

# Multimodal Interactive User Interfaces for Mobile Multi-Device Environments

Robbie Schaefer, Wolfgang Mueller  
robbie@c-lab.de, wolfgang@c-lab.de  
Paderborn University/C-LAB  
Fuerstenallee 11  
Paderborn, Germany

## Abstract

*Portable devices come with different individual limitations in user interaction like limited display size, small keyboard and different sorts of input and output channels. To allow ubiquitous, peripheral interaction, a flexible way to provide user interfaces for multiple devices and their different modalities is needed. However, current systems and formats do not sufficiently integrate advanced multimodal interaction and multiple devices. In that context, we present our approach on an advanced system for Multimodal Interaction and Rendering (MIRS) tackling three angles of the problem: UI description, selection of best device and modality, and the UI transformation/adaptation to the selected device. While the UI description is handled by our newly developed XML-based Dialog and Interface Specification Language (DISL), the device- and modality selection employs a profile-based approach, whereas the advanced UI transformations are built upon rule-based transcoding techniques.*

## 1. Background

Our research regarding multimodal user interfaces, and their support on different equipped devices, has its origin in the ITEA VHE-Project (Middleware for Virtual Home Environments) [2]. Goal of this project was to provide a middleware which is able to combine the great variety of existing devices, wired and wireless networks and services for in-home and out-home use, in order to establish a virtual home. One key issue to make these services usable on almost any device, ranging from mobile phones, PDAs, TV-Sets to PCs, was in providing adaptation mechanisms which transform UI descriptions from one format to another. However the underlying existing UI description languages were not generic enough nor did they actively support multimodality and multi-device environments.

Based on the following architecture, we will describe the building blocks, we explore in order to make peripheral awareness computing possible.

## 2. Architectural Overview

Figure 1 shows the general architecture of a server-based peripheral awareness system. The main functionality is in providing the right user interface for an application, selecting the most suited device and modality and then transform the UI for delivery to the selected device.

Applications can be either triggered by the peripheral awareness device, or directly on the device to be used e.g. by an application list showing the applications that are supported by the device. We concentrate the following reflections on the first case, which requires an automated device selection.

When an application is triggered, a generic UI description for this application is requested. In some cases a generic format is not available but maybe other formats like HTML are. Parallel to obtaining the UI, a device and the best fitting input- and output modalities have to be selected. This happens on behalf of a profile manager, which evaluates (and modifies) the current situation of a user and the preferences. Depending on the available profile data, e.g. the location, a selection can be

done. More details of the selection process are provided in section 3. Based on this selection, the transcoder can transform the UI description into a target-format which is supported by the selected device and modality. In case of a device with a browser supporting the generic UI format, the transcoding process may be omitted. The last step is to pass the user interface to the selected device. In order to push the UI to a specific device, any registered device has to contact the server regularly and in case of a positive response a client side UI browser will be started with the retrieved UI.

We will detail our concept of a generic UI description language in section 4 and the work on UI transformation is outlined in section 5.

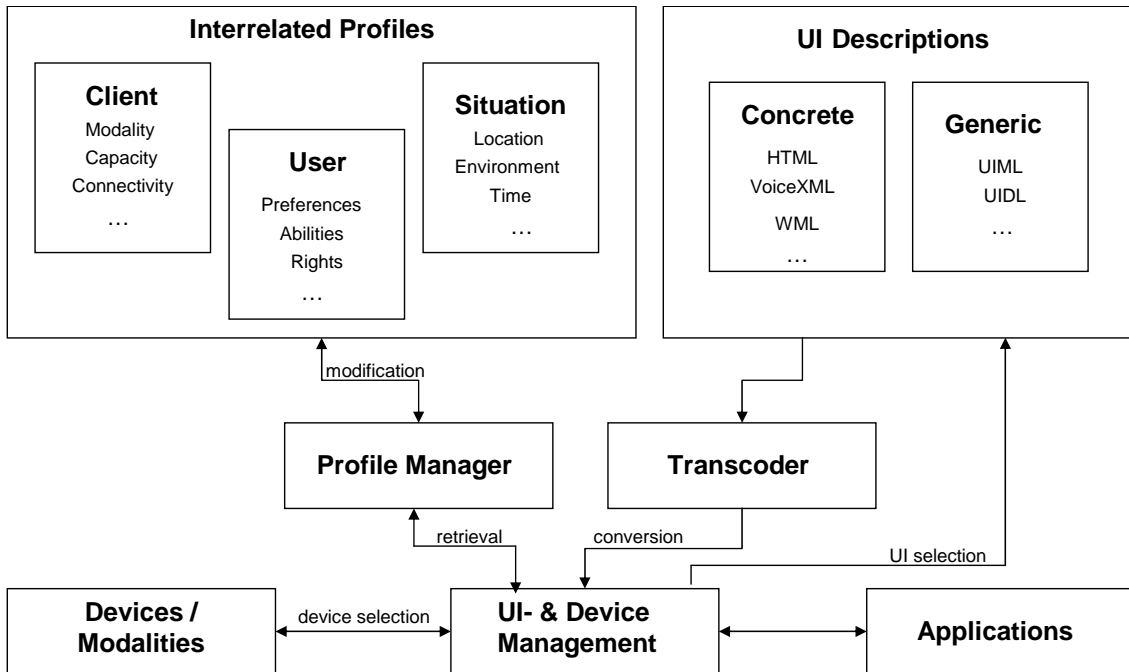


Figure 1. Architecture for a peripheral awareness system

### 3. Device Selection

When a peripheral awareness device needs to gain attention or redirect its information to a device capable of displaying all information, a device selection mechanism is needed. This selection can happen either automatically or by a user initiated process. A user for example could easily redirect the information to an appropriate device in range, by pointing into the direction of the device e.g., a TV-Set. It is expected that future peripheral devices will be equipped with several sensors, which allow computing the pointing direction. However, for our research in the near future, we will be restricted to prototypes as the SoapBox [9]. All sensor data is then stored in a frequently updated profile, which can be read by an application used to select the "best" device and modality. These profiles are evaluated with different values from other profiles forming the overall personal user situation. For example the location of the user together with her orientation is evaluated against the different screen sizes in visible range. We integrate a fuzzy-set-based approach in order to provide a more intelligent evaluation and therefore to achieve optimal results in device selection. [7]

For the profiles describing the user's context, we aim for a profile format supporting automated modification and easy retrieval of properties. This work has been presented on the the workshop in [6].

The device selection mechanism can of course also be applied to the selection of the "best" in- or output channels. For example, a speech interface should be redirected to a screen based UI, when applied in a noisy environment.

## 4. UI Description

Supporting applications for a great variety of devices with different characteristics and limitations, means providing highly scalable user interfaces. One approach is to provide an XML-based description of the user interface and either transform it to other specialized formats, or provide a direct renderer for the target device. While markup languages as HTML scale quite well, they do not provide the expressiveness needed for advanced interaction. UIML [4] proves to be a good candidate for multi-device support, however it is still too tightly connected to the target device and therefore not well suited for providing generic UI-descriptions [5]. Therefore we designed a UIML oriented language which mainly describes the dialog model (with only hints to it's appearance) in order to provide a maximum degree of generality, namely DISL.

The Dialog and Interface Specification Language (DISL) incorporates the Object-Oriented Dialog Specification Notation (ODSN) [8] which has been developed to model complex state space for advanced human-computer interaction. ODSN models the user interaction as different objects, which communicate by exchanging events. Each object is described by the definition of hierarchical states, user events, and transition rules. Each rule has a condition and body, where the condition may range over sets of states and sets of user events. The body is executed when the specified events occur and the object is in the specified state. When the condition becomes true, the execution of the body may change a state. Operating on sets of states can decrease the complexity of UI descriptions dramatically.

DISL is designed for mobile and limited devices, which means that an application can reside on a server and the user interface is obtained over the network. However, traffic must be minimized therefore we provide dedicated DISL Renderers, which locally perform UI state transitions without reconnecting to the server.

## 5. Transformation

After a user interface, which is stored in a specific source format as DISL, has been requested, it might need some transformation. Such UI transformation process is needed for several cases:

- The UI description has to be scaled
- The UI has to be adapted to a different modality
- The target device supports only a specific Format such as WML
- The source format is only available in a specific format

Our approach in transforming UIs is based on DOM-trees [3]. Therefore the UI source format has to be in XML in order to construct a tree which then can freely be modified. The target format can also be different from XML as our transformation system allows arbitrary textual output. The transformer itself is based on our transformation language RDL/TT which allows to describe rules, that are used to modify a DOM-tree. After the restructuring process the tree maps to the target language. The rules are evaluated with different properties obtained by the profiles, and rule could for example examine if the content is larger than the screen size (obtained by the profile) and in that case break the content into several chunks. The profiles are also used to select completely different rulesets before applying them, as for example a speech interface uses a different description format (e.g. VoiceXML) than GUIs. The transformation language and the accomodating profile evaluation to select different rules and rulesets is described in detail in [7].

These transcoding techniques can also be employed to support devices which were not designed for peripheral awareness applications in mind, and therefore be applied in earlier stages of research.

## 6. Future Work

In this position paper we outlined an architecture enabling multi modal user interfaces for multi-device environments. We looked at three building blocks, essential for this aim. One major goal is to provide a new generic user interface description language with multimodal support and to enable selection of devices through a set of profiles which describe the situation.

We plan to improve and enhance our developments in the context of the upcoming Nomadic Media project [1] in order to design an architecture for multi-device support in a greater industrial context. Therefore we will develop and improve these three building blocks and combine them into a functional system.

Having this architecture, research on peripheral awareness is possible, even without having these devices developed as they can be emulated by other devices as PDAs and their UIs can be transformed to any desired format and modality.

## References

- [1] Nomadic Media Consortium. Nomadic media, project page, <http://www.extra.research.philips.com/euprojects/nomadic-media/index.htm>.
- [2] VHE Consortium. Middleware for virtual home environments, project page, <http://www.vhe-middleware.org>.
- [3] Mark Davies et al. *Document Object Model (DOM) Level 2 Core Specification Version 1.0 W3C Proposed Recommendation*. World Wide Web Consortium, September 2000.
- [4] M. Abrams et al. Uiml: an appliance-independent xml user interface language. In *Computer Networks 31, Elsevier Science*, 1999.
- [5] J. Plomp, O. Mayora-Ibarra, and H. Yli-Nikkola. Graphical and speech-driven user interface generation from a single source format. In *Proceedings of the first annual VoiceXML Forum User Group Meeting (AVIOS 2001)*, 2001.
- [6] R. Schaefer and W. Mueller. Adaptive profiles for multi-modal interaction in intelligent environments. In *AI Moves to IA: Workshop on Artificial Intelligence, Information Access and Mobile Computing*, 2003.
- [7] R. Schaefer, W. Mueller, and A. Dangberg. Fuzzy rules for html transcoding. In *Proceedings of Hawaii International Conference on System Sciences (HICSS35)*, 2002.
- [8] G. Szwillus. Object oriented dialogue specification with odsn. In *Proceedings of Software-Ergonomie '93, Teubner, Stuttgart*, 1997.
- [9] E. Tuulari and A. Ylisaukko-oja. Soapbox: A platform for ubiquitous computing research and applications. In *Proc. of Pervasive 2002*, 2002.