

A Generic UIML Vocabulary for Device- and Modality Independent User Interfaces

Rainer Simon
Telecommunications Research
Center Vienna (ftw.)
Donau-City-Str. 1
A-1220 Vienna, Austria
+43 1 5052830 – 47
simon@ftw.at

Michael Jank
Kapsch CarrierCom AG
Am Europlatz 5
A-1120 Vienna, Austria
+43 1 50811 – 3721
michael.jank@kapsch.net

Florian Wegscheider
Telecommunications Research
Center Vienna (ftw.)
Donau-City-Str. 1
A-1220 Vienna, Austria
+43 1 5052830 – 47
wegscheider@ftw.at

ABSTRACT

We present in this poster our work on a User Interface Markup Language (UIML) vocabulary for the specification of device- and modality independent user interfaces. The work presented here is part of an application-oriented project. One of the results of the project is a prototype implementation of a generic platform for device independent multimodal mobile applications. The poster presents the requirements for a generic user interface description format and explains our approach on an integrated description of user interfaces for both graphical and voice modality. A basic overview of the vocabulary structure, its language elements and main features is presented.

Categories & Subject Descriptors: H.5.2

[Information Interfaces and Presentation (e.g., HCI)]: User Interfaces – graphical user interfaces (GUI), interaction styles, voice I/O.

D.3.3 [Programming Languages]: Language Constructs and Features – Data types and structures, frameworks.

General Terms: Design, Experimentation, Human Factors, Languages.

Keywords: Multimodality, multimodal user interfaces, device-independence, UIML, voice interfaces, generic user interface description, mobile devices, mobile networks.

1. INTRODUCTION

There is a growing variety of mobile computing devices and an increased use of mobile data services. Therefore, application developers and user interface designers are faced with a continuous challenge. An increasing effort is required to make sure an application is accessible from all the different types of devices on the market. This is especially true in the context of multimodal access, where traditional graphical user interfaces are complemented by voice interaction [7]. The objective of the MONA (Mobile multimodal Next-generation Applications) project [2] is the development of a generic platform for multimodal services for mobile devices. Target environments for MONA services are WLAN and 2.5G/3G mobile phone networks.

MONA user interfaces are specified in a high-level XML-based format offering an integrated device-independent description for

both graphical and voice modality. The platform translates this generic description into a specific target format. Currently implemented transformations are: WML, HTML, VoiceXML [6], SALT [3] and two proprietary multimodal formats combining HTML/VoiceXML and WML/VoiceXML, respectively.

2. REQUIREMENTS

Device Independence. The user interface description format must provide a generic representation of a user interface. It must be independent of any specific client device technology. Mapping to the target device markup language and adaptation towards device capabilities is performed by the MONA platform.

Modality Independence. Client device input and output restrictions, user preferences and currently active modalities are only known to the platform. MONA applications are unaware of these parameters.

Customizability. The user interface description format must provide mechanisms that offer a high degree of control over layout and graphical appearance. This mechanism must not be restrictive in the context of device independence.

3. GENERIC UI DESCRIPTION

The structure of both graphical and voice user interfaces can be represented in the form of a tree. For graphical user interfaces, the tree determines the geometrical grouping of the elements on the screen. For voice interfaces, the tree reflects the temporal sequence of the user interface dialogue. The MONA description format is an integrated XML tree-view of the user interface that is valid for both modalities. By allowing “voice-only” as well as “graphics-only” branches in the tree, it can also provide different behavior of visual and voice interfaces if desired.

4. UIML OVERVIEW

The User Interface Markup Language UIML [1], [4] is an abstract meta-language for a canonical XML representation of any user interface. UIML itself does not define any device- or toolkit specific tags. It introduces a basic set of tags for defining a user interface structure. In order to use UIML, one must first define a vocabulary.

A number of such vocabularies have been defined. However, many vocabularies are limited in the sense that they are specifically designed for a certain single target technology (e.g. Java AWT/Swing, HTML, WML, VoiceXML [5]). This contradicts the original concept of device independent authoring, since the author requires knowledge of both UIML and the target language. Another limitation shared among the vocabularies we

Copyright is held by the author/owner(s).

WWW 2004, May 17–22, 2004, New York, New York, USA.

ACM 1-58113-912-8/04/0005.

encountered is the fact that they assume a strict one-to-one mapping of abstract user interface elements to target markup language elements. Practical experiences within the MONA project have shown that this is insufficient. A third weak point of current UIML vocabularies (and other generic user interface markup language approaches) is poor control over layout and graphical appearance. This requirement is in conflict with the requirement of device independence and is addressed by our vocabulary's "layout control elements".

5. VOCABULARY STRUCTURE

The MONA UIML vocabulary defines a set of element classes that the author can use to build a user interface. The vocabulary consists of three classes of user interface elements: non-interactive, interactive and layout control. Table 1 lists the available MONA user interface elements.

Table 1. MONA UIML vocabulary

<i>Non-interactive elements</i>	<i>text, image, list, listItem</i>
Interactive elements	command, choice1ofN, choiceMofN, choiceItem, input, table, headerRow, headerCell, tableRow, tableCell
Layout control elements	userInterface, paragraph, columnLayout

Non-interactive elements are used for static presentation of information.

Interactive elements allow the user to communicate with the application business logic. Each interactive element can generate events which can be handled using a defined set of behavior functions. These behavior functions are translated into a suitable target scripting language by the platform.

Layout control elements are container elements that enclose other user interface elements and suggest a graphical layout for them. The platform will consider these suggestions if possible, but may be forced to ignore them due to limitations on the target device.

6. MONA USER INTERFACE EXAMPLE

Figure 1 shows two example graphical user interfaces obtained from the MONA platform's current rendering engine implementation, each displayed on two different mobile device emulators.

7. FUTURE WORK

Enhanced multimodal user interfaces. The MONA rendering engine can currently generate graphical as well as voice user interfaces. Both can be combined to form multimodal interfaces. However, a good multimodal user interface beneficially combines both modalities instead of simply offering both of them in parallel. Future work will focus on improved rendering of true multimodal user interfaces in different input- and output-modality-combinations.



Figure 1. MONA login screen and example MONA voice messaging application on PDA and WAP device emulators.

User interface semantics. More flexible layout control concepts will allow a semantics-based user interface description. With the semantic relationships of user interface elements included in the description markup, the MONA platform will be able to achieve better layout adaptation (e.g. for devices with a landscape screen aspect ratio rather than portrait) and to split a user interface into multiple screens for small-screen devices.

8. ACKNOWLEDGEMENTS

This work was supported within the Austrian competence center program *Kplus* and by the companies Kapsch CarrierCom, mobilkom Austria and Siemens Austria.

9. REFERENCES

- [1] Abrams, M., Phanouriou, C. UIML: An XML Language for Building Device-Independent User Interfaces. <http://www.harmonia.com/resources/papers/xml99Final.pdf>
- [2] ftw.'s MONA project homepage. <http://mona.ftw.at/>
- [3] Speech Application Language Tags (SALT) Forum. <http://www.saltforum.org>
- [4] UIML.org. Draft specification for language version 3.0 of 12. Feb 2002. <http://www.uiml.org/specs/docs/uiml30-revised-02-12-02.pdf>
- [5] UIML.org. Formal Vocabulary Definitions of UIML Vocabularies. <http://www.uiml.org/toolkits/index.htm>
- [6] VoiceXML Forum. <http://www.voicexml.org/>
- [7] W3C Multimodal Interaction Activity. <http://www.w3.org/2002/mmi/>