

A Tool for Adapting Visual Interfaces to Blind People

Siwar Farhat¹ and Christian Fluhr²

¹*Institut National des Jeunes Aveugles, 56, blv. des Invalides,
F-75007 Paris, France*

Phone: +33-1-44-49-35-35 – Fax: +33-1-44-49-35-36

E-mail: 100773.1624@compuserve.com, felkateb@tabarly.saclay.cea.fr

WWW: <http://www.cea.fr/>

²*Institut National des Sciences et Techniques Nucléaires, DIST/SMTI Bâtiment
528, Centre d'Etudes de Saclay, F-91191 Gif Sur Yvettes, France*

Phone: +33-69.08.70.93 – Fax: +33- 69.08.26.69

E-mail: fluhr@tabarly.saclay.cea.fr

Abstract

A graphical user interface generally uses standard components such as windows, menus, edit boxes, list boxes... offering improvements in human-computer interaction exploiting the abilities of sighted users. So, blind people using this kind of interfaces cannot profit from these advantages. That is why these interfaces should be adapted for them. This can be done either by using a standard method for all the application screens or by individualising the adaptation of each application. This makes it possible to optimise the manipulation of interfaces by blind people but it is complicated because there are many versions of the any one product and it is often necessary to readapt them. The new approach is to make the adaptation simple and easy through an authoring system which adapts visual interfaces. The behaviour of the interfaces can be altered according to different situations, even those which were not foreseen. This system can modify completely or partially the visual interface by adjusting it for the visually handicapped using a database of rules and rebuilding it as a multimodal interface.

Keywords

Adaptation tool, non-visual interface, human-computer interaction, multimodal, user model, authoring system, blind people.

Introduction

Until recently, interfaces contained information displayed in ASCII characters. Visually handicapped persons were able to use these interfaces, thanks to software which reads the screen buffer and transmits information to a speech synthesiser and Braille terminal. Graphical interfaces forced the technical solutions used since the eighties to change, so that blind people could use them. Access to these interfaces and interactivity between humans and computers requires new methods and

different conceptions. Therefore graphical interfaces should be rebuilt and adapted them to blind people's behaviour and capabilities.

The developed system consists in realising an adaptation of any software running under Ms-Windows. This adaptation does not force blind people to be submitted to the interactivity mode of Ms-Windows. The project's goal is the accessibility of software to visual handicapped persons in order to facilitate their social and professional integration. Different axes are defined to realise this project. Their essential goal is to answer the following questions:

- Who will be using this interface ? What will their visual constraints be and how will they work with software tools ?
- How can we modify an existent graphical application's interface so that it considers their different handicaps ?
- Which are the conceptions and techniques that ensure these adaptations ?

This project has several steps and proposes to resolve:

- The problem of access to the information displayed on the screen.
- The way to reconstruct the screened information according to the user's profile.

We will describe a certain number of systems already used in this field, the contribution and the advantage of our approach in comparison with the existent situation. This project is funded by I.N.J.A (Institut National des Jeunes Aveugles).

1 Description of Visual Interface: Environment

Ms-Windows is a graphical environment that allows conviviality and many other possibilities that do not exist in Ms-Dos.

- It allows many applications to run simultaneously (multi-tasking).
- It is independent from the hardware.
- It allows communication between applications (for instance, by Dynamic Data Exchange).

There are many tools that allow the creation of applications running under Windows and its interface for seeing users. They are all efficient and contain ways to build interfaces. The programmer chooses a tool suited to his needs and knowledge.

Example. Borland C++, Visual C++ or Visual Basic generate easily a variety of graphical interfaces. Each one has its own graphical interface generating tool such as Borland C++'s workshop or Visual C++'s AppStudio. An application's interface has dialogue boxes containing several controls (list box, edit box, combo box, static and button controls). It also has menus, icons, bitmaps, tables and text,... Every Windows application has some basic objects (Windows predefined controls). Their behaviour is the same in all applications and can be generally captured with their content. Some particular cases will be explained later.

Two dialogue boxes, belonging to the same application and containing controls such as list boxes, send and receive the same messages; however their initial behaviour can be changed. Widgets also communicate with the other applications exchanging messages. To enable visually handicapped people access to standard Windows applications presents two problems:

1. First, how to access, in real time, data used by the Ms-Windows operating system (windows, windows content and messages).
2. Second, how to represent this data to « non-seeing » users. It is hard to translate visual display into Braille or speech synthesis. This is why specific models of communication are developed and propose an interface adapted to blinds' problems. These models consider habits and mental representations of blind persons.

2 Data Access: General Filter

In order to allow blind users to interact with graphical interfaces, a great deal of data must be extracted from the graphical environment. The data extraction is done by a filtering system. This system intervenes between the application and the final user. The user's actions and the data sent to peripherals (such as the screen) are filtered. Data is saved in appropriate data structures. A screen model is built and refreshed in real time. Different types of filters are necessary to filter a whole Ms-Windows application:

- a filter of messages sent and received by the applications;
- a filter of widgets (e.g., dialogue boxes, buttons, edit boxes);
- a filter of windows contents (e.g., bitmaps, tables, text).

The general filter aims to retrieve a maximum of the events generated by either the application or the user. Our approach is to select certain events via a specialised filter in order to initiate the application. It differs from that of others, e.g., Guib Project and Mercator in that the specialised filter creates a database for the adaptation tool's use.

3 Information Retrieval

There are two possibilities used in our application to represent extracted information and to allow an appropriate utilisation of applications:

- A basic model built by our application. This information model is static in the developed program and has the same behaviour in all applications. It describes user interactions and the application interface.
- A specific model created by the adaptation tool. This tool is another interactive system. It will be the focus of our paper and its utility and description will be examined below.

There are many problems with the adaptation of visual interfaces for blind people:

- identification of the main characteristics of blind people and their interaction with computers in a graphical environment [Fellbaun94];
- problems linked to accessing graphical data;
- how to rebuild non visual messages out of those which are designed for a visual interface;
- finding the most appropriate software design and environment for creating multimodal interfaces and adapting of existent graphical interfaces;
- how to transform display to speech synthesis or to Braille, and at which moment to present them to the user;
- how to mentally represent the structure of documents;
- how to use Braille, speech synthesis and vocal recognition, singly or in combination to represent the different types of information;
- how to rebuild tables, images, texts.

The last two points will be developed in a separate paper.

4 Existing Products

Below are examples of existing products. Having presented them we will move on to the exposition of our own work.

4.1 VisioBraille

This system filters different widgets [Handialogue94]. This allows the user to pilot standard screens with particular commands which are the same for each application. Standard behaviour is given to non standard screens by the developer. The particular commands of a software are practically non existent.

4.2 Guib-Access

This approach consists in saving widgets and their contents in a data base (Off Screen Model). This data base can be explored by using functions.

An adaptation of a particular software needs programming which is long and needs to be done by an experienced programmer. The technique of formulating requests to the data base slows down the system [Outspoken89].

4.3 JAWS (Job Access With Speech) Windows Version

This software allows the adaptation of visual interfaces for blind people. It creates macro-commands that will be executed when the application is running. A macro is a sequence of actions. These are grouped together in one operation. A key combination executes the macro. For instance, the following macro means: if the « Page

Down » key is activated, the words “Page Down” are pronounced [Henter-Joyce95]:

```
MacroBegin
{page down
saystring(“page down”)}
MacroEnd
```

4.4 Guib Project (Textual and Graphical User Interfaces for Blind People)

The adaptation of the two chosen environments (Ms-Windows and X-Windows) is based on [Fellbaun94]:

- interfacing of suitable peripherals ;
- filters that extract information from the graphical environment ;(general filter)
- screen readers for the presentations of information.

There are different screen readers that have to be written for different environments, to match differences between them, taking into account the organisation of the dialogue in the human-machine interfaces and the effectiveness of implementable filters. There are three screen readers for example for Ms-Windows:

1. Screen reader based on the GUIDE display.
2. Screen reader based on the virtual Braille display.
3. The speech-only screen reader.

5 Our Approach: an Adaptation Tool

5.1 Description of Adopted Approach

We can adapt by either giving the application screens a standard behaviour, i.e. a general behaviour, or by individualising the adaptation of each application. This makes it possible to optimise interfaces for blind people but it is complicated and time-consuming because there are many versions of any one product and it is often necessary to readapt them.

A new approach is to make adaptation simple and easy through an authoring system, this represents the adaptation tool adapting visual interfaces. The behaviour of the interfaces can be altered according to different situations, even those which were not foreseen. These changes are introduced into the system by the sighted user (the author) in a flexible manner using the authoring interface which allows the association of macro instructions with events coming from the users and/or application. The authoring system will generate the database of rules.

This system can modify completely or partially the visual interface by adjusting it for the visually handicapped using a database of rules and rebuilding it as a multi-modal interface.

The information can be given in vocal or Braille form depending on the users' needs and visual deficiency level. This authoring system can be used by experts or ergonomics.

Let us exemplify adaptation:

- If a software package sends a warning message which contains important information to the screen, the software will put the message in evidence. An adaptation system that cannot adapt its interface to a particular software, will not be able to establish the presence of a warning message, and will not be able to inform the blind user. The blind user would have to explore the screen to notice the presence of this warning. In MS-Windows, we cannot identify this type of message. The event will be translated by the adaptation tool, because the author knows its meaning¹.
- We may give the example of the Harrap's French-English dictionary. The only thing that distinguishes an English word from a French word is the character typography (normal or italic). An adaptation system that explores text windows in the same way, cannot differentiate between the two languages and cannot associate each typography to the relevant language. The consequence is that the speech synthesis will be meaningless, in one of the two languages, because the system does not notify the speech synthesis which language it is to use. This kind of problem may be solved by using our approach.
- The adaptation tool establishes, if necessary, links between different controls, which permits a system based not only on the controls' display but also their meaning. These links are used to mention audibly, for example, that the listbox in the "Open" dialogue box contains the files of the current directory. The static control "List of files" is associated to that listbox. This kind of help is not pertinent for sighted people, because they can see a complete representation of the screen, which is not the case of the blind user.
- The rendering of information captured from the interface can be modified. For example, when the blind user calls down the menu of his application he can receive menu items and information that differ (in order to make them more explicit) from those displayed on the screen. The author can modify the commentary displayed information, reducing or adding explanations, depending on the blind user's needs. We can even change the application's language by adding a translation tool.

¹ The author is a sighted person who uses the adaptation tool.

A particular adaptation allows for different behaviours for different users. Let us consider these two examples:

- an adaptation for novice users : talkative and close to the commands used in the visual application, so the user can explore the software and have an easier collaboration with the sighted users.
- an adaptation for expert users : user shortcuts and make the blind user more rapid than the sighted user.

The interface objects in Ms-Windows have a rudimentary organisation which corresponds to their order of creation. This organisation can not be easily accessed by blind people.

So how can we represent window components in an efficient way? Must we change the initial functioning of the application to adapt it to the blind users or must we keep it?

It was necessary to ask potential users and have their opinion in order to develop a conception of a convenient system for the blind. A preliminary study has proven that everything depends on the user.

The user who knows Windows prefers to keep the original functioning, organisation and commands. Other users wish to have a different functioning. User models are then necessary, which can be defined as data structures containing the available and pertinent information about a class of users.

5.2 The Adaptation Tool Characteristics

5.2.1 General Aspect

Our project aims to give the visually handicapped the means to access graphic interfaces via the hook system and the adaptation tool. The main goal consists in:

- Making a model of the interface concepts, in order to have, at anytime, a description of the software environment.
- Presenting an adapted interface depending on the degree of handicap. Different actions can be done to realise this aim, for example:
 - ignore certain messages of the application;
 - lock, send or modify other messages;
 - add or remove controls;
 - add specific processing to controls, dialogue boxes, menus,...
 - add specific help for the user : description of the interface or the use of controls, the presence of warnings or error messages.

The following figure describes the general aspect of our project (figure 1) :

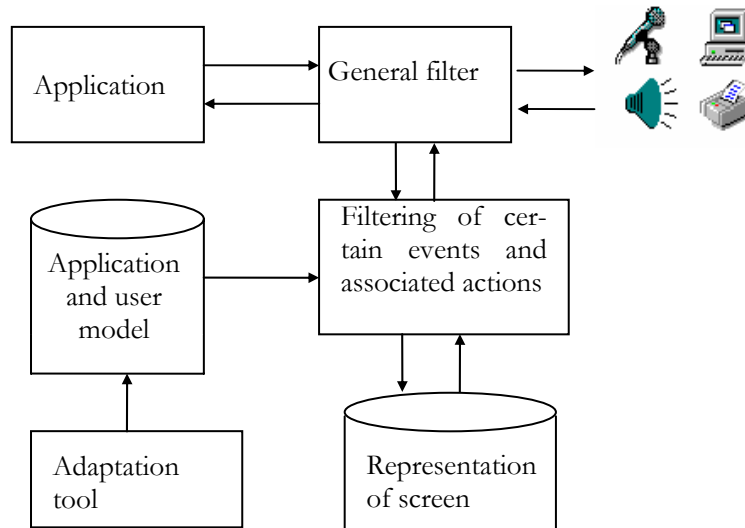


Figure 1. The developed system integrating the filter and tool adaptation

When the user is communicating with the application via a keyboard, the data flow passes through a hook to be analysed and processed according to the user model created by the adaptation tool. This is a rule based hook constituted by the proposed tool.

Example of modification of an initial processing after having used the tool adaptation. The author decides (for a specific class of users) to replace the use of the "tab" key by that of the direction arrows in order to move from one control to an other inside a dialogue box (\leftarrow , \rightarrow , \uparrow , \downarrow). Keys \leftarrow , \rightarrow allow the passage from objects of the same group, to another group. Keys \uparrow , \downarrow allow the passage from one object to an other inside a same group. This is a hierarchical exploration method that permits interface standardisation.

5.2.2 Multimodality and Ergonomic Needs

The interface is the most important part of an application. Its conception is a fundamental activity in the software production, because its success has a determinant influence on the software's success. We must take into account ergonomic needs in order to lead to the conception of the interface. In this case, the adaptation tool will solve certain ergonomic problems.

Taking the user into account. For ergonomist and psychologist, the human-machine interface represents the cognitive and physical phenomena that occurs in the realisation of computerised tasks [Coutaz90]. The interactive system developer should draw up a full description of the user cognitive processes and should give an accurate report of their representation in the software [Coutaz90].

Choosing the situation adapted mode. Multimodality is a means of improving software systems for the visually handicapped. It is one of the directions in which human machine interface is oriented. So, the system of multimodal interfaces offers the possibility for users to have interaction strategies adapted to their needs [Burger92]. The channel for transmitting messages, depends on the circumstances. A warning message should be transmitted via a speech synthesis system, while relevant information should be transmitted via a Braille display. The use of only one sensory channel, (sense of touch or sense of hearing) is restrictive. At the moment, the developed system restores information as speech synthesis.

Information about the context. Another aspect, is to give information about the context in order to improve any poor information. In this project, we reproduce this context by giving users information about:

- the type of the current processing;
- the different widgets to be manipulated;
- the different commands to be executed;
- messages sent by the application;
- the content of the different widgets;
- the different operations that users can execute at a given time.

5.2.3 Technical Aspect: Using the Authoring System

The author executes the application to be adapted in order to create a user model. This application runs under the control of the author system. To create an adapted interface, he will be able to explore all the application steps and associate macro-instructions to controls.

Description of the steps of the adaptation process:

- adaptation tool places a filter to detect present windows;
- the controls displayed on the screen will be analysed automatically;
- when a main window is detected, all the controls and the menu (if it exists) are extracted and saved in well-defined structures;
- each main window is characterised by its title, type, and the number of controls it contains;
- each window runs dynamically in memory, its deletion or creation induces the removal or addition of structures;
- when a main window is visible on the screen, the author has the possibility to use a macro-command (keyboard key or mouse button) to associate a specific processing to each of the window's controls for a given event;
- the author chooses one of the Control from the filtered objects list in order to match it with a sequence of instructions;
- s/he activates the button « Event management » and another window appears (figure 5);

- s/he selects the element to treat and associates a macro-instruction.

This last point can be repeated by selecting other events for the same control. In the same way, we can iterate the same treatment for other filtered-controls:

- the specific processing is a sequence of instructions called « macro-instruction »;
- the macro-instructions is defined according to the visually handicapped user's needs;
- the specific processing of controls can be changed dynamically during the execution of the application;

The following figure describes the functioning of the adaptation tool's system (figure 2)

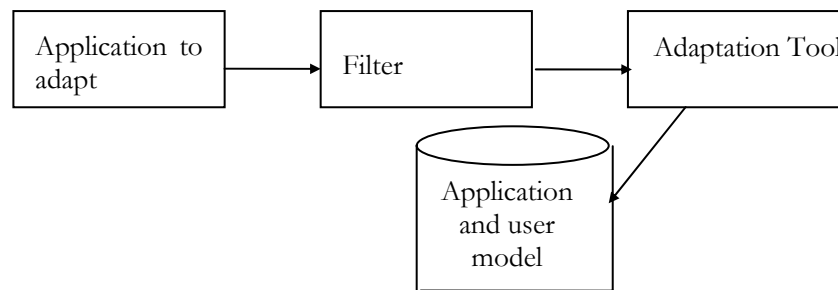


Figure 2. Adaptation tool

The Adapting Tool generates an internal structure for an application according to a given profile represented in figure 3. In this figure, the « Dialogue Box File » contains the sequence of dialogue boxes which the author has associated with treatment. The « Sequence of events and instructions for one control (file 3) » consists of a rule based system containing the sequence of events which one matched to macro-instruction. The internal structure of file 3 is : Event1 Instruction1 Parameter1, Parameter2 ; Event2 Instruction 1 Instruction 2 Parameter1,...

The different versions of a given application can easily be changed according to the user. The Adaptation Tool generates a rule based system standing on the macro-instructions adapted to a specific profile : novice, trained, or expert. Using the Filtering System, the user selects his/her profile and the desired application, the Execution System loads according to this profile the corresponding Dialogue Box File (DDB).

This DDB is composed of specific processing associated to controls that the author has chosen to execute operations. Hence, for any application, a directory is created by the Adaptation Tool in which the rule based system is generated depending on the user's profile. This rule based system will be automatically loaded at every execution of the given application. Moreover, a change of this system will be done if the user selects an other profile or an other application. The Base generation was achieved through specifying events processing using the various interfaces proposed by the Adaptation Tool.

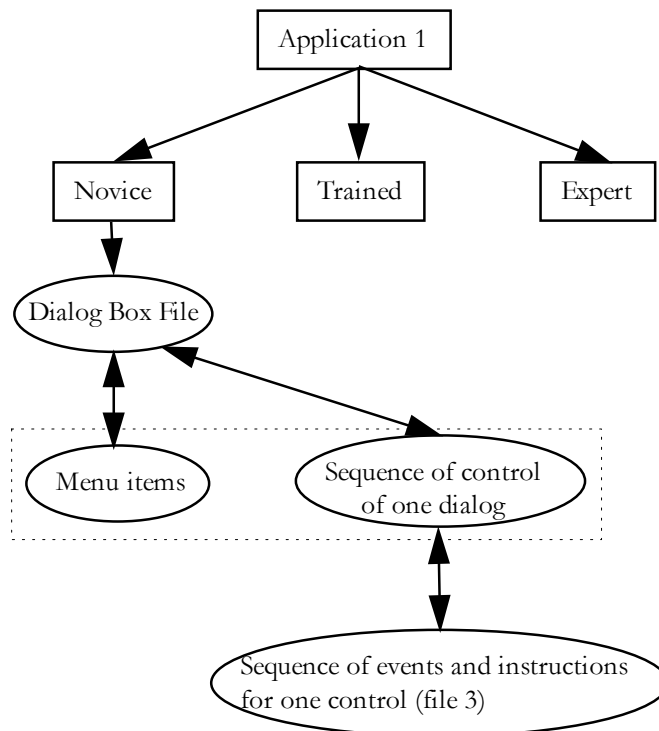


Figure 3. Application profile.

It is user-friendly because there is no code to generate and to compile. Actually, it only needs a scenario conception using the Adaptation Tool Principle that is proposed and based on the interaction between the Tool interface and the sighted author.

Example of Macro-Instructions:

```

CLASS-Control " Edit Box" ID==100
  Case event of:
    " ACTIVE ": Treat1-List(100); // Handle is an identification of list box
  END Case
END CLASS;

```

Function Treat1-List(Identification)

```

Begin
SayText(" Utility of this edit is to show different actions than you can do ")
Wait(500);
SayText ("The Content of Edit is").
Handle=RecupWindow(Identification); //Returns the handle of the window
ReadContents(Handle, Buffer); //Reads the contents of the EditText
DisplaySon(Buffer);

```

```

HandleButtonOk=RecupWindow(1); //Returns the handle of the button Ok
                                // of the dialogue box
SendClic(HandleButtonOk, x, y, BUTTONLEFT);
                                // Active Button OK, that allows to close the dialogue
                                // box that contain the edit box.
End

```

The description of the adaptation tool's mechanism is:

1. The macro-command gives the sequence of the controls displayed in the main active window (figure 4).

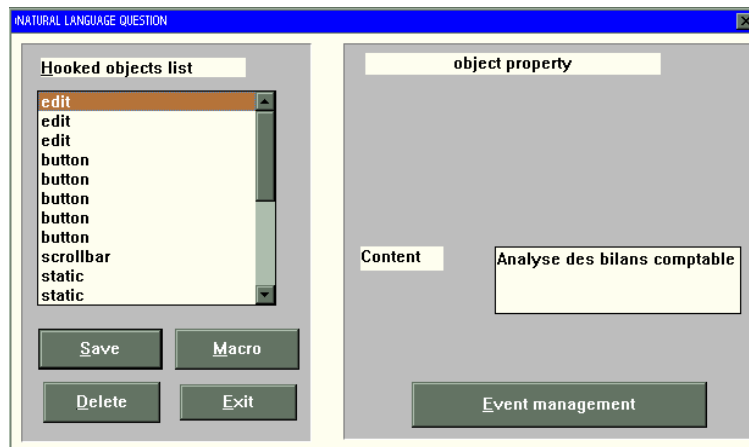


Figure 4. Interface that gives the sequence of a listbox controls

2. Definition of a macro-instruction to process a given event (figure 5).

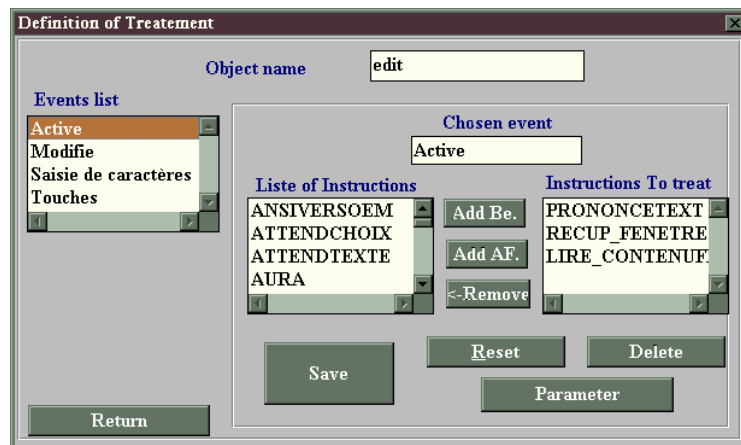


Figure 5. Interface to define a macro

5.3 Rebuilding of the Visual Interface and Information Retrieval: Example

We already mentioned that the rendering of information depends on the user's handicap level, so we thought of two types of retrieval : one adapted for the blind users and the other for visually deficient users and we defined three types of user profiles, novice, expert and intermediate. In this project we only took the blind user case into account.

5.3.1 Control Restitution for Blind People: Dialogue Box Representation

The adaptation of a dialogue box for a given profile depends on the information it contains. A dialogue box generally contains a sequence of controls or windows that are themselves, dialogue boxes. Each widget can contain other controls, or make references to other controls.

Here follows a description of a multimodal scenario to adapt a "Spirit" dialogue box. (Spirit is an information retrieval system using natural language queries). The dialogue box appearing after selecting a data base (figure 6).

The dialogue box consists of:

- EditBox to enter the question.
- EditBox to show keywords of the question.
- EditBox to show wrong words.
- Five buttons, each corresponding to application functions of the application.

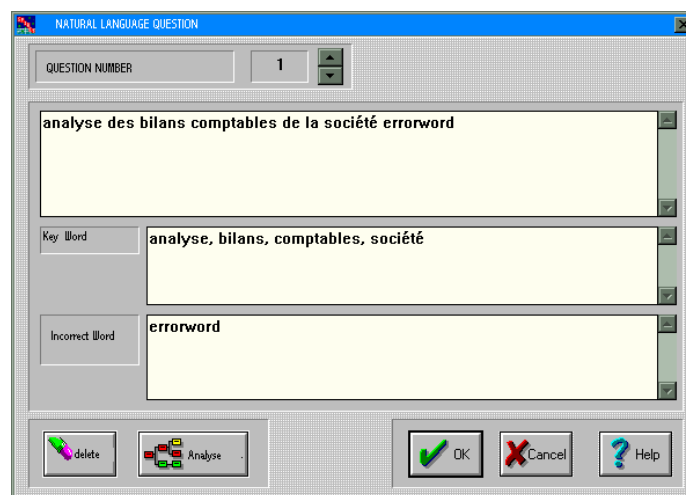


Figure 6. Spirit dialogue box : Question in natural language

We notice that one control (EditBox) has three different uses. We can filter information from each control, but we cannot describe the meaning of each control. For example, we cannot know that words contained in the third EditBox represent

wrong words, this is not a problem for the sighted user who sees the static field (« Incorrect words ») associated with the EditText.

To solve this problem, the author can ask the application to mention via a warning message the changes of the EditText contents and send the cursor to the first incorrect word in the EditText to make a search easier for the user. This kind of help is different from application to application. Here is an example of multimodal scenario:

- When the dialogue box « Question in natural language » appears on the screen, the text describing the utility of each control and the action required to initiate it is pronounced.
- For the EditText « Incorrect words », if the content changes, the following macro-instruction is executed:
 1. SayText ("There are incorrect words in the question").
 2. ReadContents(Handle); //Reads the contents of the EditText
 3. DisplayBraille("first incorrect word in the list");

This scenario corresponds to a rule for a given event.

5.3.2 Document Retrieval for Blind People

Actual retrieval system development is dominated by the conception of hypertexts allowing easy and fast access to huge databases. These hypertext links are evidenced by a particular typography. The information retrieval system « Spirit » has the advantage of producing less bulky documents than those in Braille.

A text reading system has been developed to break down the information into words, sentences or paragraphs. For example it also allows the activation of the hypertext link at the request of the blind user by the simulation of a click. The system also allows for reference to change in typography if necessary. This system is currently being developed and will be the subject of further articles.

6 Generalisation of the Developed System

The Adaptation Tool which represents a helping system to visual interface design allows to specify a given application. Indeed, our filtering system corresponds to a generic application which use can be specified through the helping system use.

The Filtering System generally retrieves information displayed on the screen at a specific moment T : text, pictures, dialogues boxes, typography, warning messages,... The base generated by the Filtering System could be exploited according to our needs using the authoring system that allows to adapt the application to be executed. The interfaces of this one will be rebuilt in accordance with our usage purpose, the user's needs or a given profile.

We can adapt its GUI and restore the information under different forms. For example, from the generated base, we can choose to reconstitute the information for blind people as a speech synthesis or a braille message.

For impaired people, the restoring could be the enlarging of the GUI depending on a coefficient defined by the user, e.g., his/her colour preferences. Impaired people differ from each other by their perception of colours.

We can even use this base for other aims as multi-linguism, and computer-assisted teaching. We can generate a teachware from the information base : add to the original application an on-line help, and many learning scenarios of the application use. The student could backup any problem encountered during his/her learning via annotations in order to show them to the teacher. As described before, the generic application can create several usage possibilities for a chosen application (e.g., for Microsoft Word®).

Conclusion

The adopted approach seems to be conclusive. A prototype has been realised and evaluated by blind users at the I.N.J.A. (Institut National des Jeunes Aveugles) in Paris and at the C.E.A. (Centre d'Etudes Atomiques). Testing the adaptation of some applications shows that the authoring system lacks some functions : cursor management in word processor applications, taking into account MDI applications.

This system could become an industrial product ready for sale though it needs to be tested. We would suggest that specialists in software adaptation for blind people test it in order to create new adaptations using this system.

Realising software adaptation can be rapid and simple. This should provide the means for progress in ergonomist research in the field of blind users. In the current version of the project we only considered input through keyboard, but in future version we will also consider speech recognition.

Reference

[Burger92] Burger, D., *La multimodalité: Un moyen d'améliorer l'accessibilité des systèmes informatiques pour les personnes handicapées*, in Proceedings of ERGO-IA'92 (Biarritz, 7-9 October 1992).

[Coutaz90] Coutaz, J., *Interfaces Homme-Ordinateur Conception et réalisation*, Dunod Informatique, 1990.

[Fellbaun94] Fellbaun, K., Crispian, K., *Interface vocales et auditive destinées à des utilisateurs non-voyants*, in Proceedings of « Interface multimodale pour handicapés visuels » (Paris, 7 November 1994), pp. 21-34.

[Handialogue94] HANDIALOGUE, *VisioBraille, système autonome et connectable sous MS DOS et MS-Windows pour non-voyant*, Reference manual, Paris, January 1994.

- [Henter-Joyce95] *JAWS for Windows*, Technical Reference, Henter-Joyce, January 1995.
- [Outspoken89] Berkeley system, *Outspoken for Macintosh*, Reference manual, 1989.