Comparative Analysis of the Accessibility of Desktop Operating Systems

Ángel Lucas González[1], Gonzalo Mariscal[2], Loïc Martínez[1], and Carlos Ruiz[2]

[1] Facultad de Informática, Universidad Politécnica de Madrid
Campus de Montegancedo, s/n. Boadilla del Monte - 28660, Madrid – Spain
{agonzalez,loic}@fi.upm.es
[2] SETIAM – CETTICO, Fundación General UPM.
Campus de Montegancedo, s/n. Boadilla del Monte - 28660, Madrid – Spain
{gmariscal,cruiz}@cettico.fi.upm.es

Abstract. This paper presents the results of ongoing research on methods for evaluating the accessibility conformance level of software and especially operating systems. Our approach is based on recommendations from software accessibility standards, and defines techniques for evaluating each of those recommendations. The proposed method has been applied to evaluate the accessibility features of one closed-source and one open-source desktop operating system, Microsoft Windows XP and the Ubuntu Linux distribution, respectively. Specifically, the functionality we have evaluated was task management and file system management. From the point of view of the evaluation process, we conclude that more work is needed on the development of support tools and techniques. And from the point of view of the specific comparison, we conclude that, taking into account the analysed functionality, the current version of the Ubuntu Linux distribution is slightly more accessible than the current Windows release, though neither of the systems fully conform to the accessibility standards.

1 Introduction

There is an increasing demand for accessible software, especially motivated by recent public policies on the public procurement of accessible products and services in the field of information and communication technologies (ICT). Some examples are Section 508 of the Rehabilitation Act in the United States [1] and the European Commission's Mandate M376 on standards supporting accessibility requirements for public procurement [2].

In this context it is extremely important to be able to determine the degree of accessibility of software products, including operating systems. There exist international and national standards defining the requirements and recommendations of accessible software for this purpose, such as ISO TS 16071:2003 [3] and UNE 139802:2003 [4]. But the evaluation process, techniques and tools required to assess the accessibility conformance level are not as well defined as they are in the web context [5].

Additionally, because of the current trend within public administrations of switching from closed-source to open-source software, it is important to know whether this change can be safely made from the point of view of accessibility features.

C. Stephanidis (Ed.): Universal Access in HCI, Part I, HCII 2007, LNCS 4554, pp. 676–685, 2007.

This paper presents work in this direction. First we will present a method for performing assessments of the operating system accessibility conformance level based on the contents of the above two standards. Then we will present the initial results of the application of this method to compare the accessibility level of two desktop operating systems: Microsoft Windows XP SP2 [6] and Ubuntu Linux 6.10 [7].

2 Objectives

This paper is part of long-term research dealing with software accessibility for people with disabilities and more specifically with current desktop operating systems user interface accessibility.

The main goal of this paper is to propose a preliminary version of a method for evaluating the accessibility of operating systems based on the level of conformance to current software accessibility standards.

In the first application of the proposed method, we are particularly interested in finding out what difference (if any) there is between the accessibility of open-source versus proprietary systems. The proposed method has been applied to perform a comparative analysis of the accessibility level of relevant representatives of these two families of desktop operating systems. On the one hand, we have evaluated the accessibility of Microsoft Windows XP Service Pack 2. On the other hand, we have evaluated the accessibility of the Linux-based operating system Ubuntu 6.10 that ships with the GNOME 2.16 desktop environment. In the past Windows developers have put more effort into accessibility than the Linux community. This situation is changing rapidly and the open source community is now strongly committed to accessibility [8].

The method divides accessibility features into several functionality groups. In this research, we have applied part of the whole proposed method. Specifically we have focused on two functionality groups: task management and file system management. For each functionality group the method then defines a set of user tasks that have to be performed to evaluate the accessibility conformance level. All the accessibility features evaluated in each of these functions will be explained in section 4.

3 Significance

The results of this research are highly significant in the current context of policies concerning the inclusion of people with disabilities in the Information Society. One prominent policy is to promote accessibility in the public procurement process. Some examples follow below.

In the USA, Section 508 of the Rehabilitation Act requires the US government to purchase accessible electronic and information technology [1]. The related minimum accessibility requirements have been defined by the Access Board [9].

In Canada there exists an accessible public procurement toolkit [10], which is based on the 508 requirements with some additions. This toolkit enables public procurers to easily generate the needed procurement requirements for accessible products.

In the European Union, the European Commission has produced a Mandate on Public Procurement of accessible ICT products and services (M376) [2], and European Standards bodies are now working on the standardization of accessibility requirements

for public procurement. This implies that Europe also intends to apply public procurement as a tool for the promotion of accessible products and services.

Another significant current trend is to promote open source systems in public administrations all around the world. But the administrations willing to adopt open source systems should not overlook accessibility requirements. These administrations should guarantee that the switch to open source software does not discriminate against their employees and the general public by lowering accessibility levels.

In this situation it is very important to have a rigorous method to evaluate how accessible operating systems are and to help public administrations in the process of defining accessibility requirements for public procurement. Unfortunately, there is no method based on official standards for evaluating accessibility on a single operating system, nor has any comparative analysis of the accessibility of several operating systems been published to date.

Additionally, note that most of the accessibility requirements and recommendations for software cannot be automatically evaluated and they require the participation of an expert human evaluator. This has already being demonstrated in the specific case of web content [11] but also holds true for other types of software, such as operating systems.

In this paper, we propose a preliminary version of a method for evaluating software accessibility based on international standards and with the participation of human evaluators. Then we apply this method to run a comparative analysis of the accessibility of two representative desktop operating systems.

4 Method

We have developed a method to perform a complete study of OS accessibility. This method is composed of four steps: checklist definition, definition of checkpoint evaluation methods, selection of system functionalities and evaluation.

4.1 Checklist Definition

We have created a checklist of issues to be checked to assure accessibility for people with disabilities. We have followed two main guidelines: ISO/TS 16071[2] and UNE 139802 [3] (Spanish standard which covers some issues not covered by ISO). We have chosen these standards instead of the upcoming ISO 9241-171 [12] for two reasons. Firstly, the new ISO standard is still not stable enough to be referenced as it was in the "Draft International Standard" phase at the time of writing this paper. Second, the combined content of ISO TS 16071 and UNE 139802 covers most of the recommendations of the future standard.

The chosen standards define different criteria based on accessibility impact (core, primary and secondary for ISO/TS 16071, and priority 1, 2, and 3 for UNE) and implementation responsibility (OS and/or application for both) to guide developers through the design of systems and interfaces that are as accessible as possible. In our current work we have chosen the core recommendations from ISO/TS 16071 and then we added the priority 1 requirements from UNE 139802 that were not covered by ISO/TS 16071. This generated a checklist of 39 items, which is presented in Table 1. We used the ISO DIS 9241-171 wording for the UNE checkpoints when applicable.

Table 1. Checklist of accessibility requirements

ID	Short Description
General Guidelines	
[ISO] (7.2.1.1)	Enable user input/output choice
[ISO] (7.2.2)	Enable user to perform the task effectively with any single input device
[ISO] (7.2.4)	Enable user setting of timed responses
[ISO] (7.2.10)	Avoid seizure-inducing blink rates
[UNE] (4.1.1)	Minimise the number of steps required to perform any task
[UNE] (4.1.4)	Provide accessibility services
[UNE] (4.1.5)	Accept the installation of keyboard and/or pointing device emulators
[UNE] (4.1.6)	Be compatible with speech recognition systems
[UNE] (4.10.1)	Provide accessible system start-up and restart
[UNE] (4.10.2)	Enable software-controlled media extraction
Assistive Technologies	
[ISO] (7.3.2)	Provide object labels
[ISO] (7.3.3)	Make event notification available to assistive technologies
[ISO] (7.3.4)	Make object attributes available to assistive technologies
[ISO] (7.3.5)	Present user notification in a relevant manner
Keyboard-input configuration	
[ISO] (7.4.2)	Enable sequential entry of multiple keystrokes
[ISO] (7.4.7)	Provide keyboard control of pointer functions
[ISO] (7.4.11)	Reserve accessibility key-mappings
[ISO] (7.4.13)	Separate keyboard navigation and activation
[UNE] (4.2.1)	Enable full use via keyboard
[UNE] (4.2.5)	Enable the locking of control keys
Software control of pointing devices	
[ISO] (7.5.2)	Enable the adjustment of the location of button functions
[ISO] (7.5.9)	Provide alternatives to chorded key presses
Display fonts	
[ISO] (7.6.1)	Enable font customization and legibility
Displays	
[ISO] (7.7.3)	Provide access to information displayed in “virtual” screen regions
[UNE] (4.4.5)	Provide non-animated alternatives to animations
Colour	
[ISO] (7.8.1)	Provide alternatives to the use of colour as the sole source of information.
[ISO] (7.8.6)	Provide alternatives to coding by hue
Audio Output	
[ISO] (7.9.5)	Allow users to choose visual indication of audio output
[UNE] (4.5.1)	Provide accessible alternatives to task relevant audio and video
[UNE] (4.5.3)	Provide speech output services
[UNE] (4.5.4)	Synchronise speech output of visual events
Errors and user notification	
[ISO] (7.10.1)	Allow task-relevant warning or error information to persist
[UNE] (4.6.1)	Provide understandable user notifications
On-line documentation and help	
[UNE] (4.9.1)	Provide understandable documentation

Table 1. (*Continued*)

Customization of user preferences	
[ISO] (7.12.3)	Enable cursor and pointer customization
Windows appearance and behaviour	
[ISO] (7.13.1)	Enable non-pointer navigation directly to windows
Keyboard input focus	
[ISO] (7.14.1)	Provide focus cursor
[ISO] (7.14.2)	Provide keyboard navigation
[ISO] (7.14.3)	Provide navigation to task-appropriate groups of controls

4.2 Checkpoint Evaluation Method

For each checkpoint in our checklist we have defined a method of evaluation. There are two main types of evaluations.

The most commonly used evaluation method is based on tasks performed by a human evaluator using the software. For each system functionality we have defined a set of user tasks (see below in 4.3), and the evaluator has to perform those tasks checking whether or not the software complies with the checkpoint.

In addition, there are some special checkpoints that are not evaluated by performing tasks, but by reading the software documentation (both user and developer documentation). One example of such a checkpoint is UNE 4.1.4 (provide accessibility services). Table 2 gives an example of the method of evaluation followed.

Table 2. Example of the method of evaluation

Checkpoint	Method of evaluation
[ISO] (7.14.2)	Verify whether the user can navigate through the keyboard across user interface elements during any task.
[UNE] (4.1.4)	Verify whether documentation for developers defines an accessibility API.

The result of the evaluation for each checkpoint is one of five possible values:

- *Pass*: the OS definitely meets the requirements for the checkpoint.
- *Fail*: the OS definitely does not meet the requirements for the checkpoint.
- *Partial*: the software almost fully complies with the checkpoint, but there is a minor issue to be solved.
- *Unknown*: the evaluator cannot decide whether or not the software complies with the checkpoint.
- *Not applicable*: the checkpoint is not applicable for the OS in the current context of use.

4.3 Selection of System Functionalities and User Tasks

To perform the accessibility comparison, we have restricted the OS functionality under observation: we have only evaluated the basic functionality of an OS viewed as

a tool that enables the user to manage his or her files and tasks, and we have not considered additional software like media players or web browsing. The functionality groups covered are:

- *Task management*. The operating system should allow users to perform a set of actions in order to manage tasks (running applications). The user tasks that are considered for this functionality group are: get a list of the running applications (with relevant information on each one), swap tasks, start a new task (launch a new program) and finalize a task (stop program execution). We have considered window management as part of this functionality.
- *File system management.* The operating system should allow users to manage the folders and files stored in the computer storage devices. In our study we have examined the following user tasks: navigate through the file system, create, copy, rename, move and delete files or folders and, finally, delete and move multiple files.

Administrative tasks like disk formatting, disk integrity checking, new hardware installation and so on were not taken into account but will be considered in the future.

4.4 Evaluation Process

We have performed a group-based cross evaluation to enhance the objectiveness of the results. We created two evaluator groups, both of them formed by users with knowledge on both operating systems, usability and accessibility:

- In the first stage, group 1 analysed Windows XP and group 2 analysed Ubuntu. All group members worked individually and did not have access to the results of the other members of their or the other group.
- In the second stage, each group analysed the other OS.
- Then we combined the scores for each checkpoint (using the rules described in Table 3), and we conducted a basic data analysis to decide which was the more accessible operating system overall and for each main functionality.

Table 3. Accessibility evaluation combination rules. i, j ∈ {1 .. number of evaluators} and $eval_i$ is the result of evaluator i. Note: N/A means "Not applicable".

Condition	Result
$\exists i, eval_i = fail$	fail
$\exists i, eval_i = partial \wedge \forall i \neq j\ eval_j \neq fail$	partial
$\exists i, eval_i = pass \wedge \forall i \neq j\ (eval_j \neq fail \wedge eval_i \neq partial)$	pass
$\exists i, eval_i = N/A \wedge \forall i \neq j\ (eval_j = N/A \wedge eval_i = unknown)$	not applicable
$\forall i, eval_i = unknown$	unknown

5 Results

Table 4 is a summary of the results of the evaluation. Details on the results for each operating system are shown in Table 5 (for task management) and Table 6 (for file management). Tables 5 and 6 show the final values after applying the combination

rules and only contain values other than "pass" for both operating systems. Therefore, the result of the evaluation of any checkpoint not appearing in either Table 5 or 6 is positive. Some requirements in the tables are labeled "Not Applicable". This means that they don't make sense in that specific context.

Table 4. Summary of the evaluation results

Result	Windows XP	Ubuntu Linux
Fail	8	6
Partial	2	0
Pass	61	64
Not applicable	7	8
Unknown	0	0
TOTAL	78	78

The first and most important result is that neither of the two evaluated operating systems fully complies with the analysed checkpoints.

Table 5. Task management evaluation results

Requirement Identifier	Results	
	Windows XP	Ubuntu Linux
General Guidelines		
[ISO] (7.2.4)	***Fail***	Not applicable
[UNE] (4.10.1)	***Fail***	***Fail***
[UNE] (4.10.2)	Not applicable	Pass
Display Fonts		
[ISO] (7.6.1)	***Fail***	Pass
Displays		
[UNE] (4.4.5)	Not applicable	Not applicable
Audio Output		
[UNE] (4.5.1)	Not applicable	Not applicable
[UNE] (4.5.4)	Not applicable	Not applicable
Errors and User Notification		
[UNE] (4.6.1)	***Fail***	***Fail***
On-line Documentation and Help		
[UNE] (4.9.1)	Pass	***Fail***

Analysing the collected data and comparing both operating systems as a whole, we can say that, for the task and file management functionality groups, the current version of Ubuntu Linux with GNOME is slightly more accessible than the current version of Windows XP. Details on the failed checkpoints are given below.

Table 6. File system management evaluation results

Requirement Identifier	Results	
	Windows XP	**Ubuntu Linux**
General Guidelines		
[ISO] (7.2.4)	Not applicable	Not applicable
[UNE] (4.10.1)	***Fail***	***Fail***
Assistive Technologies		
[ISO] (7.3.5)	***Fail***	***Fail***
Displays		
[UNE] (4.4.5)	***Fail***	Not applicable
Audio Output		
[UNE] (4.5.1)	Not applicable	Not applicable
[UNE] (4.5.4)	Not applicable	Not applicable
Errors and User Notification		
[ISO] (7.10.1)	***Fail***	Pass
[UNE] (4.6.1)	Partial	***Fail***
On-line Documentation and Help		
[UNE] (4.9.1)	Partial	Pass

5.1 Task Management

- *[ISO] (7.2.4.1)*: Windows XP fails because when a user wants to kill an incomplete process or shutdown the computer and there are documents to be saved, it shows a dialog box with a time-out asking if the user wants to wait for the program to end correctly or finish immediately. In this dialog there is no way for the user to set the time limit. Linux has no timed dialog, so this requirement is not applicable.
- *[UNE] (4.10.1)*: both systems fail because start-up and restart are not fully accessible. If anything unusual happened during the last operating system execution, the start-up and restart processes will perhaps need to interact with the user, but assistive technologies are not activated until later. This checkpoint is common for both functionality groups (task management and file system management).
- *[ISO] (7.6.1)*: Windows XP fails because the font of some texts in the task manager cannot be modified by the user.
- *[UNE] (4.6.1)*: Windows XP fails because it displays a message like "Error 0x53: ask your administrator" for some application errors and some cases when the operating system returns a fatal error. Also Ubuntu Linux displays messages like this.
- *[UNE] (4.9.1):* Ubuntu Linux fails because some help documents contain sections written in different languages (i.e. a mixture of English and Spanish).

5.2 File System Management

- *[UNE] (4.10.1)*: Common checkpoint, see "5.1. Task management" for details.
- *[ISO] (7.3.5)*: Both operating systems fail. In Windows some informative (non-error) messages are shown in a "Dialog Box" and others are shown inside a "Bubble" (i.e.

renaming a file with a non-allowed character). In Ubuntu Linux, a lot of informative messages do not have a title.

- *[UNE] (4.4.5)*: There is an animation in the progress dialog that cannot be deactivated when copying a large set of files in Windows.
- *[ISO] (7.10.1)*: There is one case in Windows that does not comply with the requirement. When the user renames a file or directory and types a non-allowed character (i.e. ":", "/","*"), the system shows the message 'A file cannot contain any of the following characters: \ / : * ? " < > |'. But if the user switches to another window or task, the message will no longer be there when he or she goes back to the original window or task.
- *[UNE] (4.6.1)*: Windows XP fails because it displays a message like "Error 0x53: ask your administrator" for some application errors and some cases when the operating system returns a fatal error. Also Ubuntu Linux displays messages like this.

6 Conclusions and Future Work

Today, software accessibility is a relevant issue concerning both the public administrations and the private sector. In particular, public administrations all around the world are starting to require accessibility compliance for the public procurement of products and services.

For this to be possible there is a need for reliable methods to assess the accessibility conformance level of software products. In this paper we have presented a preliminary version of such a method and we have applied it to two desktop operating systems —Microsoft Windows XP SP2 and Ubuntu Linux 6.10— to compare their accessibility level.

We observed some variability in the results obtained by different evaluators when applying the method. This variability cannot be explained either by their previous knowledge of the two systems or by the order in which they performed the assessment. This leads us to suspect that we need to refine and clarify the evaluation method for the checkpoints so that the results are more objective and reliable. This is the main goal of our future research. In addition we will need to develop tools to support the evaluation process.

Concerning the comparison between the analysed systems, neither fully complies with the evaluated checkpoints, but they are nearly there. Ubuntu is slightly more accessible but there are no big differences. And the results may have been different if we had been able to evaluate the latest version of Windows (Vista) which appeared recently. This would have been a fairer comparison in view of the fact that Windows XP is 5 years old and Ubuntu 6.10 was not released until last November.

Finally, as regards the debate between open-source and proprietary software, given the small differences that we identified, we do not expect accessibility to be a key factor at present or in the near future.

References

1. United States Department of Justice. 1998 Amendment to Section 508 of the Rehabilitation Act. Section 508 of the Rehabilitation Act (29 U.S.C. 794d), as amended by the Workforce Investment Act of 1998 (P.L. 105-220), (August 7, 1998)

2. European Commission. Enterprise and Industry Directorate-General. Standardisation mandate to CEN, CENELEC and ETSI in support of European accessibility requirements for public procurement of products and services in the ICT domain, M 376 – EN 2005-12-07 (2005)
3. International Standards Organisation. ISO/TS 16071:2003. Ergonomics of human-system interaction – Guidance on accessibility for human-computer interfaces. Technical Specification (2003)
4. Spanish Association of Standardisation and Certification (AENOR). UNE 139802:2003. Computer applications for people with disabilities. Computer accessibility requirements. Software (in Spanish) (2003)
5. World Wide Web Consortium. Evaluating Web Sites for Accessibility: Overview, (2007) Available at http://www.w3.org/WAI/eval/Overview.html
6. Microsoft Corporation. Windows XP Home Page. (2007) Available at: http://www.microsoft.com/windows/products/windowsxp/default.mspx
7. Ubuntu. Welcome - Ubuntu: Linux for human beings. (2007) Available at: http://www.ubuntu.com
8. GNOME. The GNOME accessibility project (2007) Available at: http://developer.gnome.org/projects/gap/
9. Access Board. Electronic and Information Technology Accessibility Standards. Final Rule. Federal Register / vol. 65, No. 246 / Thursday, December 21, Rules and Regulations. Part II Architectural and Transportation Barriers Compliance Board 36 CFR Part 1194 (2000)
10. Government of Canada. The Accessible Procurement Toolkit (2007) Available at http://www.apt.gc.ca/
11. Benavidez, C., Fuertes, J.L., Gutiérrez, E., Martínez, L.: Semi-automatic Evaluation of Web Accessibility with HERA 2.0. In: Miesenberger, K., Klaus, J., Zagler, W., Karshmer, A.I. (eds.) ICCHP 2006. LNCS, vol. 4061, pp. 199–206. Springer, Heidelberg (2006)
12. International Standards Organisation. ISO DIS 9241-171. Ergonomics of human-system interaction – Part 171: Guidance on software accessibility. Draft International Standard (2006)