI to provide multiple representations suitable for different contexts of use on a single interaction resource.
2. Distribution: Identifies the capability of a UI to present information simultaneously on multiple interaction resources, connected to different interaction devices.
3. Multimodality: Identifies the capability of the UI to support more than one modality.
4. Shareability: Denotes the capability of a UI to be used by more than one user (simultaneously or sequential) while sharing (partial) application data and (partial) interaction state.
5. Mergeability: Denotes the capability of a UI to be combined either partly or completely with another UI to create combined views and input possibilities.

These features enable UUIs to address the variable dimensions of smart environments (multiple devices, modalities, user, applications and situations). By addressing these challenges, UUIs become adaptive and can respond to dynamic alteration of one or more features at runtime. Such adaptations can be done either manually by the user or automatically by the runtime system. An important aspect in this sense is the transparency of system decisions and user control of the features. With respect to these needs, the term *meta user interface* (meta-UI) was established by Coutaz et al. [2] as a definition of "an interactive system whose set of functions is necessary and sufficient to *control* and *evaluate* the state of an interactive ambient space".

Meta-UIs have the potential to help the user in understanding and controlling the high variability within the interactive space. [3] presents a model-driven approach for developing self-explanatory UIs that make design decisions understandable to the user. In [4] a graphical representation of the system's state explains the interconnections between sensors and devices as well as their effects. These works show how the interaction in a highly adaptive interactive space can be improved when giving the user appropriate UI evaluation and control tools. However, there is yet no common understanding of the necessary features of meta-UIs for smart environments.

In the next section, we present an example UUI scenario, in which a meta-UI assists the user. In the section thereafter, based on the features of UUIs and the scenario, we describe necessary functionalities of a meta-UI for UUIs. Afterwards, we discuss the requirements for a runtime architecture for meta-UIs as well as for the actual applications. The section thereon illustrates our current implementation, addressing several of the identified challenges. Finally, we conclude the paper and denote some open research challenges.

INTERACTION IN A SMART ENVIRONMENT
The following scenario illustrates an example UUI and a possible usage of a meta-UI with the help of a calendar application utilized in a smart home environment. Thereby, we want to underline the necessity of control and evaluation capabilities that are required to analyze and configure the ubiquitous calendar application.

Dieter is living in a smart home, equipped with a broad range of networked devices and sensors. Every morning, when Dieter is in the kitchen, he asks his smart home to present him the calendar application with the appointments for today. (1) Dieter can control how the information is presented: if he utters the words "read out", the appointments are presented via voice. Saying "show there" and pointing on the kitchen screen triggers the display of information on the screen. "Silence" disables all voice output. (2) When Dieter leaves the kitchen and walks around his smart home, the voice output follows him until all appointments are read out. Similarly, the displayed information also moves with him to the screens in his vicinity until he confirms to be done with his daily planning. (3) This behavior has been configured and trained by Dieter once after he installed his new calendar application. (4) Training took some effort though, and Dieter could continually monitor the system during the training process, while the system was giving valuable hints about why certain adaptations had been applied.

Sometimes Dieter needs to reschedule appointments to avoid conflicts. (5) To do so, he orders the system to change from voice or screen output to a presentation on the TV, synchronized with the display and controls of his smartphone. This allows him to interact and check details while keeping the overview on the big screen. Rescheduling appointments occasionally raises the need to contact colleagues and customers to agree on a different date or timeslot. (6) For this purpose, Dieter can configure the calendar application to set up video calls to the provided contact data while sharing the relevant calendar information with the called person. (7) Dieter can additionally select information from his notes application to share it. (8) He has the ability to store such a configuration and is able to reactivate the configuration whenever he wants.

EVALUATING AND CONTROLLING UUIs

The above scenario exemplifies UUIs with their five features (shapeability, distribution, multimodality, shareability and mergeability) and shows how the user influences each of these features at runtime. In the following, we describe the functionalities of a meta-UI in general and for all five features of UUIs in more detail.

General Features

According to the definition given in [2], a meta-UI provides evaluation and control features, which in our case allows to manage the adaptation of UIs in our example smart environment. The evaluation functionalities allow users to understand the behavior and current status of the interactive system, while the control features allow the user to influence and change the interactive system according to their needs.

Evaluation functionalities (e.g. (4) in our scenario) address the need of the user to always have access to information about the state of the system and enable the system to inform the user about any changes in the state of the interactive space. Changes do not only include automatic adaptations of the interactive system, but also cover manual adaptations where the user has to be informed as well especially when the manual adaptation does not provide the results expected by the user. Another very important information for the user in case of automatic adaptations is the reason why the adaptation happened. Information can thereby be conveyed implicitly by the look and feel of the UI [5] or be explicitly given to the user, which might be annoying in some cases though.

On the other hand, the control functionalities enable users to configure the interactive system according to their needs. That includes the possibility to configure the features independently on various levels of detail, the triggering of adaptations as well as the control of ongoing adaptations. For automatic adaptation, there is a need to configure the triggers that activate the adaptations, or to (de-)activate such adaptations at all.

The meta-UI has to support the user in the handling of the numerous situations and the possible configurations of the interactive system. Therefore the meta-UI has to provide capabilities to learn from the changes users' made and to store configurations and reapply them when needed ((3) and (8) in scenario).

From our perspective, the meta-UI does not provide functionalities for end-user development as the user cannot create new functionality but "only" adapts and explores the interactive system based on existing functionality.

Shapeability

(5) shows how the user switches between the utilization of different devices and how this triggers the splitting of the UI to two devices. This requires the adaptation of the UI to the actual device features and the provisioning of different representations for the different utilized devices.

In terms of the evaluation of the shapeability feature, any adaptation of the graphical layout (e.g. rearrangements or reorientation of UI elements) should be made transparent for the user. For example, modern tablets and smartphones automatically change their screen orientation depending on how the user is holding them. Usually the orientation changes are animated so the user can follow and understand them. Another common shapeability feedback is a special beep tone indicating the currently configured volume for auditory UIs. Switching between different devices or device combinations, as in the scenario (5), requires even more advanced evaluation features. Users cannot follow the reshaping of the elements across devices and have to be aware of the changes between the different representations. This e.g. includes added or removed information because of more or less screen space.

One example for a more complex adaptation, which requires explicit access to information about the reason of the adaptation and means to control it, is the context-based GUI layouting functionality presented in [6]. The adaptation automatically resizes UI elements depending on the position of the user relative to the currently used display. Animations between different UI layouts are helpful, but not always sufficient to understand the adaptations. Thus, a meta-UI provides information about the position of the user currently detected by the system and the distance to the display. The user also has the possibility to turn the automatic adaptations off at any time.

Distribution

As shown in the scenario (2, 5), in a smart environment the user is able to use various interaction devices, between which the UI is distributed. Furthermore, the devices can also be changed dynamically by redistributing the UI. In terms of evaluation, the user has to be able to keep track of the distribution and may even want to explicitly inquire where a UI element has been distributed to. The user needs to know which devices are used for the output and also which devices can be used to enter data. In case of a redistribution of the UI the awareness of the changes can e.g. be transported by hints like "as you can see on the right display."

The control possibilities for the distribution of a UUI range from the application of distribution configurations preconfigured by the developer, to a very detailed shifting of single UI elements from one device (or even modality) to another performed by the user. Thereby it is also important for the user to know the devices available for a re-distribution and be informed about the potential effects; for example, if all tasks are still supported or if private information is visible to other people on a public display.

A more complex adaptation example for the distribution feature is the so called "follow me" mode illustrated in the scenario (2). Activation of the mode leads to an automatic redistribution of the UI to different devices based on changing situations. The interaction resources (IRs) available for the user are monitored and in case of changes (IRs becoming available or not) the UI elements are redistributed to a new calculated IR combination. Thereby, it is especially important to provide feedback to the user.

Multimodality

In the scenario the use case (1) illustrates how the user utilizes several modalities to interact with the application and seamlessly switches between them.

The user needs to be aware of the currently possible input modalities and ideally also the commands that are provided in each modality (e.g. currently active voice commands, which might be more than actually visible on the screen). A possible solution for implicitly transporting the usable input modalities in the graphical user interface is described in [5]. Control possibilities should at least include the turning on and off for certain modalities. Considering the numerous situations, it should also be possible to define certain situations with certain modality combinations.

Shareability

The capability to share parts of the UI or information with other users is illustrated in (7) within the scenario. This is also a basis for collaboration. While collaborating with other users, the user should be able to view and control which UI parts are shared with whom and with what rights (similar to e.g. social networking sites where it is possible for a user to view how others see the user's profile). Security and privacy thereby play a very important role for shareability. A meta-UI should make the user aware of (and in some cases even warn about) the risks of sharing security- or privacy-relevant UI parts.

Mergeability

Use case (6) shows how the user can merge different applications. This can include the transfer of information from one application to another as well as the combination of functionalities from different applications. The evaluation functionalities comprehend at least information about the current status of merged applications.

To control the merge of different applications, users need to know which applications or part of the applications can be combined with each other. Furthermore, the effects of the merge (e.g. enhanced functionality) also have to be made available for the user.

Based on the scenario analysis carried out in this section, in the next section, we derive requirements for the runtime infrastructure providing a meta-UI.

ARCHITECTURAL REQUIREMENTS

Besides some general requirements, the evaluation and control functionalities described in the previous section pose requirements on the UUIs and the runtime infrastructure in which the UUIs are deployed.

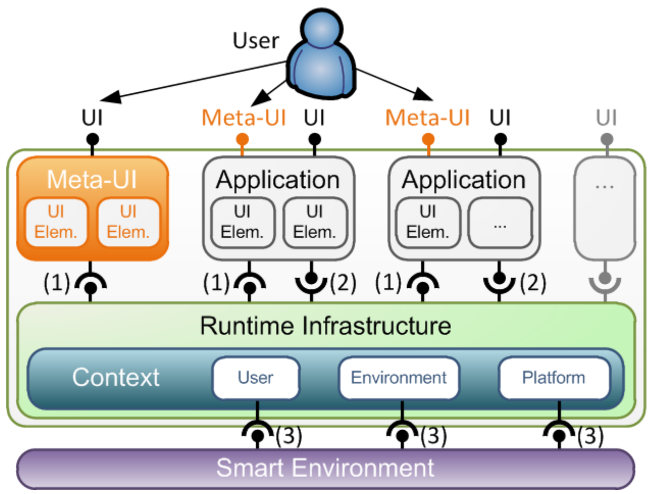

Figure 1: Meta-UI functionalities can be implemented either in a separate meta-UI application (orange box) or be part of applications. Control and evaluation interfaces of the runtime infrastructure (1), the applications (2) and the smart environment (3) are required to implement meta-UIs.

In general, a meta-UI for UUIs must be easily accessible and provide clear functionalities for evaluation and control of the UUIs in the environment. The meta-UI must hide the complexity of the interactive space (in terms of many devices, many modalities, many users, many applications, many and complex situations), while making it perceivable for the user.

As visualized in Figure 1, meta-UI functionalities can be realized twofold – either as a separate meta-UI application, or as part of the applications. In both cases, communication interfaces between the applications, the runtime infrastructure and the environment are needed.

To implement evaluation and control of each UUI feature, a meta-UI must be able to refer to every UI element affected by the respective feature. Thus, each application must provide information about its UI elements, their interaction capabilities and state ((2) in Figure 1). This information must be made accessible for the part of the meta-UI deployed within the runtime infrastructure ((1) in Figure 1). Similarly, meta-UIs require information about the environment, its users and the available platforms. The context information must be gathered at runtime from sensors and devices in the environment ((3) in Figure 1) and made accessible for the meta-UIs ((1) in Figure 1). By interpreting the information about the state of the applications and the context, meta-UIs can explain the current state of the interactive space.

As shown at various stages of the calendar application scenario (1, 5, 6, 7), meta-UI control functionalities require a detailed UUI configuration management. Through a meta-UI the UUI behavior can be configured manually (8) or automatically, e.g. by learning the user's preferences (3). Both pose a challenge for the runtime infrastructure handling different configurations and matching them with the current context situation.

A META-UI FOR SMART ENVIRONMENTS

Figure 1 shows a screenshot of our current implementation of a meta-UI. On the top in the center the user sees the modalities currently utilized for the application. At the bottom four menus enable the configuration of different UI features.

Figure 2: The Meta-UI surrounding the actual UI on the top and on the bottom.

The *Migration* menu provides possibilities to redistribute a UUI from one interaction resource to another, e.g. transfer the graphical UI to a screen better viewable from the users' current position. Through the *Distribution* menu the user can control the distribution on more fine grained levels by distributing selected parts of the UI among the available IRs. The user can also specify if the selected parts should be cloned or moved to the target IR. The selection of relevant UI elements can be done through an overlay display when activating the configuration possibility. The *Modality* configuration menu provides possibilities to configure the utilized modalities within the interaction. This allows users to e.g. switch off audio output if it is currently disturbing the user. Through the *Adaptation* menu the user controls more complex automatic adaptation functions (e.g. (de-)activates the *follow me* mode explained above).

In the future we plan to add the possibility to store and retrieve configurations. We also intend to implement the evaluation and control of mergeability and shareability.

CONLUSION

Meta-UIs are one of the available instruments for handling the variability of smart environments from the user's perspective. We have given an overview of general features Meta-UIs should include as well as of possible evaluation and control functionalities for UUIs. But to realize a well-established Meta-UI for UUIs like the traditional desktop metaphor for single PCs requires to solve many open challenges.

One open issue is to determine the concrete set of needed evaluation and configuration possibilities. Extensive user studies need to be done to solve this. Thereby question like the clustering and grouping of Meta-UI functionality has to be answered including possible different versions of Meta-UIs for e.g. users acting in a known or unknown environment (this e.g. poses additional requirements on the identification of interaction devices).

There are also several challenges for the configuration of the features by the user. One example are automatic adaptations that uses artificial intelligence. In cases of inappropriate behavior, the user should also influence and configure such algorithms. Another issue is the determination of the reason why a user reconfigures the system (context selection). Furthermore, the meta-UI is also a user interface the user is interacting with. So the same requirements for evaluation and configuration holds true for itself.

REFERENCES

1. Blumendorf, M. Multimodal Interaction in Smart Environments A Model-based Runtime System for Ubiquitous User Interfaces. Dissertation, Technische Universität Berlin, 2009.
2. Coutaz, J. Meta-user interfaces for ambient spaces. *Proceedings of TAMODIA'06*, 2006, Springer, 1-15.
3. García Frey, A., Calvary, G. and Dupuy-Chesa, S. Xplain: an editor for building self-explanatory user interfaces by model-driven engineering. *Proceedings of EICS '10*, 2010, ACM, 42-46.
4. Vermeulen, J., Slenders, J., Luyten, K., and Coninx, K. I bet you look good on the wall: Making the invisible computer visible. *Proceedings of AmI '09*, Springer.
5. Weingarten, F., Blumendorf, M. and Albayrak, S. Conveying multimodal interaction possibilities through the use of appearances. 2010.
6. Schwartze, V. Adaptive user interfaces for smart environments. *Proceedings of ICPS'10 Doctoral Colloquium*, 2010.