

A Multimodal Interaction Manager for Device Independent Mobile Applications

Florian Wegscheider

Telecommunications Research Center Vienna
Donau-City-Str. 1
A-1220 Vienna, Austria
+43-1-5052830-62
wegscheider@ftw.at

Thomas Dangl

Siemens Österreich AG
Gudrunstr. 11
A-1100 Vienna, Austria
+43-51707-46554
thomas.t.dangl@siemens.com

Michael Jank

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

Rainer Simon

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

ABSTRACT

This poster presents an overview of the work on an interaction manager of a platform for multimodal applications in 2.5G and 3G mobile phone networks and WLAN environments. The poster describes the requirements for the interaction manager (IM), its tasks and the resulting structure. We examine the W3C's definition of an interaction manager and compare it to our implementation, which accomplishes some additional tasks.

Categories and Subject Descriptors

H.5.2 [Information Interfaces and Presentation (e.g., HCI)]: User Interfaces – Evaluation/methodology, User interface management systems (UIMS)

General Terms

Design, Experimentation, Human Factors, Languages.

Keywords

Multimodal interface, session management, interaction manager, device independence, multi-user applications, mobile network.

1. INTRODUCTION

Today's mobile devices increasingly support more than just one modality. Yet there are few applications that support several modalities on mobile terminals, even though the generally small screen and the frequent change of context means that a multimodal user interface would be of great benefit to the user.

There are several reasons for this: First of all, today's mobile networks encounter problems when transmitting data and voice simultaneously. Second, creating a good UI is difficult due to the multitude of devices and capabilities, and finally creating a multimodal interface is harder than sticking to just one modality.

The first problem was relieved by GPRS and will be fully solved in UMTS systems. This (and others') work deals with the other two problems by relieving the application from having to adapt the user interface to specific devices and modalities.

2. Overview of the MONA Platform

The MONA (Mobile multimodal Next generation Applications) platform [3] is a server-based platform which makes it possible to develop applications that combine a GUI with speech input and output. The MONA system supports a range of devices: from low-end WAP mobile phones to high-end Symbian-based smart-

phones and powerful (X)HTML enabled handheld computers. Devices are connected to the platform via 2.5 and 3rd generation mobile networks as well as wireless LANs.

Within the MONA project we develop two applications to show the range of capabilities of the MONA platform: a multi-user game and a single-user messaging client.

The MONA Quiz allows several users to play a quiz game against each other. In addition to the quiz they have the opportunity to chat with each other during the game. This allows for good interaction between the users demonstrating modality independent communication between users.

The Messaging Application is a unified messaging client. It demonstrates modality-independent messaging treating e-mails, SMS, MMS and voice messages alike.

3. REQUIREMENTS

Multimodality: All issues concerning different modalities must be resolved by the interaction manager. The applications are unaware of a user's current input and output modalities.

Device independence: The application is (in general) unaware of the specific capabilities of the client device(s).

Application independence: The interaction manager must not be designed in way that it restricts or limits development of future applications and use of future input or output technologies.

Multi-user capability: MONA applications are generally multi-user applications, i.e. one or more users may concurrently be connected to and make use of an application. The IM manages users and user groups sharing an application.

Fine grained control: While the IM makes sure the application is useable with the default translation of the generic user interface to a specific device and modality, the application can take high detail control over the user experience.

Push pages: As our interaction model is request, page and browser based, we need some special means to push pages to the user. We require both a request and a push interface between the interaction manager and applications.

4. TASKS

The interaction manager, the central component of the MONA platform between applications and the rendering system, covers:

User login and authentication.

Management of user preferences. This means the interaction manager finds the user's modality preferences and device characteristics so the output generation can create interfaces adapted to

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.

the user's preferred modalities and device. It also includes methods for setting preferences.

User **session management** (*not* including application state). This includes the management of several users sharing an application and **broadcasts** to all users in the shared session. Broadcasting requires adapting the UIs as not all users get the same interfaces.

Splitting the user interface if it should be rendered on several devices of the same user (collaborative browsing).

5. INTERFACES

The interaction manager as we see it has three interfaces:

Input from User. Mobile devices access our IM via a 3rd party platform [4] which sends http requests for web pages. Any user input reaches us in the form of a page request with parameters.

Input from Application. The application sends generic user interface descriptions to the interaction manager. We chose the User Interface Markup Language (UIML [1], [2]), an abstract for an XML representation of any user interface, and defined a vocabulary for our task.

Output to Rendering System. The output generation component receives basically the same user interface as the interaction manager, minus the broadcast information, plus user preferences and device information. We use UIML here as well.

6. IMPLEMENTATION

As emphasised in [5] the W3C's interaction manager is a logical component. So is ours, as this chapter will show.

We decided to use an XML publishing framework for transforming the application's UIML input to the required target language (XHTML+Voice Profile, X+V [6]).

A page request triggers two pipelines. The first delegates logic to the interaction manager and application calls receiving UIML, the second queries the database for user preferences and client capabilities. Both results are aggregated to a single XML file which is rendered via an appropriate style sheet to X+V output for the underlying platform [4] (Figure 1).

Our architecture relies on the HTTP request/response model and does not support pushing pages. Our solution to this is a small plug-in for the client browser through which the IM can tell the browser to load a new page. This workaround avoids frequent page reloads not feasible in our low-bandwidth environment. We need this plug-in for PDA and Symbian clients only, on WAP phones we use the WAP push mechanism.

7. CONCLUSION

We have shown requirements, tasks and the implementation of an interaction manager in a multimodal platform.

While we generally stick to the task distribution the W3C suggests for the components of their framework, we did make a few changes: We integrated broadcast functionality into our system and we added some aspects of the session management and the output generation to the interaction manager. Most importantly, we seek to keep the IM independent from the applications and moved the application state back into the application.

Our IM concentrates on modality management and device independence which it efficiently achieves for our applications.

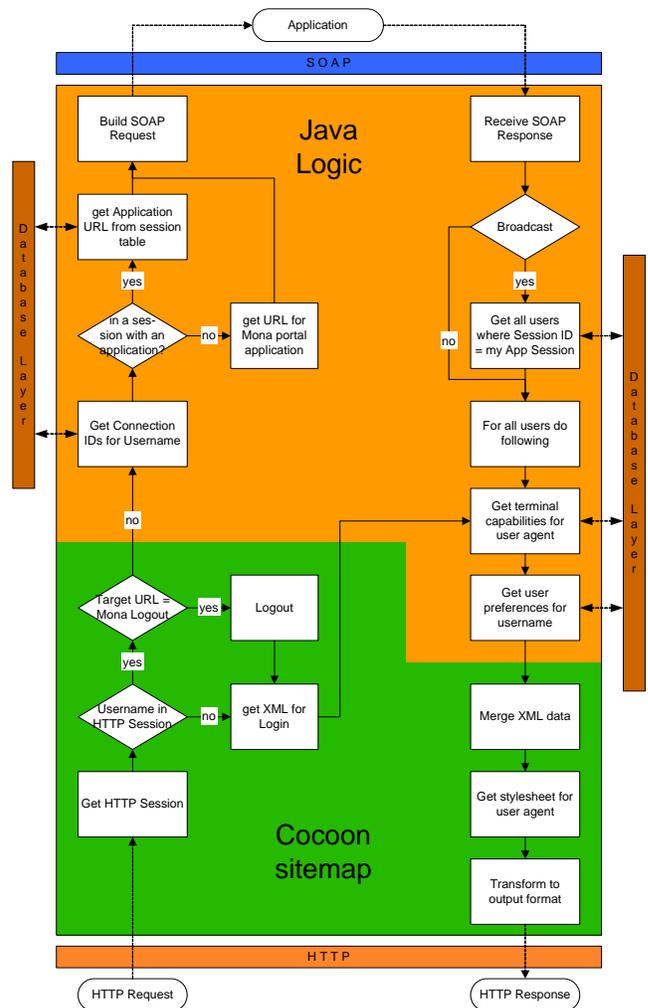


Figure 1. MONA's Rendering Pipeline Including the Interaction Manager's Functions

8. ACKNOWLEDGEMENTS

The MONA project is funded by Kapsch CarrierCom AG, Mobilkom Austria AG and Siemens Österreich AG together with the Austrian competence centre programme Kplus. Special thanks go to Kirusa, Inc. for their multimodal platform.

9. REFERENCES

- [1] Abrams, M. et al. UIML: An Appliance-Independent XML User Interface Language. In Proceedings of The Eighth International WWW Conference, Elsevier 1999
- [2] Draft specification for UIML 3.0. 2002-02-12. <http://www.uiml.org/specs/docs/uiml30-revised-02-12-02.pdf>
- [3] Ftw's project MONA web page. <http://www.ftw.at>
- [4] Kirusa, Inc. website: <http://www.kirusa.com/>
- [5] Larson, J.A., Raman, T.V., Raggett, D. W3C Multimodal Interaction Framework, W3C Note, 6 May 2003. <http://www.w3.org/TR/2003/NOTE-mmi-framework-20030506/>
- [6] XHTML+Voice Profile 1.0. W3C Note, 21 December 2001. <http://www.w3.org/TR/2001/NOTE-xhtml+voice-20011221/>