Multimodal Interfaces in Mobile Devices – The MONA Project

Hermann Anegg, Georg Niklfeld, Michael Pucher, Raimund Schatz, Rainer Simon, Florian Wegscheider
Telecommunications Research Center Vienna (ftw.)
Donaucitystraße 1, 1220 Wien, Austria
{anegg, niklfeld, pucher, schatz, simon, wegscheider}@ftw.at

Thomas Dangl
Siemens Österreich AG
Gudrunstraße 11
A-1100 Vienna, Austria
thomas.t.dangl@siemens.com

Michael Jank
Kapsch CarrierCom AG
Am Europlatz 5
A-1120 Vienna, Austria
michael.jank@kapsch.net

Abstract

This paper describes our work on the MONA project (Mobile multimOdal Next generation Applications, [2]). Within this project we have developed a presentation server for multimodal device independent applications for mobile devices. We discuss two sample applications running on this server: a multiplayer quiz game and a unified messaging application. These applications can be deployed on a wide range of devices including PocketPC PDAs, Symbian based smartphones and even low-end mobile phones with a WAP browser only. The MONA presentation server ensures that the applications' look and feel is consistent on all devices.

1 Introduction

We have developed the MONA presentation server, which enables a new class of mobile applications providing powerful and flexible user interfaces on a wide range of networked devices in telecom networks.

A carrier or third-party service provider who deploys the MONA presentation server can deliver end-user services that combine speech and visual interface techniques for single-user or multi-user applications. MONA applications offer a unique and recognisable user experience which combines several familiar user interface paradigms.

Developers of a MONA application provide a single implementation of their user interface. The MONA presentation server then renders a sophisticated multimodal user interface on devices as diverse as low-end WAP-phones, Symbian-based smartphones, or PocketPC PDAs. Support for other devices could be added to the server easily on customer demand; the applications do not need to be updated.

At the core of the MONA presentation server is the concept of multimodality. A modality is a way to convey information between a user and the interface of an application, using a single distinct human capability to process information. The MONA presentation server supports the modalities of speech, text, graphics, and non-speech audio. Today's mobile devices increasingly support more than just one modality. In fact, almost all current mobile terminals, be they phones or PDAs, come with a microphone, speaker, and graphical display. Yet there exist few applications on mobile terminals that can exploit several modalities in a flexible manner, even though the generally small screen and the frequent change of context suggest that a multimodal user interface should be of great benefit to the user.

There are several reasons for this dearth of multimodal applications. First of all, today's 2G mobile networks and devices encounter problems when transmitting data and voice simultaneously. Second, it is difficult to create user interfaces due to the multitude of devices and their different capabilities, and finally creating a multimodal interface is more difficult than designing for voice or graphics alone.

The first problem is relieved by GPRS and will be fully solved in the 3G systems now emerging. This work deals with the other two problems by relieving the application from having to adapt the user interface to specific devices and modalities.

The rest of this paper will give an overview of the MONA presentation server and the two sample applications we developed on it. After a short section on related work, section 3 gives a detailed view of the architecture of the MONA presentation server. Section 4 showcases our two sample applications, and the last part gives an outlook on our plans for future projects.

2 Related Work

The **W3C multimodality group** [11] works on the standardisation of architectures and languages for multimodal user interfaces. Important to mention is the multimodal interaction framework [6], which formalizes the major components of multimodal systems.

Each component represents a set of related functions and comes with markup languages used to describe data flowing through its interfaces. Our system follows the lines of this framework.

The **W3C's Device Independence Group** [10] is paving the way towards device independent Web access and single authoring.

ETSI is working on a set of generic user interface elements for mobile terminals and services [1]. This set, scheduled for completion in October 2004, will be highly relevant to our work.

Other efforts toward device- and modality- independent user interface description languages include the **eXtensible Interface Markup Language (XIML,** [7]). XIML aims at describing user interfaces for different devices, modalities and contexts.

The **User Interface Markup Language (UIML**, [9]) is a XML-based meta-language for the canonical representation of any user interface. Our user interface description language is based on UIML but extends the original UIML by concepts inspired by the related work mentioned above.

3 The MONA Presentation Server

The MONA presentation server supports the deployment of device-independent applications that combine a graphical user interface (GUI) with speech input and output. Figure 3-1 gives a high level overview of the MONA architecture concept.

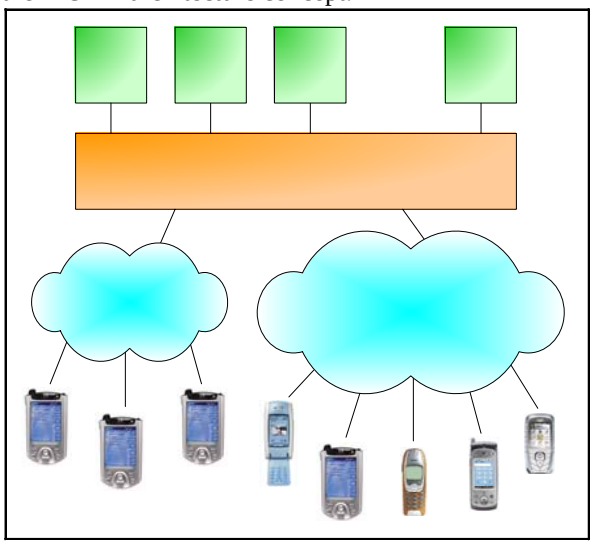

Figure 3-1. MONA Server and Clients.

Client devices access the server over a wireless network. The MONA system supports 2.5G and 3G mobile networks as well as wireless LANs.

In order to allow for a flexible and yet uniform application concept across all different device categories, we chose a browser-based approach. Client devices interact with applications using a browser and a circuit- or packet-switched voice connection. This way MONA applications scale gracefully from low-end WAP phones to high-end Symbian-based smartphones and powerful (X)HTML-enabled handheld computers.

The main innovative features supported by the MONA presentation server are:

- **Adaptive rendering** of a single, abstract user interface description to a concrete target format suitable for a specific client device.
- Support for **asymmetrically multimodal interaction** between users. The MONA presentation server can convert messages exchanged between users in such a way that they are suitable for the current modality settings of each individual user.
- **Broadcasting** of user interfaces to several clients without application involvement.
- **Application-initiated push** of new user interfaces. Mona applications are not restricted to the standard browser request/response interaction metaphor. The client browser also allows pushing of user interfaces. On low-end phones that do not support a client browser, WAP push [12] can be used for pushing pages to the user.

3.1 Requirements for the MONA Server

Modality independence: All issues concerning different input and output modalities must be resolved by the server. A multimodal application is (in general) unaware of a user's current input and output modalities. Users' input and output modalities may change on the fly during a session.

Device independence: When using the term "user interface" in the context of the application, this term refers to an abstract, descriptive and device-independent representation of a user interface. The application is (in general) unaware of the specific capabilities of the client device.

Application independence: MONA aims to provide a generic solution for the deployment of any conceivable multimodal application. The server must not be designed in way that it restricts or limits development of future applications and use of future input or output technologies. It must also not be designed in a way that prevents the application developer from having full control over user interface content (for all users separately, if required) at all times, within the device's restrictions.

Application push: Communication between the application and the user is expected to be initiated by either side. A mere "user requests"-"application responds" model is not sufficient.

Fine grained control: While the presentation server makes sure the application is useable with the

default translation of the generic user interface to a specific device and modality, the application may take high detail control over the user experience. There should be several levels at which the application can determine rendering and presentation settings.

3.2 Implementation / Server Architecture

Figure 3-2 shows the architecture of the server we implemented at ftw. MONA applications are implemented as web services that interact with the server via a SOAP interface. The payload of the SOAP messages are the user interface descriptions in the MONA-specific UIML format.

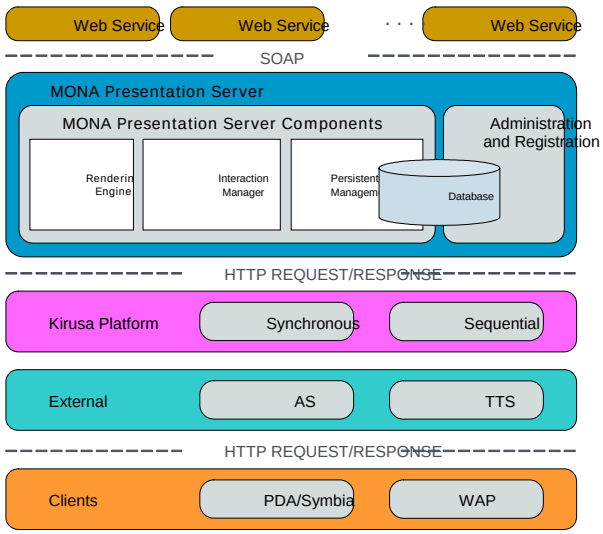

Figure 3-2. MONA System Architecture

The main module of the MONA presentation server contains the interaction manager, the core server component taking care of user session management and broadcasting. The rendering engine performs the transformation process from abstract user interface to concrete target markup language. Currently implemented target languages are HTML, WML, VoiceXML and two proprietary multimodal formats combining VoiceXML and HTML or WML, respectively.

The persistent data management handles data stored in the MONA database. Persistent data includes data related to registered users, currently active sessions and registered applications.

Database administration via web interfaces is enabled by the administration and registration module.

Input integration and synchronization issues for multimodal client devices are resolved by the Kirusa [3] platform(s), which communicate with the MONA server via HTTP.

High-End devices like PDAs or Symbian smartphones can access the MONA server via the Kirusa synchronous multimodality platform, which offers a full multimodal user experience with synchronized voice and visual user interfaces.

Low-end WAP phones can connect to the MONA server via the Kirusa sequential platform, which only offers sequential multimodality, i.e. the user has to switch between voice-UI and GUI.

External components connected to the Kirusa platform include a speech recognition engine by Nuance [4] and a Text-To-Speech module by SVOX [5].

4 MONA Sample Applications

We are currently developing two sample applications for MONA: a unified messaging client and a multimodal quiz game. In order to cover a high diversity of aspects, we conceived the messaging client (MONA@Work) as single-user, dialog based application and the quiz (MONA@Play) as multi user real-time game (Figure 4-1).

Our applications serve following purposes:

- Demonstration of usage scenarios for mobile multimodal user interfaces in different domains such as work and entertainment.
- Guiding the development of the UIML vocabulary used for abstract interface descriptions.
- Providing a reference software architecture and framework for future MONA applications.

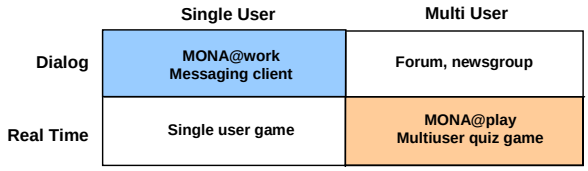

Figure 4-1: Application classification matrix.

UIML development: Application developers should not be concerned with handling different device characteristics and interface modalities and users should be free to use all interaction modes offered by their client terminals. Therefore, MONA applications deliver device- and modality-independent interface descriptions on an abstract level, to be subsequently rendered by the server for specific clients.

The UIML standard itself only defines an overall markup syntax and structure. In order to practically apply UIML, one must define a vocabulary. For the MONA project we defined a custom generic vocabulary for the device- and modality independent description of user interfaces. It consists of three classes of UI elements:

- Non-interactive: used for static information.
- Interactive: allow the user to communicate with the application business logic.

- Structure control: enclose other UI elements, capture the overall semantic and task structure of the interface, and suggest graphical layout rules.

Combining UIML with a browser based approach: Most high-level user interface description languages aim at describing all screens of an entire application. Since the MONA presentation server is based on the Kirusa platform and a browser based client environment, our user interface- and rendering-system must be designed for single user interface screens, not entire "sets" of user interface screens. This page-based approach is different to the concept of describing all possible interactions of an application in a single specification. It results in a considerable limitation: real-time synchronization between client-side user interface and the server-side application cannot be achieved due to network round-trip latencies.

Software Architecture: MONA Applications only contain back-end business logic reacting to events and responding with abstract UIML interface descriptions. For this reason we implement them as J2EE-webservices in a JBoss [13] container. Communication with the MONA server uses SOAP [8] messages containing user-actions or UIML pages. This loosely coupled architecture allows third party providers to reuse existing business logic and easily connect their applications to a remote service provider's platform.

Application Design Process: Figure 4-2 depicts our application design process. We targeted three different devices representing relevant mobile client categories: PDAs (IPAQ 5450, PocketPC 2003), Smartphones (Nokia 7650, Symbian Series 60), and WAP phones (Nokia 5100). Based on use cases with multimodal extensions we designed multimodal interface prototypes (HTML + VoiceXML) for each device. These prototypes drove the development of abstract interface elements (e.g. choice 1ofN, button) used in the applications' UIML interface descriptions. The designs also served as a yardstick for server development since they represent the desired output of the rendering engine.

Usability activities: In order to ensure high usability and acceptance of our demo applications, we involve users throughout the project. We started with three scenario based design workshops in which participants refined and validated our exploratory application concepts and sketches. The workshops provided valuable insights regarding prioritization and usage patterns of application features in mobile contexts.

We subjected our visual design prototypes to a heuristic evaluation: evaluators with design experience used a set of heuristics for mobile multimodal interfaces to provide feedback and suggestions for improvement.

We tested our first multimodal prototypes for PDA and smartphone with 20 undergraduate business students in a user-driven design workshop focusing on voice usability, in particular on timing and prioritization of spoken content.

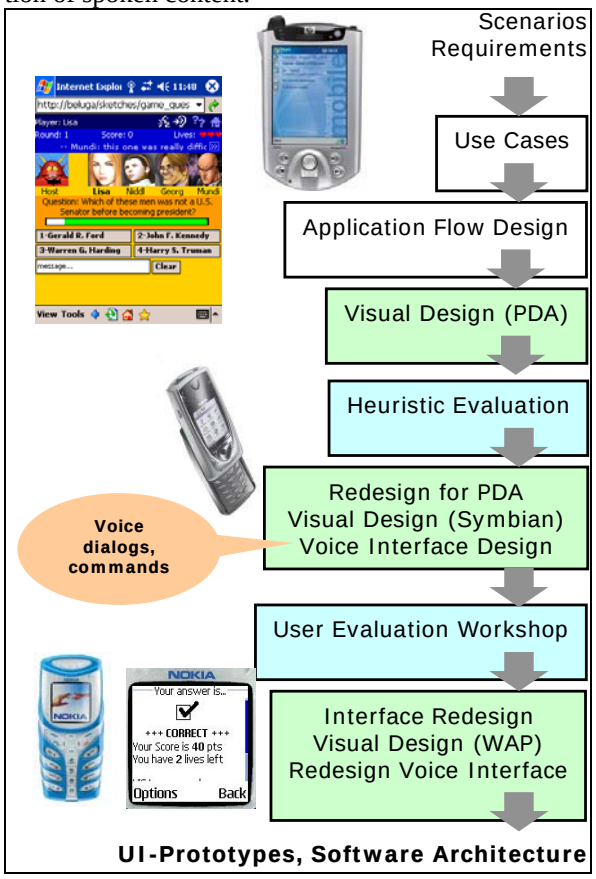

Figure 4-2: Application Design Process.

4.1 MONA@Play – The MONA Quiz

Our sample application for the entertainment domain is a multi-user quiz in the style of "Who wants to be a millionaire?" [14]. 2-4 human players interact in real-time with a virtual game host posing quiz questions (Figure 4-3). Players have limited time for answering. If a player gives a wrong answer, he loses one of his lives. The last player alive wins the game.

Rich interaction between players (on the social level as well as on the gameplay level) is highly relevant for the entertainment value of multi-user games. Therefore the quiz provides a multimodal chat for user-to-user communication. Players can send and receive messages using any modality (visual or voice), which demonstrates the MONA server's cross-modality features: spoken chat is automatically translated to written messages for GUI-only receivers and vice versa.

Personalization and social presence are important aspects of multi-user games: therefore each user is

represented by an avatar expressing different moods, depending on the current game situation.

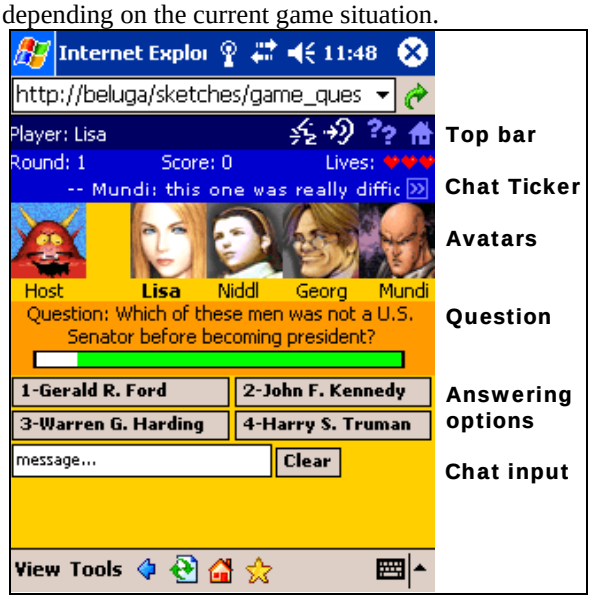

Figure 4-3: Question & Answer (PDA) [15].

During a quiz session, various time-constraints are imposed on players by the game's rules and the UIs of multiple users have to be synchronized (e.g. when the next question is asked). Along with the real-time demands of the chat, the fulfilment of these requirements relies heavily on pushing content to user clients. In this regard MONA@Play abandons the request-response pattern typical for web applications in favour of a more asynchronous, push-based interaction model.

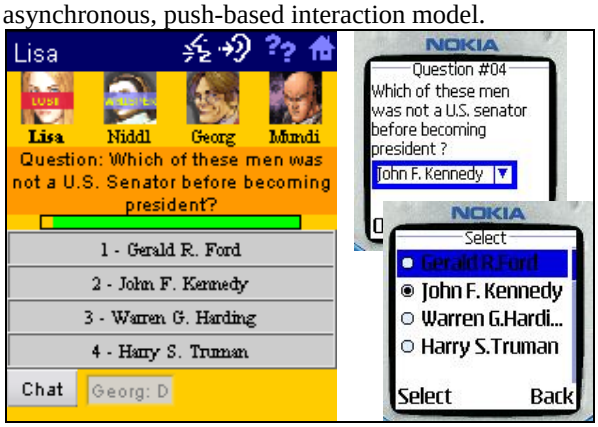

Figure 4-4: Q&A for Smartphone / WAP [15].

Maintaining a consistent game interface across the three targeted device classes revealed following design challenges: screen sizes differ considerably (240x320 on PDA vs. 96x65 for WAP-phone) making prioritization of widgets and features necessary to enable graceful UI degradation (Figure 4-4). Different interaction techniques (point & click on PDA vs. joystick & keys on smartphone) as well as different types of modality (simultaneous on PDA vs. sequential on WAP-phone) demand for an ordering of prompts and widgets. Therefore we extended our MONA UIML vocabulary with attributes enabling developers to prioritize and determine the sequence of user interface elements.

4.2 MONA@Work – A Messaging Client

The second MONA sample application is a mobile multimodal unified messaging client. It enables the user to administrate emails, SMS, MMS and voice messages through an interface that is especially designed to assist operation for a mobile user. Frequently needed functions may be performed visually, with help of voice commands or by a voice-only interface. The underlying scenario is a business user who quickly wants to get an overview over new messages while walking in the city, listen to the terminal reading the content of a new emails, and reply instantly with a voice message.

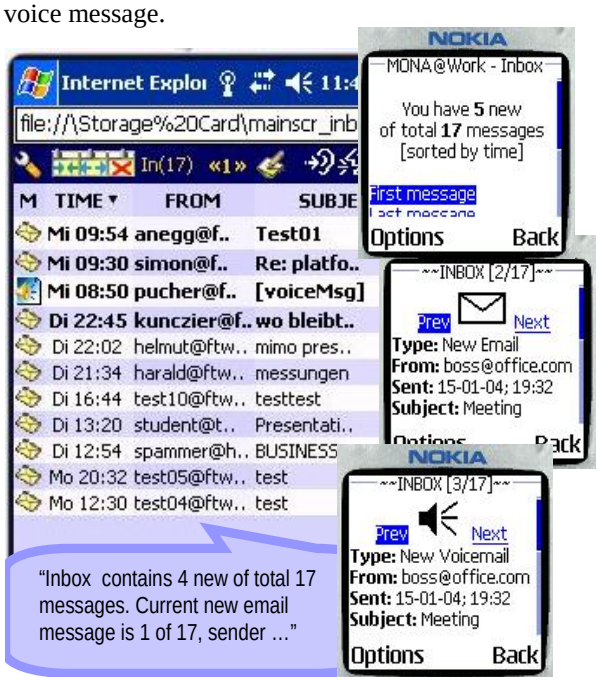

Figure 4-5: Messaging for PDA / WAP [16].

In contrast to the quiz application, the messaging client uses other services in addition to the MONA presentation server: an IMAP / SMTP server for email and voice messages and a ParlayX gateway for SMS and MMS communication. Furthermore the messaging client is a single user application, the push mechanism of the server allows the application to inform the user when a new message has arrived. The user may read or listen to the content of text messages by TTS. Preferences and user information are stored in the application and not in the terminal. Thus the user may switch devices, e.g. borrow the mobile from a colleague, log on to the MONA server and operate MONA@Work without limitations. This is an advantage over a built-in

email client that stores access data in the terminal. An address book will assist the user and provide convenience if the user selects a contact by voice. Messages are grouped like in known messaging clients.

Figure 4-5 shows the box for incoming messages together with some of the corresponding WAP screens and an example for voice output. The top bar is used for application specific navigation, e.g. links to different mailboxes as well as switching modality options. Most users are familiar with a message box presented in tabular form. While on a PDA due to its screen size this is feasible, for a WAP phone it is not. Also the voice interface provides navigation throughout the content of the message box. For WAP the table on the PDA is divided and every row of the table is rendered to a separate WAP page. The first WAP page of the Inbox provides an overview over the box. Its voice presentation features a dialogue system that guides the user from table row to table row. Since the application provides the content of the table to the presentation server in UIML, this is one example how the design process helped to develop the UIML vocabulary. However, the UIML table object is designed as a generic object without any messaging-specific features and may be reused by future applications.

5 Further Research

The following aspects are shall be studied later.

Conversational Speech Recognition is supported by the MONA platform. We will test this type of speech recognition, based on statistical language models, with several models. Robust recognition and summarization methods will increase the accuracy and usability of Conversational Speech Recognition.

Interface migration: Moving the user interface to a different device with different capabilities poses no basic challenge to our server but simply requires the user to login on the new device. The presentation server needs to load the new capabilities and forward subsequent UI descriptions to the new device.

Multi device interaction: We currently expect one user to have only one active device at a time, responsible for all modalities. As many people own and use several mobile devices – generally a phone and a PDA, both with screen and audio – it seems sensible to make use of them and investigate in this direction.

Bandwidth limitations: It will be interesting to see whether our multimodal interfaces remain usable in limited bandwidth (e.g. GSM) environments. We will therefore carry out some of our usability tests in areas with heavy GSM traffic, e.g. downtown Vienna.

Other modalities: While we currently do not support other modalities than voice and graphics, we see that especially handwriting recognition will be an important input modality in the foreseeable future.

Partial UI Updates: In order to minimize user interruption, pushing complete pages should be avoided as it tends to block the client interface for some seconds. In future versions of the MONA presentation server an application may update single elements of the user interface using DOM mechanisms. It remains to be seen whether this simplifies application development and reduces bandwidth requirements.

6 ACKNOWLEDGEMENTS

We wish to thank the rest of the MONA team for their help and insights contributing to our work. Further acknowledgments go to Kirusa, Inc. for allowing us to build upon their multimodal platform, and to Nuance and SVOX for contributing their technologies.

The MONA project is funded by Kapsch CarrierCom AG, Mobilkom Austria AG and Siemens Österreich AG together with the Austrian competence centre programme Kplus.

7 References

[1] ETSI generic user interface elements for mobile terminals and services: http://portal.etsi.org/stfs/hf/STF231.asp

[2] Ftw's project MONA webpage. http://mona.ftw.at

[3] Kirusa, Inc. website: http://www.kirusa.com/

[4] Nuance, Inc. website: http://www.nuance.com/

[5] SVOX Ltd. website: http://www.svox.com/

[6] Larson, J.A., et al. W3C Multimodal Interaction Framework, W3C Note, 6 May 2003. http://www.w3.org/TR/2003/NOTE-mmi-framework-20030506/

[7] Puerta, A. and Eisenstein, J. XIML: A Common Representation for Interaction Data. Proceedings of IUI 2002 (San Francisco, California, USA), International Conference on Intelligent User Interfaces, ACM Press.

[8] Simple Object Access Protocol (SOAP) 1.1, W3C Note, 8 May 2000. http://www.w3.org/TR/2000/NOTE-SOAP-20000508

[9] Abrams, M., et al. UIML: An Appliance-Independent XML User Interface Language. In Proceedings of The Eighth International WWW Conference (Toronto, Canada, 11-16 May 1999), Elsevier Science Publishers.

[10] W3C Device Independence Group (DI) website. http://www.w3.org/2001/di/

[11] W3C Multimodal Interaction Group (MMI) website. http://www.w3c.org/2002/mmi/

[12] WAP Push Architectural Overview. Wireless Application Protocol Forum 2001. http://www1.wapforum.org/tech/documents/WAP-250-PushArchOverview-20010703-a.pdf

[13] JBoss group website. http://www.jboss.org/

[14] TV-Show: "Who wants to be a millionaire?" http://www.millionairetv.com.

[15] Schatz, R., et al. Mona@Play Design Document (Internal Report). 2003. Available from: http://mona.ftw.at/

[16] Anegg, H., et al. Mona@Work Design Document (Internal Report). 2003. Available from: http://mona.ftw.at/