An Approximation Simulator for Designing and Developing Accessible Java Swing Applications

Theofanis Oikonomou[1], Konstantinos Votis[1,2], Dimitrios Tzovaras[1] and Peter Korn[3]

[1] Informatics and Telematics Institute, Centre for Research and Technology Hellas,
6th km Charilaou-Thermi Road, P.O. Box 60361, Thessaloniki, GR-57001 Greece
{thoikon, kvotis, tzovaras}@iti.gr
[2] Pattern Recognition Laboratory, Computer Engineering and Informatics, University of Patras, Rio Patras
botis@ceid.upatras.gr
[3] Sun Microsystems, Inc.,
17 Network Circle, MPK17-101, Menlo Park, CA 94025, USA
Peter.Korn@Sun.COM

Abstract. Failure to determine the requirements for accessibility support in rich Internet applications (RIA), and to verify that such support meets the needs of people with disabilities will result in the continued poor or non-existent levels of access for these applications. Developers and designers face a real challenge when implementing software that has to cope with a lot of interaction situations, as well as specific directives for ensuring an accessible interaction. In this paper we propose an approximation simulator for Java™ Swing applications. The proposed tool will assist developers and designers to explore user-centred design and incorporate important accessibility features for their Java™ Swing software implementations.

Keywords: Human Computer Interaction, Web Accessibility, Simulation, software design, User-centered design.

1 Introduction

The use of Information and Communication Technology (ICT) is increasing and proving to be of greater importance in people's day-to-day lives. However, this increase in use and importance of ICT raises concerns as to whether these technologies are fully accessible for all people, especially to disabled and elderly people.

Access by disabled and elderly people to technology is a major issue in allowing their integration in society. The growing global proportion of disabled and elderly people signifies that the urgency of this issue will not decrease in the years to come.

Accessible ICT systems can open up new possibilities and opportunities for people with disabilities, because they build in facilities that enable such people to use the systems independently. ICT systems which do not build in these accessibility factors will cause enormous frustration because they cannot be simply used by disabled people on their own, as independent human beings.

Disabled people will often use assistive technology, such as a screen reader or a modified mouse. These add-ons help but are rarely a complete solution. Systems work best when they are specifically designed for able and disabled people, using and positively supporting accessible technologies.

Most ICT systems are not fully accessible. This is not because the designers/developers have discriminated against people with disabilities on purpose, but because accessibility does not happen on its own. Many developers and designers are not fully equipped with evidence and knowledge related to the accessibility of their products or services [1]. Consequently, even the newest developments are not adequately accessible, missing the opportunity of tackling this issue at the development stage, when costs are compatible and solutions by design can be found, rather than making aftermarket adaptations. Moreover, they usually do not know how to specify accessibility requirements or how to design and test their product’s compliance against the ones defined.

Existing development tools and packaged solutions (e.g., several CAD tools or simulation environments) give little out-of-the-box assistance in most cases or, at worst, make it impossible to design and develop accessible ICT Java™ solutions for impaired users. It is important that the design and development of accessible ICT solutions

can be supported in an automated fashion, as much as possible. Thus, developers and designers need tools that provide guidance to them in how to apply the accessibility principles.

This paper presents an approximation simulation standalone tool to achieve accessibility design and development of Java™ Swing applications.

2 Related Work

It is usually difficult for designers and developers to understand the problems users with disabilities face when accessing their software implementations that are not designed with their needs in mind. Thus, in order to have a better view of accessibility needs, in some cases developers use relevant simulation tools that can provide a way to experience an application using simulated disabilities. It is obvious that simulation tools cannot simulate all kinds of disabilities and cannot provide the exact impact they have, but they certainly provide information and help designers make user interface content more accessible. These tools can enable, encourage, and assist designers/developers in the creation of accessible applications.

Much work has already been done in exploring the accessibility for Web applications. There are many tools that can help Web designers and developers to create accessible applications. This is accomplished by providing to the designers a report with the problems found in the page, by simulating the page against a chosen disability or by a combination of these two approaches.

The aDesigner [2] is a disability simulator that helps Web designers to ensure that their pages are accessible and usable by the visually impaired and especially by low-vision and blind people. Color Doctor [3] is a simulator that can check colour accessibility and shows the display content through four conversion filters: Grayscale, Protanopia, Deuteranopia and Tritanopia. The Visual Impairment Simulator (VIS) for Microsoft Windows [4] is another tool that simulates everything displayed on the desktop and supports Cataract, Color Blindness, Diabetic Retinopathy, Glaucoma, Hyperopia, Macular Degeneration, Magnifier and Retinitis Pigmetosa. The Vischeck [5] tool was created, in order to help web developers check their work for colour blind visibility. WebAIM Low Vision Simulator [6] provides users with the

opportunity to experience a web page as a user with visual disabilities such as Macular Degeneration, Cataract or Glaucoma. Finally, Cambridge University has developed a vision impairment simulator which is included to the inclusive design toolkit [7].

All these simulators are considered to be a good guide to the accessibility enhancement of Web sites and Microsoft applications. The main drawback is that they don't offer simulation capabilities for other applications than Web such as Java™ Swing applications.

In order to test a Java™ application for accessibility an accessibility test tool should be used. Currently, Sun Microsystems Java Accessibility Utilities [8] contain a set of tools which can be used to help in testing the accessibility of an application. For example, the Ferret tool will display each component's accessibility information (e.g. AccessibleName and AccessibleDescription). A developer can use the tools to ensure that relevant information is available on components. Furthermore, the Java Accessibility Helper [9] is a graphical tool that helps software developers and test engineers examine Java-based applications for accessibility issues. For example, the Helper verifies that all Graphical User Interface (GUI) components can be reached using only the keyboard and that their names, roles, states, and more can be obtained by assistive technologies. Examples of assistive technologies include screen readers, screen magnifiers, on-screen keyboards and voice control. The Helper can be used to evaluate any AWT- or Swing-based application and creates accessibility reports based on test sets. Finally, a tool for simulating various vision impairments in developing Java™ Swing applications was presented in [10]. This tool can be used as a part of the Netbeans Integrated Development Environment (IDE) [9] as well as a standalone application, aiding the designer/developer throughout the phases of the whole development process. This way accessibility barrier can be overcome and the overall quality of the developed applications improved.

3 An Approximation Simulator for Accessible Applications

The purpose of the approximation simulator is to assist developers and designers to better empathise with those who have certain capabilities and to help understand how this loss affects the ability to interact with

software applications and services. The simulator has been designed and developed as a standalone application, in order to present accessibility drawbacks within the design and development phase of Java™ Swing applications. Moreover, any developer/designer and/or evaluator can verify the accessibility status of any standalone Java™ Swing application without the necessity of accessing the source code of the application.

Thus the proposed simulator can serve as a specific approximation tool to answer relevant accessibility questions for developers and designers of Java™ Swing applications by receiving feedback through message boxes and/or useful forms. The tool supports several widespread visual and upper limp impairments and disabilities, such as colour blindness, loss of central and peripheral vision, blurred vision, extreme light sensitivity, night blindness and Parkinson's disease. Most of the supported impairments, especially central and peripheral vision loss, have a negative impact on computer use, since modern operating systems employ GUIs which require the use of eye-to-hand coordination to operate the mouse. In the following table a short description for the supported impairments is presented.

Table 1. Short description of the supported impairments.

Impairment	Short Description
Cataract	A clouding of the lens in the eye that affects vision
Macular Degeneration	A medical condition usually of older adults which results in a loss of vision in the centre of the visual field
Glaucoma	A group of diseases that can damage the eye's optic nerve and result in vision loss and blindness
Retinitis Pigmentosa	In the progression of symptoms for retinitis pigmentosa, night blindness generally precedes tunnel vision
Hyperopia	Hypermetropia, far-sightedness or long-sightedness causes inability to focus on objects at near distance (or at any distance in extreme cases)
Night Blindness	Nyctalopia (Greek for "night blindness") is a condition making it difficult or impossible to see in relatively low light
Extreme Light Sensitivity	Hemeralopia (Latin for "sun blindness") is the inability to see clearly in bright light and is exactly opposite of Nyctalopia
Protanopia	A severe type of colour vision deficiency that is a form of dichromatism in which red appears dark
Deyteranopia	A colour vision deficiency that moderately affects red-green hue discrimination
Tritanopia	An exceedingly rare colour vision disturbance in which there is a total absence of blue retinal receptors
Protanomaly	A mild colour vision defect which results in poor red-green hue discrimination

Impairment	Short Description
Deyteranomaly	Is by far the most common type of colour vision deficiency, mildly affecting red-green hue discrimination
Tritanomaly	A rare, hereditary colour vision deficiency affecting blue-yellow hue discrimination
Achromatopsia	Rod monochromacy is a rare, non-progressive inability to distinguish any colours
Achromatomaly	Blue cone monochromacy is a rare, total colour blindness that is accompanied by relatively normal vision, electoretinogram, and electrooculogram
Parkinson's Disease	A degenerative disorder of the central nervous system that often impairs the sufferer's motor skills, speech and other functions and is the most common cause of chronic progressive parkinsonism, a term which refers to the syndrome of tremor, rigidity, bradykinesia and postural instability.

The approximation simulator has been implemented in order to give the ability to designers/developers to explore, run and verify their software implementations. One of the main capabilities of the tool is the effective simulation of applications during their running process and when new windows, such as dialogs, choosers or frames, may appear due to the performed user interactions. Thus, each user can simulate any of the appeared windows of the application that he/she prefers. On startup the Java™ Swing GUI application is specified by the user. The only requirement is that it is bundled in a jar file. After that the approximation simulator runs the application and performs the simulation process based on preferable variables. An overview can be seen in Fig. 1, where the user can select any of the supported impairments through the impairment chooser panel of the approximation simulator.

The user can be navigated through its software implementations through the runtime panel in the left area of the tool where the user can interact with the actually running software application. The effects of the simulation process are depicted to the simulation area in the right panel of the simulator. Thus, any modification to the running application can be presented in parallel to the simulation panel.

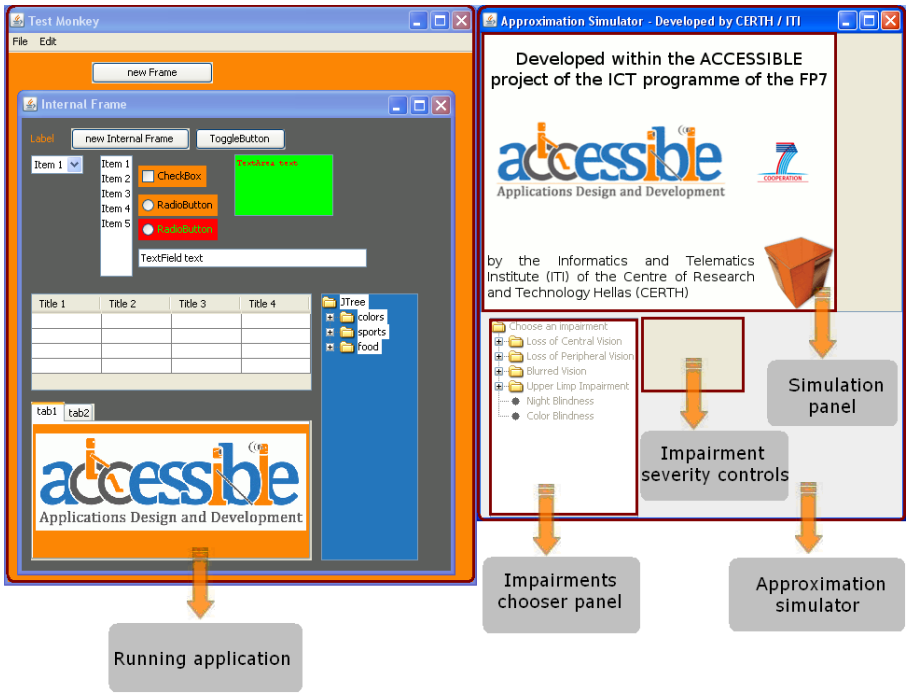

Fig. 1 Approximation simulator in initial state.

For each supported impairment, various factors regarding the specific impairment can be controlled from the severity controls panel. This way, different degrees of severity for that particular impairment can be simulated. Another interesting feature is that the approximation simulator inherits the Look and Feel (L&F) that was set to the application by the developer. Furthermore, the approximation simulator respects the preferences of the developer/designer regarding the display high contrast accessibility options and the mouse pointers. This means that the tool can adapt to the high contrast appearance scheme as well as the mouse pointer scheme set by the designer/developer.

4 An Approximation Simulator Execution Scenario

Consider the following case as an example. The user wants to simulate glaucoma. After selecting the impairment, appropriate controls show up. In this case the Windows L&F was set, the high contrast feature of

the display accessibility options was deactivated and the “Windows default” mouse pointer scheme was chosen.

As can be seen in Fig. 2 any potential patient that suffers from glaucoma may have some problems navigating through the application.

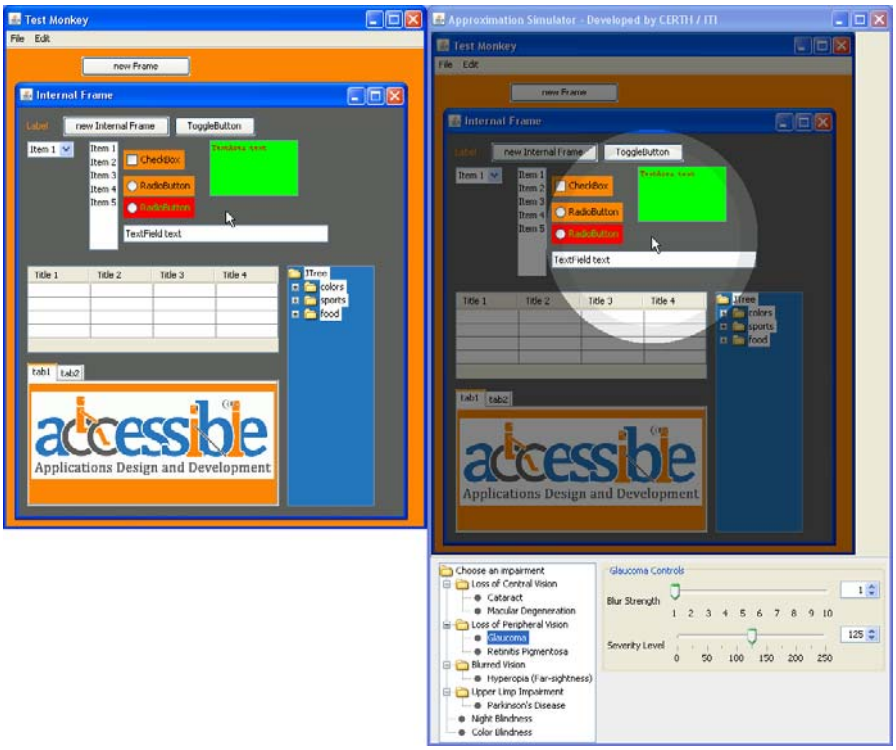

Fig. 2 Approximation simulator simulating glaucoma.

This can be better understood when increasing the severity level of the impairment as seen in Fig. 3. In this case the factors that affect glaucoma were altered. Additionally, the Windows L&F was set as well, the high contrast feature of the display accessibility options was activated and set to the “High Contrast White” value and the “Windows Standard (extra large)” mouse pointer scheme was chosen. This way the developer can better understand how a user with different degrees of a specific impairment can perceive the application.

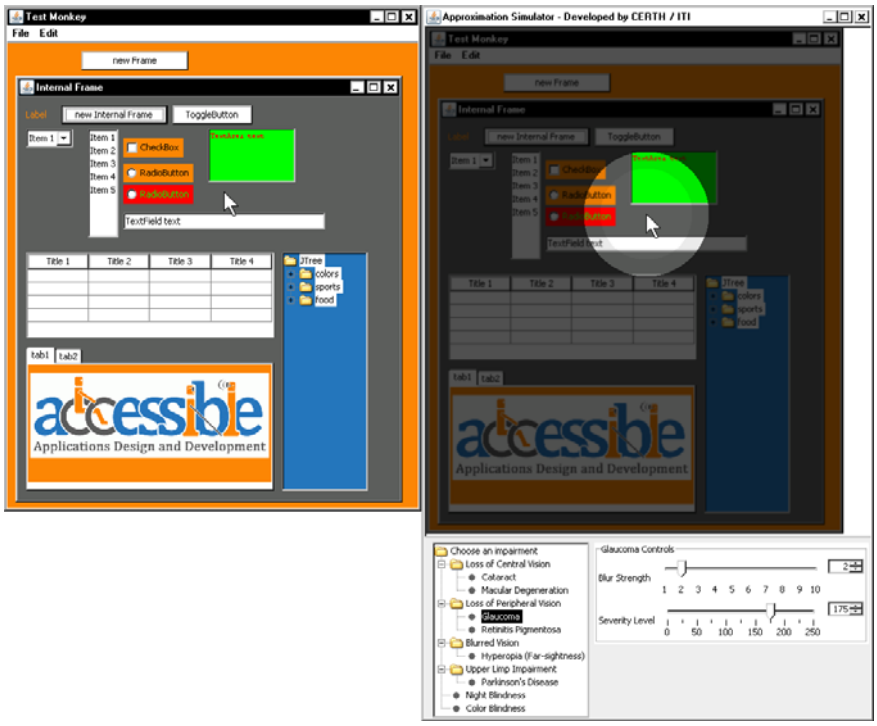

Fig. 3 Approximation simulator simulating a more sever case of glaucoma.

So far the designer/developer is able to receive appropriate visual feedback regarding the way a potential user with some disability would perceive the developed GUI application. Moving one step forward the approximation simulator gives useful information on any Java™ GUI component with possible accessibility problems. As an example, in the case of protanopia, as shown in following Fig. 4, several red rectangles are drawn on top of the simulated version of the running application. These rectangles mark the boundaries of the problematic GUI components found in the application. The problem is that they do not have the required background/foreground colour contrast ratio in order to be easily perceived by an impaired user.

The designer/developer can get more details about a specific GUI component by clicking inside the red rectangle drawn around that component. Then, an information window will appear (Fig. 5) containing some information regarding the GUI component and why there is a problem with it as well as recommendations on how to fix it.

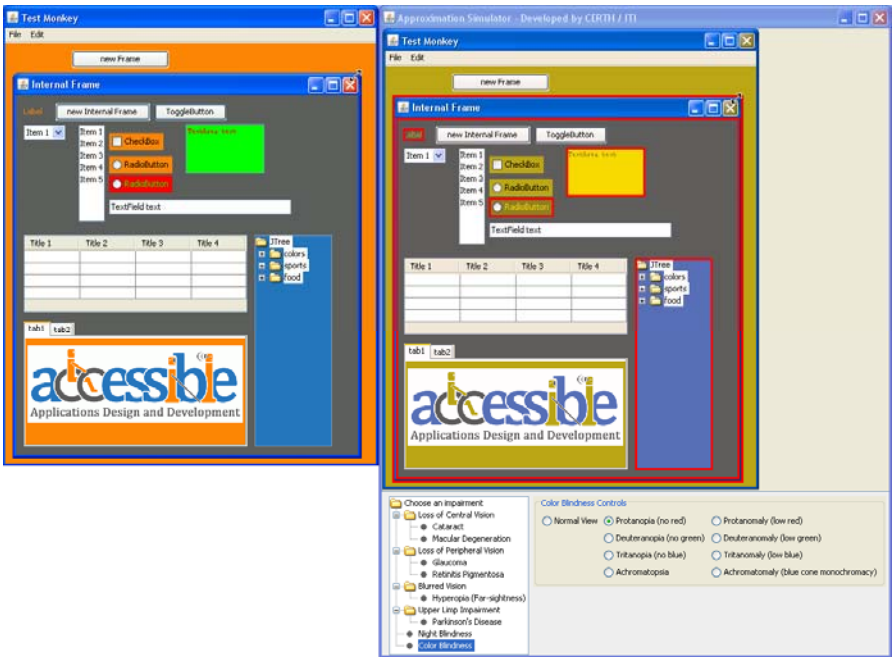

Fig. 4 Approximation simulator simulating protanopia.

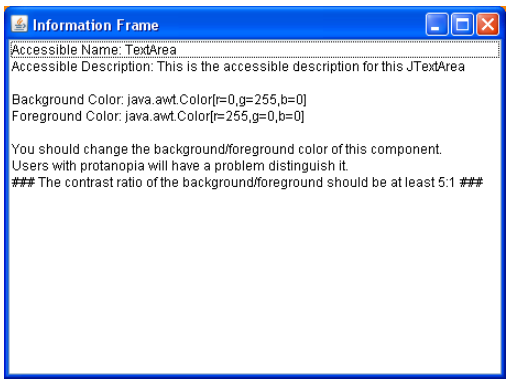

Fig. 5 More detailed information when a problematic issue has been detected.

Finally, it is important to mention that the usage of the approximation simulator does not portray exactly how it is like to have a single or a combination of vision impairments. However, fully-sighted users who spend some time using the simulator's functionalities, throughout their design and development processes, can

quickly acquire a sense of some of the design issues that should be taken into account in order to create accessible software applications.

5 Conclusion and Future Work

In this paper a standalone approximation simulator tool for simulating various impairments is presented. The tool would assist designers/developers, throughout the phases of the whole development process, in creating accessible Java™ Swing applications. The designers/developers are able to test the functionality implemented for each GUI component. Furthermore, various impairments as well as different severity levels for each impairment can be approximately simulated. Any display high contrast appearance and mouse pointer schemes set by the designer/developer are respected and adapted. Furthermore, information about GUI components with accessibility problems and recommendations on ways to fix these problematic components are provided when simulating colour blindness and Parkinson's disease. There is work on progress in improving approximation simulator capabilities in order to enhance the simulation outcome and/or supporting more impairments, and covering more impairments with information and recommendations about problematic components.

Acknowledgments. This work was partially funded by the EC FP7 project ACCESSIBLE - Accessibility Assessment Simulation Environment for New Applications Design and Development [11], Grant Agreement No. 224145.

References

1. Simon Harper, Ghazalah Khan, Robert Stevens 2005. Design Checks for Java Accessibility. In Accessible Design in the Digital World, UK, accessed 10 January 2010 from: http://www.simonharper.info/publications/Harper2005zr.pdf
2. IBM Adesigner accessed 10 January 2010 from: http://www.alphaworks.ibm.com/tech/adesigner

3. Fujitsu Color Doctor 2.1, accessed 10 January 2010 from: http://www.fujitsu.com/global/accessibility/assistance/cd/
4. VIS simulator, accessed 10 January 2010 from: http://vis.cita.uiuc.edu
5. Vischeck tool, accessed 10 January 2010 from: http://www.vischeck.com
6. WebAIM low vision simulator, accessed 10 January 2010 from: http://www.webaim.org/simulations/lowvision.php
7. Accessibility Inclusive Toolkit – visual impairment simulator, accessed 10 January 2010 from: http://www-edc.eng.cam.ac.uk/betterdesign/downloads/impairmentsims/impairsim2.html
8. Java Accessibility Utilities accessed 10 January 2010 from: http://java.sun.com/javase/technologies/accessibility/docs/jaccess-1.3/doc/index.html
9. Java Accessibility Helper accessed 10 January 2010 from: http://java.sun.com/developer/earlyAccess/jaccesshelper/docs/index.html
10. Netbeans IDE accessed 10 January 2010 from: http://www.netbeans.org
11. T. Oikonomou, K. Votis, D. Tzovaras and P. Korn, "An Open Source Tool for Simulating a Variety of Vision Impairments in Developing Swing Applications", HCII2009, San Diego, USA, LNCS, 5614, 135-144, 2009
12. ACCESSIBLE project accessed 10 January 2010 from: http://www.accessible-project.eu