

RESEARCH REPORT

Device Independent User Interface Description for Wearable Computers

Marcus Nilsson, Peter Parnes

Research report

Institutionen för Systemteknik
Avdelningen för Programvaruteknik

Device Independent User Interface Description for Wearable Computers

Marcus Nilsson, Peter Parnes

August 19, 2002

Abstract

This report discusses the aspect of user interfaces on wearable computers. Today wearable computers can be of very different size and capabilities. The way of communicating with the user also varies for different devices. This paper looks into several different approaches of describing a user interface that is independent of the underlying device and architecture.

Chapter 1

Introduction

Wearable computer are no longer something for the future and we can today see a range of different version being made. From the hobbyist making their own on spare time to companies like Xybernaut, whose business idea is to build wearable computers. As we move forward the need for applications that are using the unique abilities of the wearable computer is emerging. A problem that arises is then how can we target an application for wearable computers when the capabilities of all these devices are very different. One can have a head mounted display with see through capabilities and on another there is only audio as interface. The same problem exists when we look at the different inputs that are available in wearable computers today. One can have a chording keyboard and mouse, while another only have voice as input. If an application is targeted at wearable computers and developed with the most common techniques available today we would have to design an interface for almost every wearable that would use the program. It would therefore be an advantage in finding a way to describe a user interface that is not bond by the capabilities of a certain wearable computer but instead is describing the user interface in a way that can be adopted to a range of different interface techniques.

Chapter 2

Related work

Several languages and techniques have been developed to ease the creation of user interfaces for different Operating Systems and devices. In this section we will look at some of the work that has been done, and what their strong and weak part are from the perspective of this paper.

2.1 UIA

IBM has developed a concept they call UIA [3], universal information appliance. By abstracting the user interface from the rendering on a certain device, it can be adopted to the characteristics of the device that is running it. In the UIA project user interface is described by a XML [2] based language called Mobile Document Application Language (MoDAL). This Language consists of widgets for common objects in a user interface. It also has widgets for sending network messages. These messages are used to send events from the UIA to the service provider on the network. It is up to the device to render a MoDAL interface in a way that fits its capabilities. The UIA project does not stop at user interface but also describe the communication technology layer and a messaging architecture for communication between devices. The UIA project has somewhat a different goal then the problems that we are focusing on in this paper. In UIA the focus is on services and always connected. The UIA services never run directly on the device, instead the UIA application on the device is a thin client that sends and receives messages to and from the service provider. The MoDAL language is interesting in our study as it defines the user interface independent of the underlying OS. The problem is that MoDAL still is designed in terms that are counting on that the technology that will display the user interface is a screen with graphical capabilities. We also have the problem that objects position are not relative but absolute and therefore not as dynamic as they should be when we are talking of devices with these differences in size and resolution.

2.2 XUL

XML User Interface Language (XUL) [1] is an XML based language for describing user interfaces. It is developed in the Mozilla project for describing the user interface in a way that is independent of the OS. It also made it possible to change user interface on the Mozilla program depending on taste, so called themes. XUL has become a very robust and feature rich user interface language and is starting to show up in other applications then Mozilla. Almost everything you would want to do in a graphical user interface can be done in XUL and it also has the ability to have a dynamical placement of objects, which is important when we are moving between devices of different sizes. But the thing XUL lacks in the perspective of this paper is a way to describe a user interface not defined by different graphical objects. XUL is therefore not solving the problem as we want to develop applications that are not only targeted for graphical devices, but also devices with only audio and also interface methods that are not yet developed.

2.3 Java

Java [4] is able to run on a range of different devices and is therefore a good choice for the underlying functionality. But when we come to the user interface there are parts in Java that are positive for this study but also things that are no. One big problem is that the user interface classes for different version of Java are not the same. In J2ME we have one way of describing a user interface and on J2SE and J2EE we have another way. This is of course good if the focus is to use the underlying technology in the best way, but will make it impossible for us to use Java's user interface classes directly as we want to target as many devices as possible. One good thing with Java is that it can describe the user interface in a dynamic way, so that it will to some degree adapt to the display. An interesting thing that Java has is the different Layout Managers. Where a separate class, that is replaceable, defines how different objects will be arranged on the display. In Java we can use both graphical- and audio-based interfaces, but there exist no way of describing a user interface that will adapt to the underlying architecture.

2.4 VoiceXML

The goal with VoiceXML [6] is to develop a standard way to describe interfaces that incorporates synthesized speech and prerecorded audio for output and for input uses speech recognition, DTMF key and recording of audio. VoiceXML is as the name suggests based on XML and describes the user interface as a dialog between the computer and the user. There are for example widgets to define a questions, where to store the answer and what to do when an answer is stored. VoiceXML is a simple and easy language to describe a user interface dependent on only audio, but no support exists for describing anything else. VoiceXML is describing an audio user interface but has the same shortcoming for our purpose as the graphical user interface languages. It describes the user interface in terms that is only applicable on one kind of interface techniques in this case audio.

2.5 UIML

UIML [5] is the language that shares most of the same ideas that are presented in this paper. The vision of UIML is to be able to create an interface that is independent of the device that renders it. The vision seem good, but when looking into the specification it can be seen that the result have not really made the goal. UIML is a XML based language that has the same syntax for all devices. There are widgets in this language that are general and can be used in most interface techniques. But in most cases specific code must be written for every rendering engine. There is a possibility to define several different user interfaces and device specification in the same UIML document, but every rendering device must still be described for the user interface to work on it. There are several rendering methods that are supported such as, Java both awt and swing, PalmOS, HTTP, WML and VoiceXML, and more can easily be added.

Chapter 3

Evaluation

As we can see in chapter 2, there are many attempts to describe user interfaces in a device independent way. The problem is that all of them are designed in such a way that a user interface must be targeted at a special category of devices, for example computer with big screen. The only one not doing this is UIML, but it has its focus more on developing a way of describing user interfaces with the same syntax on all devices, not to find a general way to describe user interfaces.

3.1 Future work

The proposal of this paper is that a model based language [7] would be used to describe the user interface and then use an interpreter to render it. By just defining the models of the user interface, we can leave it to the device to represent these models in a matter that is logical for the user of the device. The problem that arises is to render these models as an user interface that is not only functional, but is also nice looking. If an application looks amateurish it will not be used and classed as not competent for the task it is supposed to do by many users. This way of describing the user interface would not replace the different ways of describing a user interface that exist today. It would instead exist on top of them and use the one that exist on the running platform.

Bibliography

- [1] Goodger Ben, Hickson Ian, Hyatt David, and Waterson Chris. *XML-based User Interface Language (XUL) 1.0*. <http://www.mozilla.org/projects/xul/>, 2001.
- [2] Tim Bray, Jean Paoli, C. M. Sperberg, and Eve Maler. *Extensible Markup Language (XML) 1.0*. World Wide Web Consortium, <http://www.w3.org/TR/REC-xml>, second edition, 2002.
- [3] K. F. Eusticea, T. J. Lehman, A. Morales, M. C. Munson, S. Edlund, and M. Guillen. A universal information appliance. *IBM Systems Journal*, 38(4):575–601, 1999.
- [4] James Gosling, Bill Joy, Guy Steele, and Gilad Bracha. *The Java Language Specification*. Sun Microsystems, Inc., http://java.sun.com/docs/books/jls/second_edition/html/j.title.doc.html%, second edition, 2000.
- [5] Abrams Marc and Helms Jim. *User Interface Markup Language (UIML) Specification*. Harmonia, Inc., <http://www.uiml.org/specs/docs/uiml30-revised-02-12-02.pdf>, February 2002.
- [6] Scott McGlashan, Dan Burnett, Peter Danielsen, Jim Ferrans, Andrew Hunt, Gerald Karam, Dave Ladd, Bruce Lucas, Brad Porter, Ken Rehor, and Steph Tryphonas. *Voice eXtensible Markup Language VoiceXML*. World Wide Web Consortium, <http://www.w3.org/TR/voiceXML20/>, April 2002.
- [7] Egbert Schlungbaum. Model-based user interface software tools - current state of declarative models. Technical Report 96-30, Georgia Institute of Technology, 1996.