BCS

human computer interaction 2003

BCS CONFERENCE SERIES

Eamonn O'Neill,
Philippe Palanque and Peter Johnson (Eds)

# People and Computers XVII – Designing for Society

Designing for Society
BATH
HCI 2003

Springer

This volume contains the full papers presented at HCI 2003, the 17th Annual Conference of the British HCI Group, a specialist group of the British Computer Society. The conference has become the premier annual conference on Human-Computer Interaction in Europe. Attracting researchers, practitioners, educators and users from all over the world, with interests in many facets of human-computer interaction, usability and interactive systems, these published proceedings form an important part of the archive of HCI research.

As advances in computing and communications technologies extend the human-computer interface beyond the desktop and into our clothes, streets and buildings, mobile and pervasive applications provide exciting challenges and opportunities. *People and Computers XVII - Designing for Society*, addresses the main areas of HCI research while focusing on its position and usage within today's society. The papers raise and discuss numerous questions, such as:

- How do we design for usability when human-computer interaction is dispersed and interwoven throughout our environment?
- How can we understand and account for the web of influences amongst society, environment and technology?
- How do we interact successfully with and through devices and networks with many form factors?
- And, how do we design these devices?

BCS

BCS CONFERENCE SERIES

ISBN 1-85233-766-4

9 781852 337667

ISBN 1-85233-766-4

www.springer-ny.com
www.springer.co.uk
www.springer.de

# *WebTouch: An Audio-tactile Browser for Visually Handicapped People*

**M Macías[†], A Reinoso[‡], J González[†], J L García[‡], J C Díaz[‡] & F Sánchez[†]**

[†] *QUERCUS Software Engineering Group and* [‡] *DSP Processing Group, Departamento de Informática, University of Extremadura, Escuela Politécnica. Avda Universidad s/n, 10071 Cáceres, Spain*

Email: *fernando@unex.es*

**The Internet offers new possibilities to the access of information, but sometimes the design of Web pages obstructs the contents making them inaccessible to everybody, especially for those people with visual disabilities. The problem has several sides. On the one hand, the inaccessible design of the pages. On the other hand, most of the browsers used to surf the net are thought to be managed by users without visual disabilities. Although there are tools to help in the right design of Web pages or in the interpretation of Web pages for people with visual handicaps, to our knowledge there is not an integrated tool useful for both, the designers and the visually handicapped users. Our research group has developed such a tool called KAI (Kit for the Accessibility to the Internet). KAI is based on two main pillars: a new markup language with accessibility features called Blind Markup Language (BML) and WebTouch, a multimodal browser taking blind people into special consideration. In this paper we focus on WebTouch and its two modalities for surfing the net: voice and tactile skills.**

**Keywords:** Web accessibility, Web browser, speech recognition.

## 1 Introduction

Since its creation the Internet has offered a new branch of possibilities to the access of information. Through this media, published data are available in the entire world at the same time. But sometimes the design of pages and Web applications are not accessible to everybody. People with disabilities often find difficulties when retrieving information from the net. Some designers and interfaces have replaced the functionality and simplicity with aesthetics and attractiveness; this obstructs the access to the contents, especially for people whose physic disabilities make them unable to enjoy with the design, the blind.

There are tools available to help this kind of users: screen readers such as JAWS[1], hearing browsers [2] and others. But these tools only exploit the hearing skills. Currently, haptic skills are gaining more and more interest in the development of user interfaces for people with disabilities [Sjostrom 2001; Brewster 2001; Challis & Edwards 2000].

Our research group has developed KAI, a Kit for the Accessibility to the Internet for the blind. One of the main components of KAI is WebTouch, a multimodal browser. Multimodal interfaces imply the use of multiple ways of interaction between the user and the computer. WebTouch allows the user to recognize the elements on a Web page by using the sense of touch. The activation of commands and the interaction with the different elements of the Web page is done via voice using an automatic speech recognition system. This multimodality allows the user to form an internal cognitive map of the Web page being visited [Lahav & Mioduser 2001].

In this paper the fundamentals of WebTouch are introduced. The rest of the paper is as follows. In Section 2 the main ideas behind KAI and WebTouch are outlined. Section 3 presents the voice recognition system. Finally conclusions are presented in Section 4.

## 1.1 KAI

KAI [Macías et al. 2002; González et al. to appear] is an integrated tool with software and hardware components. The main goal of this tool is increasing the accessibility of Web contents to users with visual handicaps, especially the blind.

The architecture of KAI (Figure 1) has two main components: the BML language and the WebTouch browser.

## 1.2 BML

The Blind Markup Language (BML) has been developed to build accessible Web pages easily. It is derived from XML and quite similar to HTML but including new tags to structure the page better. BML also allows the possibility of assigning different ways of presentation to the different elements of a Web page: textual, graphical, sound and tactile. Web designers can use BML directly or through an editor, which is accessible itself.

A Web designer can write code directly for BML with or without the editor. But, what happens with existing Web pages that are not written in BML? There is another component in KAI that translates existing Web pages from HTML or XHTML into BML by analysing their structure. This process implies several phases: separation of contents and presentation, recovery of inaccessible contents and reorganization of the information using the new tags included in BML. Once the page is coded in BML it can be presented in the WebTouch browser explained below. For other browsers

[1] http://www.freedomscientific.com/fs_products/software_jaws.asp.

[2] For example, Home Page Reader, http://www-3.ibm.com/able/hpr.html.

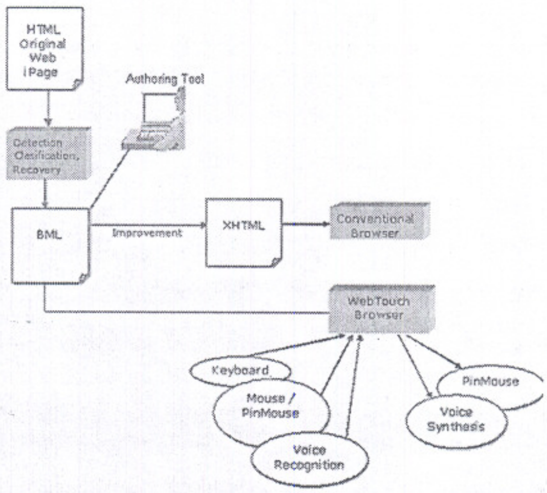


**Figure 1:** A general overview of KAI.

there is a module in charge of translating the page from BML into XHTML again. The original source code of the Web page has been modified. Now it is more accessible, but its visual presentation is the same.

## *1.3 WebTouch*

WebTouch is a multimodal browser especially developed for blind users. It provides several possibilities of navigation combining audio, voice and tactile skills. It is possible to configure the navigation modality according to users needs. Figure 2 shows a Web page in WebTouch. The different areas of the browser are also outlined.

WebTouch allows one to obtain a tactile representation of the contents of a Web page in the *Representation* area. In this way, visual handicapped users can build a mental map of the contents before surfing. This is an improvement comparing it with other tools such as screen readers. Each element of the page is presented as an icon. Each icon can be recognized by passing a special mouse (PinMouse) over it. PinMouse, also developed in the context of KAI, has two cells of pins like that shown in Figure 3. Whenever the mouse passes over a specific icon then the pins raise up forming a shape. The first cell of PinMouse tells the user the element below the mouse (for example, the presence of a table). The second cell provides additional information regarding accessibility issues, for example, the existence of a summary for the table detected. This helps the user to decide whether continuing surfing or not.

Once the icon has been selected then the user can interact with it using the *Interaction* area. For example, given a text, the user could activate the control for reading it. The user can also use the *Navigation* area to surf the page through the different structures: only tables, only e-mail addresses and so on. This is one of the main advantages of WebTouch: the possibility of selective reading.

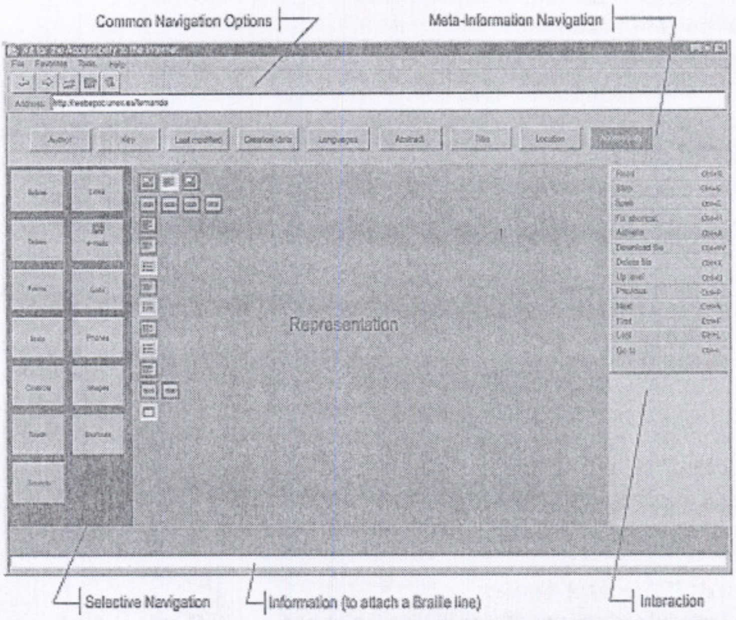


**Figure 2:** WebTouch interface and working area.

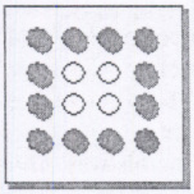

**Figure 3:** Tactile icon in a pins cell.

Both options and icons are associated with a keyboard combination and a word. So users can interact with the different areas with keyboard, mouse and voice. For this purpose, a speech recognizer has been added to KAI in order to extend the possibilities of navigation. Users can concentrate on a tactile exploration of contents while using their voice to manage the interface.

One of the main objectives in the design of KAI was the final price for the user and the ease of use. The tool is quiet cheap. The user only needs a special mouse and software. The experiments developed up to now demonstrate that the tool is very intuitive for users.

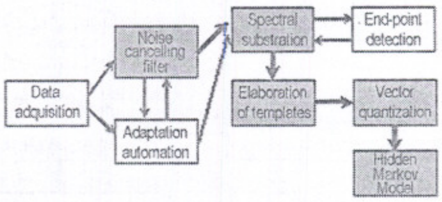


**Figure 4:** Architecture of Ivory speech recognition system.

## 2 The Automatic Speech Recognition System

We have considered Ivory [Díaz et al. 2001] as the automatic speech recognition system (ASR). Ivory is a speaker independent ASR methodology for isolated words that is robust to noise. Ivory was originally developed in C for the Texas Instruments TMS320C31 digital signal processor and it is maintained in the current processors of the TMS320C family such as the TMS320C6711. As can be understood, the speaker independence of Ivory makes an important feature for wide use applications such as those related with the World Wide Web. Internally, Ivory is based on a probabilistic model that structures the recognition process in a pipeline of stages that are applied to the voice stream (Figure 4). Each one of these stages is a transformation function.

A feature we have removed from Ivory is the noise robustness. This is due to the environments that it targets, mainly desktop systems, should not need it at all. Thus, the Ivory gets reduced to the four stages that Figure 1 shows. The Java use of Ivory has imposed to define clearly the original interface. Thus, the main functions of the Ivory interface are:

```
void IVORY_spectrum(Channel * spectrum,
    Sample * signal, Energy * energy) ;
void IVORY_template(Template * template,
    Channel * spectrum, Average average[][]) ;
void IVORY_quantization(Centroid * centroid,
    Template * template, int context) ;
void IVORY_parser(Winner * winner, Centroid centroid,
    Energy energy, state_t state, int context) ;
```

The three first stages extract the more significant features of the speech signal. The last one compares these features to the statistical models that have been previously elaborated, the so-called vocabulary of the recognizer. In the case of WebTouch the vocabulary is the set of buttons in the different areas and the different kinds of icons. If the features of the input signal match one of the built-in models, Ivory takes as recognized the word corresponding to that model. The statistical model of each word of the vocabulary is known as the Hidden Markov Model and it has to be previously built. Several samples (observations) of each word are needed. Altogether they form what it is known as the word database, which is later used to construct the set of Hidden Markov Models associated to the vocabulary.

On the interaction with any application, the user usually has available a set of active options or commands. If we define 'work context' as the current state in the handling of the application, the work context comes defined by the set of valid

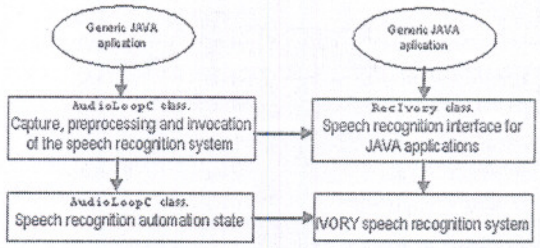


**Figure 5:** ASR component architecture.

commands. Thus, the work context can be mapped to a given vocabulary, called **active vocabulary**, composed by the set of words used to name such commands. In the case of WebTouch the meaning of a word is different depending on the area being interacted with.

A feature that has been added to Ivory is the work context. An advantage of this focus is that the more reduced the vocabulary, the more precise the recognition task results. The work context has implied to extend the interface of Ivory. Thus, the quantifying stage will use the work context in order to determine the centroids codebook. Similarly, the parsing stage uses the context information to choose the right Markov model.

## 2.1 *The Architecture of the ASR Component*

The ASR component consists of two main modules or classes, *AudioLoopC* and *RecIvory*, and a third auxiliary class, *ParserState*, imposed by the work context concept. *AudioLoopC* runs an infinite loop of voice caption and supplies it to the recognizer. *RecIvory* acts as the Java interface to the C functions of Ivory. *ParserState* is used to keep the state of an automaton that manages the voice and silent periods. Figure 5 illustrates the modular design of the component.

It is important to highlight that the execution of the ASR component does not interfere at all with the rest of the application functionalities. This design allows WebTouch to be controlled both by the usual devices such as the mouse, the keyboard and by the human voice.

### 2.1.1 *The AudioLoopC Class*

*AudioLoopC* constitutes the main class of the component. It implements the functionalities of capture and preprocessing of the voice stream and the further invocation of the Ivory pipeline. Likewise, it eventually invokes the method that the user application associates to the recognized word.

The basic task of its constructor is the aperture of an input stream that allows the application to receive the audio data. For portability and efficiency reasons, we have chosen to obtain the stream in a direct way from the general controller of the audio resources of the system, thus avoiding the treatment of specific drivers. This general controller considers the microphone as the standard audio input, which automatically provides the class with a voice stream coming form this device. Once the stream has been obtained, it is opened indicating the audio format of its data and the size of the requested buffer.

*AudioLoopC* has been designed and implemented in such a way that the whole process of capture, treatment and recognition is executed in the context of an independent thread of control running in concurrency with the rest of the threads or activities launched by WebTouch. The activation of this thread is performed by invoking a method of the *AudioLoopC* class. This method activates the stream and initiates the cycle of capture and recognition. The signal is normalized and fragmented in overlapped windows of 256 samples before being supplied to Ivory.

The current work context is stored in the `context` attribute of the *AudioLoopC* class, which provides methods for reading and writing. When a word is recognized, *AudioLoopC* makes a callback to WebTouch to serve the event.

### 2.1.2 The RecIvory Class

*RecIvory* links *AudioLoopC* to the C Ivory interface. It is a class whose methods are declared as natives in order to indicate that its implementation is not done inside the own class and, furthermore, that it has been written in a language different from Java. Each of these methods maps an Ivory function. The instantiation of the class is not necessary because it is just an interface class. For the same reason all its methods are defined as static.

The object code of Ivory is supplied to *RecIvory* loading in memory the dynamic library that results from the compilation of its native routines. This loading process is performed in the class initialization section.

Finally, the class *RecIvory* supplies a mechanism that is independent from the voice capture procedures. This makes it possible for any application to use it in an independent way.

## 2.2 Using Ivory from Java

Ivory is a DSP application developed in a TMS320C6711 digital signal processor. Code Composer is a C/C++ development environment for these processors. As a result, Ivory is implemented as a C module. The problem posed is using Ivory, a compiled C library, from *RecIvory*, a JAVA class. Fortunately, JAVA helps with the Java Native Interface (JNI). JNI is the Java programming interface to native code. JNI allows Java code, written to run on the JVM, to interoperate with applications and libraries coded in other programming languages, such as C, C++ or assembler. It also allows embedding the JVM in these native applications. This section deals with two issues, the renaming of the Ivory methods and its pointer arguments.

The first problem posed by JNI is that it imposes a renaming of the original Ivory methods. The source file *RecIvory.java* contains the declarations of all the original native methods of Ivory, as well as the load sequence of the resulting library. This file is compiled with the javac tool in order to obtain the *RecIvory.class* file. The javah tool with the -jni option is applied to it to obtain a header file of native methods. As a result we get the header file *RecIvory.h*, with the renamed Ivory methods. Once the native methods have been adapted, we proceed with its compilation. In system such as Solaris a shared library is obtained. In windows systems a dynamic link library is used.

The second problem that this section deals with is the parameter passing to Ivory. Ivory needs to operate upon arrays that contain windows of samples from the voice stream. This means that the access to the elements of an array and its manipulation ability from a native C method is a fundamental issue.

JNI allows specifying if a native method either access to a copy of the array kept by the user Java code or reference directly to it. The real-time requirement of speech processing demands that any interchange between Java and Ivory be performed by reference.

The treatment with matrices deserves a special consideration. Some methods of Ivory receive as argument a matrix of floating point data. Although the declaration and use of this matrix in the Java code does not present any difficulty, the same does not happen on the native side. The procedure of access to the elements of the matrix in the native code come determined by the consideration of it as an array of elements that, in their turn, are collection of data. This view of the matrix is supported by JNI by the general type *jobjectArray*, that allows to declare arrays that contain references to objects instead of data of a primitive type. Thus, the Ivory functions that need the matrix of templates will do it through the corresponding parameter declared of *jobjectArray* type.

### 2.3 Integration of the ASR System with WebTouch

This section presents the procedure that allows integrating the component in a user application such as WebTouch. In first place, the classes of the component are added to classes belonging to WebTouch. In second place, the main class of WebTouch declares and instances an object of the *AudioLoopC* class, providing the format of the audio stream. Finally, at run-time, WebTouch invokes the method *AudioLoopC*, that activates the thread that starts the activities of voice capture.

Eventually, WebTouch provides Ivory with a new work context, depending on the user actions. This information is supplied through the method that *AudioLoopC* provides with this purpose. When Ivory recognizes a word returns the identifier of the word to *AudioLoopC* that invokes the method of WebTouch associated to the word.

## 3 Conclusions

KAI, Kit for the Accessibility to the Internet, has a multimodal haptic/voice browser, WebTouch, developed to be used mainly by the blind.

From the haptic point of view, this browser contains a special mouse, PinMouse, which allows the detection of elements of a Web page using the sense of touch. From the voice point of view, the browser allows an interaction using an automatic speech recognition system (ASR). This paper has briefly presented the main ideas behind the ASR system and its integration with WebTouch.

Future works include the possibility of integrating features of initiatives such as Speech Application Language Tags (SALT[3]) in BML. With this, not only the user can interact with the different commands of the browser but also with the page itself.

[3] http://www.saltforum.org/

## Acknowledgement

This work has been developed with the support of CICYT under project TIC2002–04309-C02–01 and Junta de Extremadura under project 2PR01A023.

## References

Brewster, S. [2001], Haptic Interaction, http://www.dcs.gla.ac.uk/~stephen/research/haptics/index.shtml (last accessed 2003.05.26).

Challis, B. P. & Edwards, A. [2000], Design principles for Tactile Interaction, http://www.dcs.gla.ac.uk/~stephen/workshops/haptic/papers/challis.pdf (last accessed 2003.05.26). Proceedings of the Workshop on Haptic Human–Computer Interaction, University of Glasgow.

Díaz, J. C., García, J. L., Rodríguez, J. M., Álvarez, J. F., Espada, P. & Gómez, P. [2001], DIARCA: A Component Approach to Voice Recognition, *in* P. Dalsgaard, B. Lindberg & H. Benner (eds.), *Proceedings of the 7th European Conference on Speech Communication and Technology (Eurospeech 2001)*, Aalborg Universitet, pp.2393–6.

González, J., Macías, M., R., R. & Sánchez, F. [to appear], Accessibility Metrics of Web Pages for Blind End-users, *in* J. M. Cueva, M. González, L. Joyanes, E. Labra & M. P. Paule (eds.), *Proceedings of the International Conference on Web Engineering (ICWE'2003)*, Lecture Notes in Computer Science, Springer-Verlag.

Lahav, O. & Mioduser, D. [2001], Multisensory Virtual Environment for Supporting Blind Persons Adquisition of Spatial Cognitive Mapping: A Case Study, http://muse.tau.ac.il/publications/74.pdf (last accessed 2003.05.26).

Macías, M., González, J. & Sánchez, F. [2002], On Adaptability of Web Sites for Visually Handicapped People, *in* P. De Bra, P. Brusilovsky & R. Conejo (eds.), *Adaptive Hypermedia and Adaptive Web-based Systems: Proceedings of the 2nd International Conference (AH2002)*, Vol. 2347 of *Lecture Notes in Computer Science*, Springer-Verlag, pp.264–73.

Sjostrom, C. [2001], The IT Potential of Haptics. Touch Access for People with Disabilities, Licenciate Thesis, Certec, Lunds Universitet. Available at http://www.certec.lth.se/doc/touchaccess/ (last accessed 2003.05.26).