A Framework for Blind User Interfacing

Fernando Alonso, José L. Fuertes, Ángel L. González, and Loïc Martínez

Facultad de Informática, Universidad Politécnica de Madrid, Campus de Montegancedo, 28660 Boadilla del Monte, Madrid, Spain
{falonso, jfuertes, agonzalez, loic}@fi.upm.es

Abstract. There are specific usability requirements that have to be met when developing dual interfaces, that is, graphical user interfaces that are adapted for blind users. These include task adequacy, dimensional trade-off, behavior equivalence, semantic loss avoidance and device independence. Consequently, the development of human-computer interfaces that are based on the task, domain, dialog, presentation, platform and user models has to be modified to take into account these requirements. This paper presents a framework that includes these requirements, allowing for the development of dual interfaces. The framework includes a set of guidelines for interface design, a toolkit for the low effort implementation of the user interface, and a programming library for the inclusion of speech and Braille in applications. A case study of the development of one such dual interface application is also presented.

1 Introduction

Graphical user interfaces were designed to increase the usability and to improve the functionality of applications [1]. However, a graphical interface is obviously a barrier for blind and visually-impaired people, although assistive devices (primarily speech synthesis, Braille displays, Braille and Qwerty keyboards, voice recognition and screen readers [2,3]) solve many of the problems they come up against. Nevertheless, the use of assistive technologies does not in itself guarantee that blind users will be able to access the application. To this end, the applications need to be built following special design rules [4,5].

Hence the adequacy of a graphical user interface for blind people involves some specific usability requirements for these users. This means that (i) the designer has to present the interface information in an appropriate structure that can be understood by blind users, (ii) none of the meaning of the interface content should be lost during the adaptation and (iii) the information supplied should be adapted for the assistive devices used by blind people. These requirements have an impact on HCI modeling. HCI modeling involves the creation of the task, domain, dialog, presentation, platform and user interface models [6], the design of which is therefore affected by this adaptation.

Based on the above, this paper describes a framework for developing graphical user interfaces for blind people based on usability requirements acquired from our experience in developing this kind of applications [7,8,9].

K. Miesenberger et al. (Eds.): ICCHP 2006, LNCS 4061, pp. 1031 – 1038, 2006.

The paper is organized as follows. Section 2 introduces related work. Section 3 presents usability requirements for blind people. Section 4 describes a framework for blind user interfacing. Section 5 presents an application developed using this framework and the paper ends with some general conclusions.

2 Related Work

The adaptation of an application designed for sighted users to the needs of blind users has the drawback of the original dialog design addressing the specific needs and abilities of sighted users. Also the syntax has to be restructured to meet the needs and abilities of blind people, and this is not always possible [10]. A clear example of this would be a musical score editing application. For a blind user to be able to use such an application, the blind user-adapted tasks and dialog process need to be defined.

A better solution is to create applications with special-purpose interfaces for blind users or with dual interfaces. Toolkits like HAWK [11], which provides a set of standard non-visual interaction objects and interaction techniques that have been especially designed to support high quality non-visual interaction, have been created to facilitate this process. Another tool aiding the development of dual interfaces suitable for both sighted people and blind users is the User Interface Management System (UIMS) [10]. HOMER [12] is an example of an UIMS that improves the development of dual user interfaces.

In any case, we have found, as pointed out by [13], that in order to design an interface suited for use by blind people it is essential to examine the fundamental accessibility issues for such users and define appropriate usability guidelines for them. Not only should these guidelines be formulated as general design principles or low-level and platform-specific recommendations, but they should also be based on experimental evidence [14].

3 Usability Requirements for Blind People

When specifying usability requirements for blind people, it is important to bear in mind that these requirements have to coexist with those of other user types in accordance with Design for All principles (equitable use, flexibility in use, simple and intuitive use, perceptible information, error tolerance, low physical effort, size and space for approach and use) [15] and standard HCI dialog principles (suitability for the task, self-descriptiveness, conformity with user expectations, suitability for learning, controllability, error tolerance, suitability for individualization) [16].

The usability requirements for blind and non-blind people differ to a large extent. Firstly, there is the fact that blind people cannot perform all tasks at the current state of technology. For instance, a blind person cannot as yet drive a car, which means that this task would not have to be adapted for this group of people. This question is referred to as *task adequacy*.

Secondly, blind people and non-blind people use different dimensional access schemes. For sighted users, it is a 2+1 scheme: the user interface objects are distributed in two-dimensional regions (the screen, a window, etc.) and their position typically provides additional semantics about each object. The “+1” component represents the transition from one 2D region to another (navigation), which is performed by user actions. The dimensional access scheme of blind people is 1+1. Devices for blind people (speech and Braille displays) are one-dimensional, which means that user interface objects are presented using a list structure and that the navigation is towards another list. This is what we have called the *dimensional trade-off*.

Thirdly, there is the requirement of maintaining a *behavior equivalence*. A non-blind person interacts using direct manipulation of screen objects, which is not possible for a blind user using assistive technologies. Thus, the interaction process should define how to make each interactive object accessible for the blind. Extra actions need to be added for all objects, including the messages that must be sent to the assistive technologies, the mechanism for accessing each part of the item, and any browse functions that can be performed.

Fourthly, it is not enough just to adapt each object for blind users. The information provided to blind users during task performance also has to be adapted, assuring that there will be no semantic loss during the process, because relevant information may be conveyed by means of images or the spatial distribution of objects located in a window. This is what we have termed *semantic loss avoidance*.

Finally, the assistive technologies used by blind people are not standardized. Speech devices differ both in the functionality they offer and the programming interfaces that have to be used, and the same applies to Braille displays. Applications must deal with this diversity of devices. We call this the *device independency requirement*.

4 A Framework for Blind User Interfacing

The above-mentioned requirements have to be dealt with when applying the modeling approach of user interface development [6], which is based on the task, domain, dialog, presentation, platform and user models.

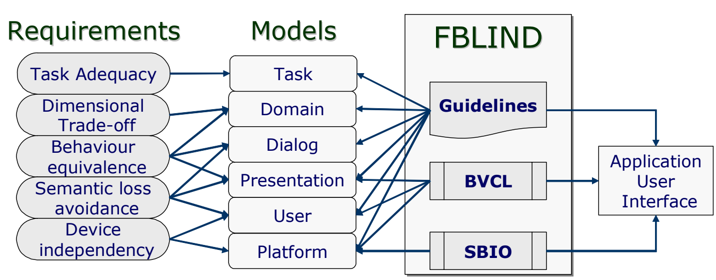

Fig. 1. Blind user requirements, HCI models and FBLIND

Based on the experience acquired over the years, we have developed a framework for the development of dual user interfaces for sighted and blind people: FBLIND (Framework for BLind user INterface Development). Fig. 1 shows the relationship between the blind-user requirements, the HCI models and our framework.

FBLIND has three main components: a set of user interface design guidelines, an interface development toolkit consisting of automatically adapted user interface objects (BVCL) and a programming library providing support for speech and Braille input and output (SBIO).

4.1 Blind User Interfacing Guidelines

These guidelines are based on the recommendations of the Spanish and international standards on software accessibility [4,5]. These recommendations, in whose development we are participating, along with other tips that we have picked up over the years have been incorporated into the proposed framework.

Our guidelines affect all the HCI models to a greater or lesser extent and contain useful information for all the stages of interface design and implementation. One very important aspect of these guidelines is that the application should be compatible with screen readers and other assistive technologies that blind people use to access graphical user interfaces. Below is an overview of the main issues for each of the models.

In the case of the *task model*, there is only one constraint: the tasks described should be checked for incompatibility with the capabilities of blind people using assistive technologies. This caters for the task adequacy problem.

The *domain model* is affected by the dimensional trade-off and the behavior equivalence requirements. To solve the dimensional trade-off, the sequence of windows and their content need to be defined hierarchically as a series of trees linked by transitions between windows to be compatible with 1+1 navigation. As regards behavior equivalence, the designer should define the information about what happens when transitions take place between tree levels or trees to be presented to the blind user as speech or Braille.

For the *dialog model*, the actions to be taken are determined by the behavior equivalence and semantic loss avoidance requirements. To solve the behavior equivalence problem, the dialog model should take into account that the main input device is the keyboard (or equivalent), and it should define each user interface object's interaction process and the access mechanisms to these objects' properties. With respect to semantic loss avoidance, a number of additional interaction commands should be incorporated to improve user operations, such as informing users about where they are in the application, repeating the last message, spelling the last message, etc.

The *presentation model* is subject to the behavior equivalence and the semantic loss avoidance requirements. With respect to the behavior equivalence, the speech and Braille presentation of each of the application's standard and special-purpose user interface objects should be defined. In terms of semantic loss avoidance, several detail levels will first have to be established for this information so that the interface is suited for both novice and experienced users. Additionally, special attention should be paid to non-standard objects and objects with a high graphical content to assure that no semantic information is lost.

The *platform model* is affected by the device independency requirement. Dual interface applications should provide speech output, Braille output or, preferably, both types of output. The platform should provide services for this functionality, and, in order to deal with device independency, it should provide standardized API for speech and Braille devices. The platform model, then, should contain information about the capabilities of those APIs.

Finally, the *user model* has to deal with the semantic loss avoidance and the device independency requirements. In particular, the user model should represent configuration parameters for the functionality required to solve both these problems, such as speech parameters (speed, tone, volume…), Braille parameters (cursor type, Braille code type…) and help level (for novice, intermediate or advanced user).

4.2 BVCL

The usability requirements for blind people applicable to the presentation, platform and user models have been built into a toolkit, called BVCL (Blind user adapted VCL), which is an extension of the Borland C++ Builder development environment and has been developed as a result of three Master's degree dissertations.

BVCL incorporates user interface objects with automated presentation in speech and Braille (presentation model), which saves development time. It also eases the programming of additional speech and Braille messages (platform model).

Finally, the specific messages generated in speech and Braille for each user interface object can be easily configured through BVCL, and it also incorporates layout managers to automate the layout and sizing of the user interface objects according to user preferences on font sizes, colors, etc. (user model).

4.3 SBIO

SBIO (Speech and Braille Input / Output) consists of a series of programming components that improve speech and Braille input and output. It is a package for Microsoft Windows, as this is one of the platforms most commonly used by blind people. It was developed jointly by Microsoft, the ONCE (the Spanish National Organization for the Blind), and the UPM (Madrid Technical University).

SBIO contains four components: the Microsoft speech API, a Braille API, a set of ActiveX components that offer a simplified and unified speech and Braille interface (SBIO-Core), and a dynamic link library (SBIO-Lib) that facilitates the use of IOSB.

5 An Environment for Bilingual Dictionaries with a Dual Interface

DABIN is an environment that includes bilingual dictionaries with a generic interface (Fig. 2) combining visual and blind interfaces (using speech synthesis and Braille). Its basic function is to search for words in one language and get their translation in another language from a range of bilingual dictionaries. Users can choose the source and target languages of the translation, as well as the user interface language.

All the tasks in *task model* were checked to ascertain whether they were adequate for blind users and accessible by means of the keyboard, speech synthesis and Braille display, and that they did not require hand-eye coordination or simultaneous control of different visual items.

In the *domain model*, DABIN has a hierarchical structure based on linked trees that can be easily explored in both the 2+1 and 1+1 access schemes. It should be noted that the tree structure for each software window is simple and quite shallow. Because blind users must necessarily navigate by levels within the tree and cannot "take a look" to find out where to go, the software windows are designed to assure that their navigation tree is not overly complicated.

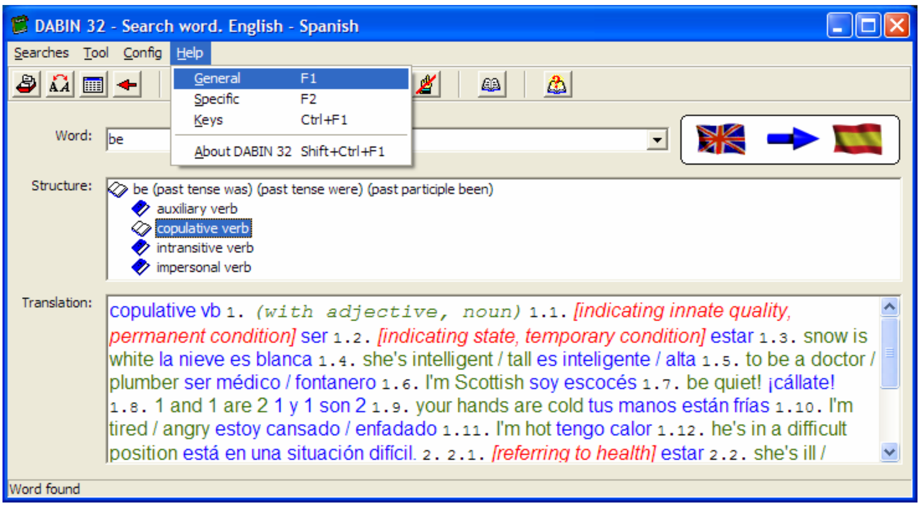

Fig. 2. DABIN's graphical user interface

In the *dialog model*, mechanisms were enabled to allow blind users to do all the tasks using key combinations or system menus. The interaction procedure for each user interface object was also defined for adaptation to blind people's needs. In addition to the standard user interface objects included in BVCL, a new complex object was needed: a translation browser, used to display the translation (bottom box of Fig. 2). We defined an interaction procedure for this component to make the reading of a translation a natural activity for blind users. This procedure is based on enabling several ways of getting the information using the keyboard only, allowing users to move forwards and backwards over the translation, select text and start a search for the word which the cursor is over. Behavior compatible with standard edit boxes was applied in the translation browser design.

Commands to access additional semantic information, such as user location, repetition of the last message, spelling or context-sensitive help, have been added to the system interaction techniques to prevent semantic information loss.

To build a dual interface the *presentation model* considered the visual appearance (following the Windows standards) and the non-visual appearance (focusing on sound –speech synthesis– and haptic –Braille– presentation). The sound and haptic interfaces for each user interface object were defined using BVCL and SBIO. The output in response to the different user actions on the translation browser was also defined in detail.

Finally, we identified three user types in the *user model*: sighted users (no special-purpose adaptation required), visually impaired users (the colors and the sizes of the texts can be configured), and blind users (the voice parameters can be configured for each of the languages as can the general appearance of the Braille).

To check that the information provided by the system was correct, several users of the three identified types tested this interface. As a result, the model was refined until the assistive technologies provided equivalent semantic information to what was displayed on screen.

6 Conclusions

We believe dual interfaces development to be the best option for efficiently developing and building user interfaces that satisfy the needs of both sighted and blind users. To this end, we have defined specific requirements that take into account all the problems that blind users face and that coexist with the standard dialog principles and Design for All guidelines. We have built these requirements into the HCI models. We have also presented a framework (FBLIND) to improve the development of these user interfaces. This framework includes a set of *guidelines*, which help the developer to focus on relevant issues during the construction of the six HCI models; a *toolkit* for auto-adapted user interface objects, which greatly reduces the implementation costs of dual user interfaces, and an *API* for the use of assistive devices, which provides an easy way to implement speech and Braille output. The use of this framework has been illustrated by means of DABIN, a software environment for bilingual dictionaries for use by blind and sighted users.

More tool support should be provided for tasks that are not yet covered by the toolkit and the API. A CASE tool covering the whole user interface development life cycle, providing support for manual tasks would be the ideal thing.

References

1. Cox, K., Walker, D.: User interface design. Prentice Hall (1993)
2. Baum Engineering SRL: Welcome to Gnopernicus!, available at URL: `http://www.baum.ro/gnopernicus.html` (consulted January 10, 2006)
3. Freedom Scientific BLV Group: JAWS(r) for Windows(r) Quick Start Guide, 11800 31st Court North, St. Petersburg, Florida 33716-1805 USA (2005)
4. ISO: Ergonomics of human-system interaction – Guidance on accessibility for human-computer interfaces, Technical Specification ISO TS 16071 (2003)
5. AENOR: Computer applications for people with disabilities. Computer accessibility requirements. Software (Spanish title: Aplicaciones informáticas para personas con discapacidad. Requisitos de Accesibilidad al Ordenador. Software), Spanish Standard UNE 139802 (2003)

6. Szekely, P., Luo, P., Neches, R.: Beyond Interface Builders: Model-Based Interface Tools. In: Proc. of InterCHI'93, ACM Press, New York (1993) 383-390
7. Alonso, F., Frutos, S., Montes, C.: Interfacing for the Blind. The DILE Model. In: Proc. CSUN´s Twelfth Annual International Conference on Technology and Persons with Disabilities, Los Angeles, March, (1997)
8. Alonso, F., Frutos, S., Fuertes, J. L., Martínez, L. A., Montes, C., Sánchez, A.: Un Diccionario Automático Bilingüe para Invidentes (DABIN). Integración 25 (1997) 14-31
9. Alonso, F., Barreiro, J. M., Caraça-Valente, J. P., López, G., Montes, C., Navajo, R. J.: An Electronic Newspaper for the Blind. In: Proc. XV IFIP World Computer Congress, Computers and Assistive Technology ICCHP '98, Vienna (Austria) and Budapest (Hungary) (1998)
10. Savidis, A., Stephanidis, C.: Developing dual user interfaces for integrating blind and sighted users: the HOMER UIMS. In: Proceedings of the SIGCHI conference on Human factors in computing systems, Denver, Colorado, United States (1995) 106-113
11. Stephanidis, C., Savidis, A.: Interface Development Toolkits for Non-visual and Switch-based Interaction. ERCIM News, 46 (2001), available at URL: `http://www.ercim.org/publication/Ercim_News/enw46/stephanidis2.html`.
12. Savidis, A., Stephanidis, C.: The HOMER UIMS for dual user interface development: Fusing visual and non-visual interactions, Interacting with Computers 11(1998) 173-209
13. Morley, S.: Design and Evaluation of Non-Visual Information Systems for Blind Users, PhD, University of Hertfordshire, UK (1999)
14. Casali, S. P.: A physical skills-based strategy for choosing an appropriate interface method. In: Edwards, A. D. N. (ed.): Extra-ordinary Human Computer Interaction: Interfaces for people with disabilities, Cambridge University Press (1995) 315-342
15. Connell, B. R., Jones, M., Mace, R., Mueller, J., Mullick, A., Ostroff, E., Sanford, J., Steinfeld, E., Story, M., Vanderheiden, G.: The Principles Of Universal Design, NC State University, The Center for Universal Design (1997)
16. ISO: Standard 9241-10:1996, Ergonomic requirements for office work with visual display terminals (VDTs) – Part 10: Dialog principles, International Organization for Standardization (1996)