A Visual Impaired Simulator to achieve Embedded Accessibility Designs

Konstantinos Votis*^, Theofanis Oikonomou*, Peter Korn#, Dimitrios Tzovaras* and Spriridon Likothanasis^

* Informatics and Telematics Institute, 6th km Charilaou-Thermi Road, Thessaloniki, GR-57001 Greece, {kvotis, thoikon, tzovaras}@iti.gr

Sun Microsystems, Inc., 17 Network Circle, MPK17-101, Menlo Park, CA 94025, USA Peter.Korn@Sun.COM

^ Pattern Recognition Laboratory, Computer Engineering and Informatics, University of Patras, Rio Patras {botis, likothan}@ceid.upatras.gr

***Abstract*—The ever increasing of software technologies has brought closer technology to users with visual impaired inabilities. This diversification poses a real challenge to "authors" (developers and designers) when creating software that has to cope with a myriad of interaction situations, as well as specific directives for ensuring an accessible interaction. Presenting an advanced accessibility simulation tool, "authors" can obtain a better understanding of the accessibility constraints for visual impaired users within a Java Swing application by simulating each element. The proposed NetBeans-enabled plugin tool will assist them, with a minimal effort, to explore user-centered design and important accessibility issues for their Java Swing implementations**

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

I. INTRODUCTION

Despite the rapid evolution of Information and Communication Technologies (ICT) over the last years and the increasing acknowledgment of the importance of accessibility, the developers ad designers of mainstream ICT-based products still act and struggle under total absence of structured guidance and support for adjusting their envisaged products and services with their user's real-time accessibility needs. As a result, a critical mass market of ICT based products, targeting older people and people with disabilities, remains highly locked.

Thus, the lack of non accessible software applications can cause large productivity losses, with many people being unable to fully participate at work, in education, or in a wide range of economic and social activities. The existing barriers on accessibility progress, reflects the current fragmented approaches to producing accessible products and services, which rather limit their economic potential and create a barrier to a thriving single market for them in Europe.

Moreover, there is mounting evidence that people with physical, cognitive and behavioural /physiological impairments such as blindness, mobility, cognitive, hearing and speech inabilities, cannot access effectively ICT applications and services, especially due to the deficiencies in the ICT design and development process and the lack of accessibility support tools for developers.

So, many developers and designers are not fully equipped with evidence and knowledge related to the accessibility to 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 adopting aftermarket or 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.

However, each user is different from the next. Monitoring the social situation in Europe with regard to the significant issue of inclusion, an estimate of >11% of the world's population having some kind of disability, indicates that the spectrum of user diversity is enormous[2,3]. This becomes even more complex with the (un)availability of simulation and authoring tools for Java applications, since the power of Web 2.0 brings new sets of possibilities but also several challenges and difficulties on accessibility aspects [4].

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 visual impaired users. It is important that the design and development of accessible ICT solutions be supported in an automated fashion, as much as possible. Thus, although the existing simulation tools are considered to be a good reference for authoring tools, it is currently a work under progress. Thus, developers and designers need tools that provide process-integrated and constructive guidance to them in how to apply the accessibility principles.

This paper presents a Visual Impaired Simulation tool to achieve embedded accessibility design for the development of Java Swing applications, through the NetBeans Integrated Development Environment (IDE).

II. 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 they use relevant simulation tools that can provide an opportunity for users to experience a software application (e.g. Web page) 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 provide certainly information and help designers make user interface content more accessible. These toos can enable, encourage, and assist users ("authors") in the creation of accessible applications.

A good open-source effort, enabling authors to increase the accessibility of their applications, is the NetBeans Accessibility Testing Utilities, well known as the “NetBeans accessibility module”[5]. This is distributed as a separate NetBeans module, which authors can download and install to the NetBeans IDE. The Java Accessibility Helper (provided by the installation NetBeans accessibility module) aids Java software developers in making their JFC-based programs accessible to persons with disabilities. The Helper generates a report that includes a prioritized list of problems and potential problems with the application being tested.

On the visual impaired simulators side, work has already explored for Web applications accessibility as well as for assistive devices as the one that has been proposed by [6]. The aDesigner [7] is a disability simulator that helps Web designers to ensure that their pages are accessible and usable by the visually impaired. Web developers can use aDesigner in order to test the accessibility and usability of Web pages for low-vision and blind people. aDesigner is mostly oriented in testing the degree of colour contrast on the page, the ability of users to change the font size and the existence of alternate text for images. ColorDoctor [8] is a simulator that can check color accessibility. It converts any images displayed on the screen such as websites and other presentation contents into gray scale or colors that can be perceived by people with color blindness. ColorDoctor not only simulates website display, it is also possible to simulate real-time display of proposals, presentations, and moving images such as Flash by selecting the "Transparent" mode. ColorDoctor shows the display content through four conversion filters: Grayscale, Protanopia, Deuteranopia and Tritanopia.

Furtehmore, the Visual Impairment Simulator (VIS) [9] for Microsoft Windows is another tool that simulates what it is on users desktop. When the program runs, it manipulates the images on the user’s screen so that it seems like the user has a visual impairment. Users can choose which impairment they wish to simulate from a drop down menu. The impairments that can be simulated are: Cataract, Color Blindness, Diabetic Retinopathy, Glaucoma, Hyperopia, Macular Degeneration, Magnifier and Retinitis Pigmetosa. The Vischeck [10] tool was created, in order to help web developers check their work for color blind visibility. Also the WebAIM Low Vision Simulator [11] provides users with the opportunity to experience a web page as a user with visual disabilities. As it can be discerned, users can see the specified web page as if they suffer from Macular Degeneration, Cataract or Glaucoma. Finally Cambridge University has developed a vision impairment simulator which is included to the inclusive design toolkit [12]. The vision simulator modifies a digital image to show what the image might look like when viewed with a variety of different vision conditions. Each condition can be applied with different severity levels.

Even though most of visual impairment simulators are considered to be a good guide to the accessibility enhancement of Web sites and Microsoft applications, there are a number of major drawbacks regarding this effort. One major concern is that, they don’t offer visual impaired simulation capabilities for other applications than Web such as Java Swing applications. In addition all of these applications are excluded from well known IDE that developers use for their software implementations. Thus, it is very unlikely that “authors” who are not working exclusively on accessibility issues will ever use them.

III. A SIMULATOR FOR EMBEDDED ACCESSIBILITY DESIGNS

A. Basic Concepts of Simulator

The purpose of our vision simulator is to assist developers and designers to better empathise with those who have reduced vision capabilities, and to help understand how capability loss affects the ability to interact with software applications and services. Our simulator has been designed and developed for the following objectives: 1) A developer-designer aid module for assistance to complete user centered design and accessibility simulation of how a Java Swing application can be viewed for users with visual impairments. 2) A Self-learning software, which can be obtained as a NetBean IDE plugin and/or as a standalone application, in order to present accessibility drawbacks and visual content problems of Java Swing applications

The philosophy behind the implementation presented here is to provide a complete free and open Source software application toolkit that can be offered through two versions, a standalone version and a plug-in version for the NetBeans IDE from Sun Microsystems Inc. It was considered that this kind of tool should be available as widely as possible and should be manipulated easily by experienced or not users. For that reason it was decided to be implemented with the Java programming language and its products that provide us an integrated machine-independent execution environment, simple GUI building facilities and so on. And moreover all users can receive the advantages of the NetBeans development environment that is really accepted and well known by many software developers. In our implementations we used the Sun JavaTM Standard Edition Development Kit (JDK), the Java Accessibility Application Programming Interface (JAAPI) and the Netbeans IDE. A User-centered design method was applied in the implementation of the simulator in order to understand and extract user needs and system requirements as well as to produce appropriate design solutions that should be validated against the extracted requirements. For that reason two iterative phases have been decided to be followed. The first phase

"Information Gathering" was conducted the first trimester of 2009, by performing a literature review survey. The user requirement collection started from the already assessed knowledge provided by the ACCESSIBLE project [13] software developers as well as representative organizations of people with disabilities. Identified users such as developers and designers have been contributing to the second phase, namely the interaction with users, which has been achieved via appropriate questionnaires and interviews. The results from the aforementioned phases were the foundation for the prototype design. In interviews, developers emphasized that a big motivation for the acceptance of a vision impairment simulator should be the including of this tool within an Integrated Development Environment, because most of existing simulators are excluded from well known IDE that developers use for their software implementations. This lead to the design decision to implement a NetBeans plug-in in addition to the standalone application version. The simulator plugin can be installed easily into the Sun NetBeans IDE 6.0, NetBeans IDE 6.1, or NetBeans IDE 6.5.

B. Simulator Funtionalities

The NetBeans plugin version of the simulator includes the preview design and the run main simulation functionalities. The "Preview Design" provides users a visual design preview feature that allows them to see how their implemented Swing forms can be displayed with visual impairments. Preview Design enable developers to preview the form layout of their implementations like the simple preview design functionality of the NetBeans application. Before NetBeans 6.0, there was no mechanism to allow the project's look and feel to be set in the designer for a regular Swing application. It was left to the developer to add the appropriate code to set the application look and feel. Thus, it is obvious that the supported functionality within NetBeasn 6.0 and above enhanced the development of the preview design vision impaired simulator. The preview design functionality can be activated by clicking on the appropriate small image of an eye with an arm-chair preview icon, at the IDE's Toolbar (Figure 1).

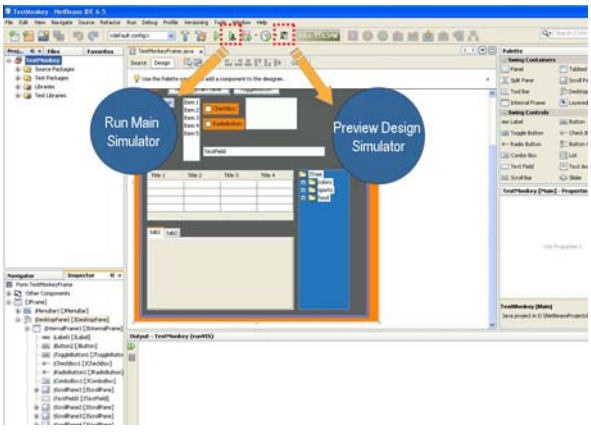

Figure 1. NetBeans with new modules installed

Even if the "preview design" functionality can overcome the gap between developer's knowledge on accessibility issues and the development of accessible and tailored software Java Swing applications, however, the real-time interactions of developers with their Swing application forms are really important within the lifecycle design of their implementations. Thus, static and non interactive look and feel views of applications are fine when developers and designers developing independent forms, but this situation it doesn't assist them concerning how form components should behave as the application is running in real time, where relevant modifications came about (e.g. modification of size, usage of combo boxes and buttons, change the ordering of tabs, etc). To overcome this, "Run Main" functionality has been implemented in order to give the ability to users to explore, run and test their implementations. While their applications are running new windows, such as dialogs, choosers or frames, may appear due to user interactions. This functionality of the simulator can be activated by clicking on the appropriate image of a green triangle with an arm-chair, at the IDE's Toolbar. An overview of the "Preview Design" simulator is presented to the following figure 2.

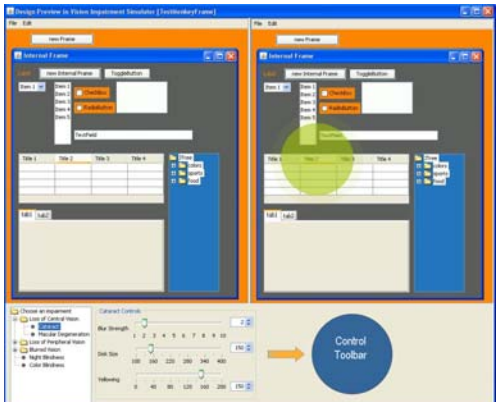

Figure 2. NetBeans "preview design" module

The vision impairment simulator allows users to create and/or open Java Swing applications and view them through the graphical view of the GUI components, with selected simulated impairment This figure demonstrates how the simulator is working on the NetBeans IDE 6.5 where a developer uses the installed plug-in in order to simulate cataract impairment within a Swing application form. The Graphical User Interface (GUI) makes it possible, where appropriate, to control the parameters of the whole simulation. Thus, for cataract the user can modify the blur strength, the disk size and the yellowing attributes through the cataract control toolbar as depicted to figure 3.

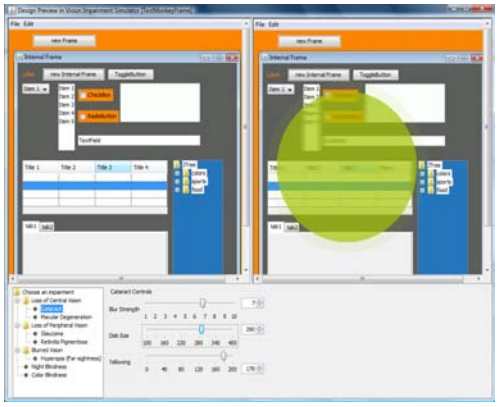

Figure 3. "preview design" control impairment

The usage of the "Run Main" module is depicted to the following figure 4. This is a sample view of the program as it simulates what a person with Hyperopia could experience when navigates an implemented Swing application. It is obvious that the user can be navigated through the offered services/functionalities of its implementation (e.g. pressing buttons, inserting text) within the left area of the application, in parallel with the simulation application running in the right area. Thus, each modification to the Swing application is depicted to the simulator application view area. In the same way with the "preview design" module, the GUI of the simulator provides control functionality through an offered control toolbar. Thus, by using the control toolbar developers and designers can modify the range of the appropriate variables to specify the type of vision impairments.

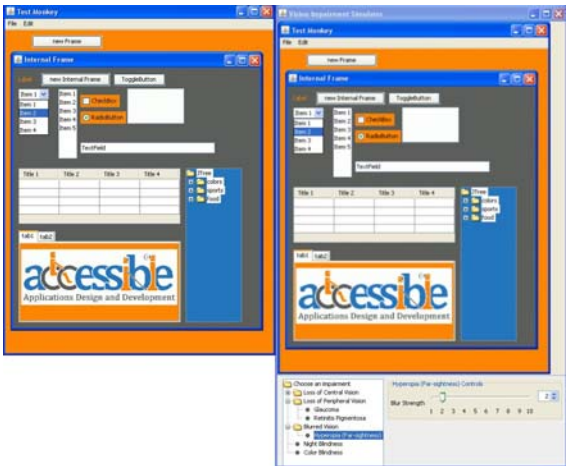

Figure 4. NetBeans "Run main" module

Finally, it is important to mention that the usage of a vision impairment simulator, like the proposed one, does not portray exactly how it is like to have a single or a combination of vision impairments, however, fully-sighted users who spending some time using the simulator's functionalities, through 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.

C. Vision Imapirement Capabilities

Our simulator provides a reasonably accurate picture of some of the functional limitations and abilities that may be experienced with different types of visual impairments. Some of the most common causes of low vision in the developed world are cataracts, hyperopia, macular degeneration, glaucoma, etc. In order to proceed in a structured manner, the selection of the visual impairments that have been included to the simulator was based on the body functions from the International Classification of Functioning, Disability and Health (ICF) [14] and translate them into relevant functional limitations. An example for the loss of central vision is depicted to the following table 1.

The simulator has been developed in order to support a variety of low vision impairments such as loss of central and peripheral vision, blurred vision, extreme light sensitivity, night blindness and color blindness

Table 1. Example for the loss of central vision impairment

Disability(ies)	Functional limitations	ICF classification
Vision impairments **Low vision** ▪ **Loss of central vision**	The loss of central vision creates a blur or blind spot, but side (peripheral) vision remains intact. This makes it difficult to read, recognise faces, and distinguish most details in the distance. Mobility, however, is usually unaffected because side vision remains intact.	b210 Seeing functions b2100 Visual acuity functions b21000 Binocular acuity of distant vision b21001 Monocular acuity of distant vision b21002 Binocular acuity of near vision b21003 Monocular acuity of near vision b21008 Visual acuity functions, others specified b21008 Visual acuity functions, unspecified b2101 Quality of vision b21023 Visual picture quality b21028 Quality of vision other specified b21029 Quality of vision, unspecified b2108 Seeing functions, other specified b2109 Seeing functions, unspecified

Within color blindness capability the simulator supports Protanopia, Protanomaly, Deyteranopia, Deyteranomaly, Tritanopia, Tritanomaly, Achromatopsia and Achromatomaly. These 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.

IV. EVALUATION RESULTS

A first comparative evaluation analysis has been performed in order to verify design decisions and get input for modifications in future iterations. The evaluation comprised of a preliminary free exploration of the simulator's visual impairment capabilities in comparison with existing vision impairment simulators. Thus, we have compared the functionality of our simulator with the functionalities of the Accessibility Colour Wheel [15], Inclusive Design Toolkit, Adesigner, ART [16], Colour Doctor, Colour Blindness [17], Vischeck, VIS, and WebAIM Low Vision simulators. An overview of the analysis is presented to following table 2, where we present the supported vision impairments of well known vision impairment simulators in addition to our implemented simulator.

The results from the evaluation indicated that although most of existing simulation tools are considered to be a good reference for the simulation of Web pages and images, however there is a obvious shortage of simulators for Java Swing applications. Actually it should be considered that the support of these kinds of applications in contrast with Web and media content (e.g. images, presentations, etc.) is a work under progress. In addition we have noticed that most of existing vision impairment simulators can be used as independent applications and they aren't offered through common Integrated Development Environments. For that reason it is very unlikely that potential developers and designers who are not working exclusively on accessibility issues will ever use them for their implementations. The reality is that developers often don't want to use applications that are excluded from the main IDE software packages they prefer to use. This is the reason that we have decided to create a simulator that should be provided to users not only as a standalone application but also as an official plug-in for NetBeans IDE from Sun Microsystems organisation.

Table 2. Comparative Evaluation Results.

Tools / Vision Impairment	Our	Accessibility Colour Wheel	Inclusive Design Toolkit	ADesigner	ART Simulator	Color Doctor	Colour Blindness Simulator	Vischeck	VIS	WebAIM Low Vision Simulator
Deuteranope	x	x	x	x	x	x	x	x	x	
Tritanope	x	x	x	x	x	x	x	x	x	
Protanope	x	x	x	x	x	x	x	x	x	
Grayscale	x					x			x	
Cataract	x		x						x	x
Diabetic Retinopathy			x						x	x
Glaucoma	x		x						x	x
Hyperopia	x								x	x
Macular Degeneration	x		x						x	x
Retinitis Pigmetosa	x								x	
Protanomaly	x									
Deyteranomaly	x									
Tritanomaly	x									
Achromatopsia	x									
Achromatomaly	x									

V. Conclusion and Future Work

Developers and designers often have difficulty understanding the problems users with visual disabilities face when accessing their software applications that are not designed with their needs in mind. The proposed simulator would assist developers in creating accessible Java Swing applications by simulate most of vision impairments. This simulation provides an opportunity for "creators" to view their implemented components, using simulated visual impairments, in order to understand how these can impact their work and how can better designed with accessibility aspects. By this way they can overcome accessibility barriers and improve the overall accessibility of their Java applications. Ongoing work is currently being done in several fronts, including: improving the simulator capabilities to cover more vision impairment simulation situations, (2) to extend the functionalities of our tool in order to apply the same simulation techniques to new and innovative JavaFX [18] applications that can be used for the development of Rich Internet applications, and (3) to include our tool within the official release of next version of Netbeans with the collaboration of Sun Microsystems.

Acknowledgment

This work was partially funded by the EC FP7 project ACCESSIBLE - Accessibility Assessment Simulation Environment for New Applications Design and Development, 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 06 May 2009 from: http://www.simonharper.info/publications/Harper2005zr.pdf

[2] Eurostat yearbook 2008 , accessed 06 May 2009 from: http://ec.europa.eu/eurostat

[3] Report of the Inclusive Communications (INCOM) subgroup of the Communications Committee (COCOM), COCOM04-08, accessed 22 June 2009 from: http://ec.europa.eu/information_society/activities/einclusion/docs/access

[4] Brian Kell, David Sloan, Stephen Brown, Jane Seale, Helen Petrie, Patrick Lauke, Simon Ball 2007. Accessibility 2.0: people, policies and processes. ACM International Conf. Proceeding Series; Vol. 225, Pages: 138 – 147. DOI= http://doi.acm.org/10.1145/1243441.1243471

[5] NetBeans Accessibility module accessed 28 April 2009 from: http://a11y.netbeans.org

[6] Pradipta Biswas 2007. Simulating HCI for special needs. ACM SIGACCESS Accessibility and Computing archive Issue 89, Pages 7-10, ISSN:1558-2337, DOI= http://doi.acm.org/10.1145/1328567.1328569

[7] IBM Adesigner accessed 28 April 2009 from: http://www.alphaworks.ibm.com/tech/adesigner

[8] Fujitsu Color Doctor 2.1, accessed 28 April 2009 from: http://www.fujitsu.com/global/accessibility/assistance/cd/

[9] VIS simulator, accessed 28 April 2009 from: http://vis.cita.uiuc.edu

[10] Vischeck tool, accessed 28 April 2009 from: http://www.vischeck.com

[11] WebAIM low vision simulator, accessed 28 April 2009 from: http://www.webaim.org/simulations/lowvision.php

[12] Accessibility Inclusive Toolkit – visual impairment simulator, accessed 28 April 2009 from: http://www-edc.eng.cam.ac.uk/betterdesign/downloads/impairmentsims/impairsim2.html

[13] FP7 strep project ACCESSIBLE - Accessibility Assessment Simulation Environment for New Applications Design and Development, Grant Agreement No. 224145, http://www.accessible-project.eu/

[14] ICF - International Classification of Functioning, Disability and Health. World Health Organization. accessed 10 March 2009 from: http://www.who.int/classifications/icf

[15] Accessibility Colour Wheel tool, accessed 28 June 2009 from: http://gmazzocato.altervista.org/colorwheel/wheel.php

[16] ART simulator, accessed 28 May 2009 from: http://www.ubaccess.com/artsimulator.html

[17] Colour Blindness Simulator –Etre, accessed 15 July 2009 from: http://www.etre.com/tools/colourblindsimulator/

[18] JavaFX software applications, accessed 15 July 2009 from: http://www.sun.com/software/javafx