

Web Toolkits Accessibility Study

Zdenek Mikovec, Jan Vystrcil, Pavel Slavik

Czech Technical University in Prague, Faculty of Electrical Engineering, Dept. of computer graphics and interaction
{xmikovec, vystrcjan, slavik}@fel.cvut.cz

Introduction

Access to increasing number of web applications is very good news especially for disabled people. These applications become richer and offer wide variety of services used in everyday's life (document creation, instant messaging, data storage, email communication, social networking, etc.). The problem is that at the same time these rich internet applications (RIA) become less accessible. There is a Web Accessibility Initiative (WAI) 2 related to W3C 1 which is trying to cope with this problem. In particular the Accessible Rich Internet Application suite of W3C (WAI-ARIA) 3 is describing how to improve the accessibility of dynamic web content and advanced user interface controls in rich internet applications.

The accessibility issues covered by WAI-ARIA can be divided into three main categories from the user interaction point of view:

- keyboard navigation in the application,
- perception of presented information,
- and user input.

RIA toolkits introduce new complex UI components and dynamically changing content. This brings problems especially in the area of keyboard navigation, which is essential for the whole web application accessibility. The question is how serious these accessibility problems are, what the assistive tools can manage and in what extent, and how can we efficiently fix these problems by means of implementation of WAI-ARIA recommendations.

We are trying to answer these questions in a framework of Success project 4. This three year project is funded by Sun Microsystems 5. The result of this project should be improvement of the web application development process which will result in increase of RIA accessibility. The project has three stages – first is focusing on the web toolkits, second on the integrated development environment and third on the disabled developers. From the end-user point of view we focus on the users with visual impairment where the consequences of missing accessibility solutions are most serious.

In this report we will describe the first part of our project – the analysis of two existing RIA web toolkits from the accessibility point of view. We will identify the most important accessibility problems and propose fixes by means of implementation of WAI-ARIA recommendations.

The Study

Test setup

For our study we have chosen two opensource web toolkits – Ext JS 2.2.1 6 and ICEfaces 1.8.0 7 which provide server based and client based solutions for creation of rich internet

applications. In these toolkits we see big potential to develop at the end highly accessible toolkits.

Ext JS is a JavaScript library for RIA development which is available on commercial and also open source licenses. Set of JS libraries are stored on a server and functions are called from client's web browser. Component behavior can be easily modified by setting property parameters.

ICEfaces is an open source RIA framework which enables easy development and deployment of Java EE AJAX based applications. ICEfaces applications are pure JavaServer Faces (JSF) applications so there is no need for any JavaScript related development. Toolkit, which consists of approximately 50 components, can be integrated to common IDE's like NetBeans or Eclipse. For beginners it's more difficult than Ext JS to be used because of necessary Java EE development knowledge.

As a reference toolkit the Dojo web toolkit 8 was chosen.

We have prepared two test setups. First was running on OpenSolaris 2008.11 9 with Firefox 3.0 browser 10 and Orca 2.24 reader 11. The second was running on MS Windows 12 with Firefox 3.0 browser and JAWS 10.0 reader 13.

Testing approach

In our tests we wanted to check the accessibility of real-life applications developed in real-life development process. For this reason we avoided the formal check of the implementation of the ARIA recommendations in the components generated by the toolkits. We also avoided checking of isolated components or pure usage of some predefined test examples.

In our case we took 21 developers who used selected toolkits and tried to develop own non-trivial real-life applications. Three were 10 developers using Ext JS, 8 using ICEfaces and 3 using Dojo. Examples of some real-life applications developed are depicted in **Fig. 1**, **Fig. 2** and **Fig. 3**.

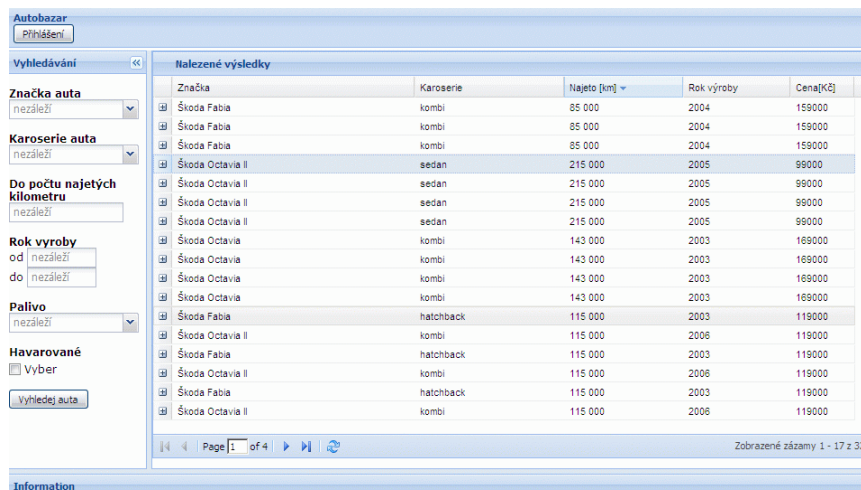


Fig. 1. Car shop website. This web represents a typical car shop web applications, where the user can login (dialogue window will popup), chose from number of listboxes and browse in table with cars with special features (sorting, expansion, paging).

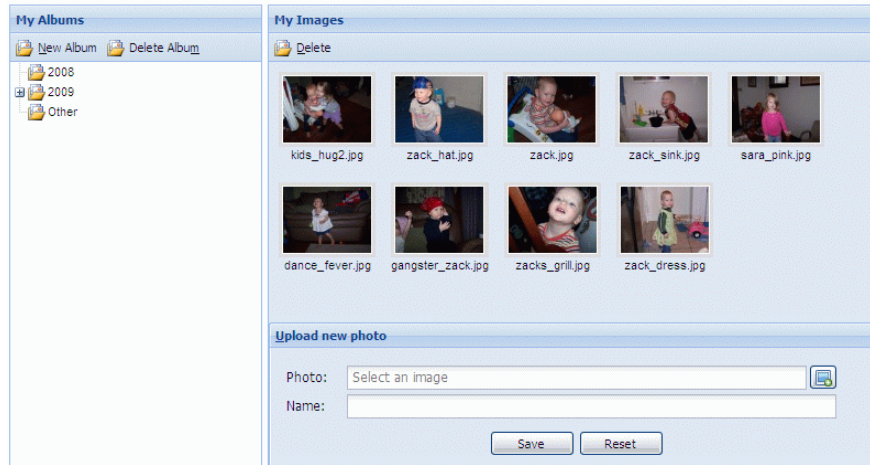


Fig. 2. Online photo album. This web represents typical photo album application, with possibility to upload photos, organize them in albums and tree structure representing time.

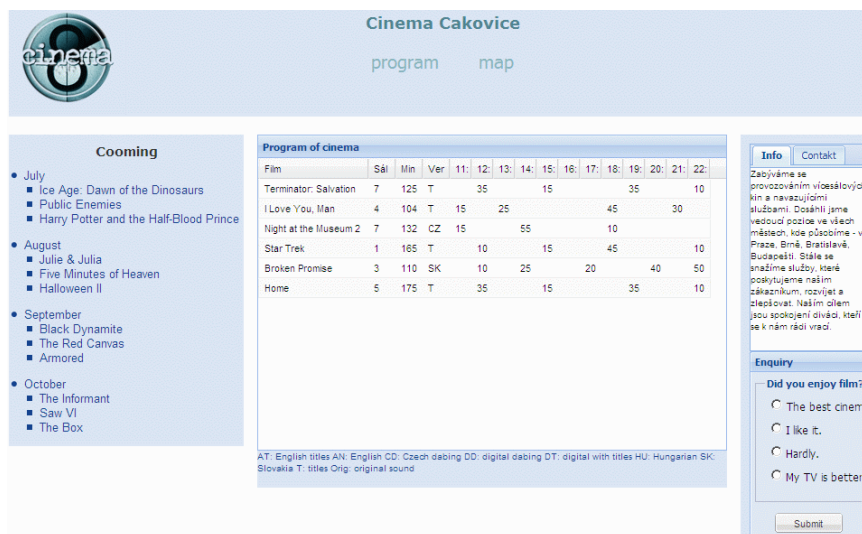


Fig. 3. Online cinema. This web represents a website of cinema complex, where the user can search for offered movies.

These applications were then searched for accessibility issues with Orca or JAWS reader. The tests were not intended to provide a comparative study of selected toolkits. Our goal was to cover as broad as possible range of real-life applications.

With this testing approach we have got more than 10 test applications which are very close to those developed by programmers in everyday life.

The web application components occurred in various contexts. For instance the grid component appeared in a context of a viewport, of a menu structure (see **Fig. 4**) or directly in the body of an HTML page. For each context the accessibility of such component differed.

Our test applications did not by nature cover all the existing components, but the most likely used ones.

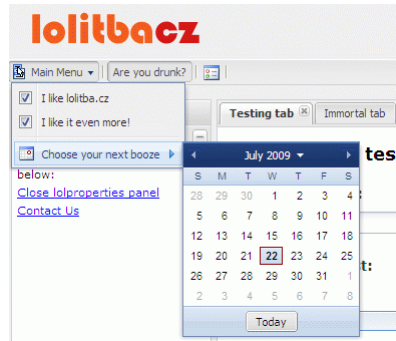


Fig. 4. Components in various contexts. Checkbox and grid (calendar) as a part of menu.

Accessibility issues analysis

Our test has identified accessibility issues in 30 components of Ext JS toolkit (in the time of finishing the report the new version of Ext JS was released, which should address some ARIA issues) and 18 components of ICEfaces toolkit. In the reference toolkit Dojo, which is considered as the best implementation in the means of ARIA recommendations, we have identified 15 components reporting an accessibility problem. The total number of accessibility issues found is larger than 100 (approximately 50 in Ext JS, 40 in ICEfaces and 20 in dojo toolkit). Table 1 provides is a list of the most important issues found in both toolkits.

component	issue description	major cause
<ul style="list-style-type: none"> tab panel 	<ul style="list-style-type: none"> impossible to open another tab after entering one 	<ul style="list-style-type: none"> missing roles tabpanel, tablist, tab missing state aria-hidden
<ul style="list-style-type: none"> collapsible panel 	<ul style="list-style-type: none"> when collapsed disappears for the reader 	<ul style="list-style-type: none"> missing state aria-expanded
<ul style="list-style-type: none"> grid (list of structured items) 	<ul style="list-style-type: none"> navigation in the grid not possible (in one row only, no information about current cell) 	<ul style="list-style-type: none"> each row is a special table (this disables the reader table navigation feature) missing roles application, grid, gridcell, row, columnheader, rowheader missing state aria-selected
<ul style="list-style-type: none"> grid (calendar) 	<ul style="list-style-type: none"> strange behavior of the reader navigation within the grid; no information about the current cell 	<ul style="list-style-type: none"> missing role application missing state aria-labelledby, aria-describedby
<ul style="list-style-type: none"> dialog 	<ul style="list-style-type: none"> displayed dialog is ignored by the reader 	<ul style="list-style-type: none"> missing roles dialog missing state aria-hidden
<ul style="list-style-type: none"> alert, dialog 	<ul style="list-style-type: none"> reader ignores title and description 	<ul style="list-style-type: none"> missing state aria-labelledby, aria-describedby
<ul style="list-style-type: none"> messagebox 	<ul style="list-style-type: none"> reader reads the focused button only; other buttons are ignored 	<ul style="list-style-type: none"> missing role button
<ul style="list-style-type: none"> checkbox (ownerdraw) 	<ul style="list-style-type: none"> reader does not inform the user about the current state of the checkbox 	<ul style="list-style-type: none"> missing role checkbox missing state aria-checked
<ul style="list-style-type: none"> chat window 	<ul style="list-style-type: none"> if new message arrives, the reader starts reading form the beginning of the chat session 	<ul style="list-style-type: none"> missing state aria-activedescendant

Table 1: The most important issues found in both toolkits

Conclusion

The main result of our study is, that real-life applications (more than 10) created by chosen web toolkits are seriously inaccessible. Ext JS 2.2.1 (the version 3.0 already implements some WAI-ARIA recommendations) and ICEfaces 1.8.0 which does not implement WAI-ARIA recommendation unfortunately produce inaccessible application if browsed by screen reader. Even Dojo web toolkit which tries to implement the WAI-ARIA records serious accessibility issues.

It is essential to implement recommendations of WAI-ARIA properly in order to make the rich internet application accessible. The most serious issues are of the keyboard navigation nature. Based on our study we have identified five biggest accessibility issues (see Table 2). Each issue is described by a recommendation and WAI-ARIA implementation proposal.

Recommendation	WAI-ARIA implementation
<ul style="list-style-type: none"> Distinguish between "application" and "document" mode of the reader. 	<ul style="list-style-type: none"> Implement role attribute application whenever you want to avoid interference of keyboard navigation defined by your application with the reader's navigation. You will need to provide references to the labels (aria-labeledby and aria-describedby attributes).
<ul style="list-style-type: none"> Define states of owner draw user input widgets (e.g., checkboxes, comboboxes, trees). 	<ul style="list-style-type: none"> Implement state attributes (e.g., aria-checked, aria-expanded, aria-selected). These states must be updated by JavaScript or CSS (visual effect will be bind with accessibility states).
<ul style="list-style-type: none"> Define roles for structured user interface elements (e.g., tabpanels, menus with checkboxes and radios, trees, expandable grids, toolbars). 	<ul style="list-style-type: none"> Implement role attribute (e.g., menuitemradio, menuitemcheckbox, tabitem, treegrid).
<ul style="list-style-type: none"> Define roles for complex document structures (e.g., grids). 	<ul style="list-style-type: none"> Implement role attribute (e.g., grid).
<ul style="list-style-type: none"> Properly describe alerts and dialog windows. 	<ul style="list-style-type: none"> Implement role attribute dialog and alertdialog and state aria-hidden.

Table 2: Recommendations and WAI-ARIA implementations

Acknowledgement

This research was funded by the Sucess project – Sun Center of Excellence for Accessibility & Usability (<http://amun.felk.cvut.cz/coe/>).

References

1. W3C Consortium. <http://www.w3.org/>, retrieved July 2009
2. Web Accessibility Initiative of W3C. <http://www.w3.org/WAI/>, retrieved July 2009
3. Accessible Rich Internet Applications Suite of W3C. <http://www.w3.org/WAI/intro/aria.php>, retrieved July 2009
4. Sucess Project: Sun Center of Excellence for Accessibility & Usability. <http://amun.felk.cvut.cz/coe/>, retrieved July 2009
5. Sun Microsystems, Inc. <http://www.sun.com/>, retrieved July 2009

6. Ext JS web Toolkit. <http://extjs.com/>, retrieved July 2009
7. ICEfaces web toolkit. <http://www.icefaces.org/>, retrieved July 2009
8. Dojo web toolkit. <http://www.dojotoolkit.org/>, retrieved July 2009
9. OpenSolaris operating system. <http://opensolaris.org/>, retrieved July 2009
10. Firefox web browser. <http://www.mozilla.com/firefox/>, retrieved July 2009
11. Orca screen reader. <http://live.gnome.org/Orca/>, retrieved July 2009
12. MS Windows XP operating system. <http://www.microsoft.com/windows/windows-xp/>, retrieved July 2009
13. JAWS screen reader. <http://www.freedomscientific.com/products/fs/jaws-product-page.asp>, retrieved July 2009

About the authors:



Zdenek Mikovec is a researcher and a lecturer at the CTU in Prague, Department of Computer Science and Engineering. He received his PhD in 2007 at the same university. His fields of interest include formal picture description, special user interfaces, and XML/XSL, with a focus on the blind and visually impaired users. Between 2001 and 2002 he was working on adaptation of multimedia documents on PDA at ZGDV Darmstadt, Germany (analysis, design, prototype implementation). After his return to Prague he became involved in the EU project Mummy and ELU. Currently he is active in the projects i2home, AEGIS and VitalMind. He is an author or a co-author of more than twenty publications on various international events in HCI.



Jan Vystrèil is a researcher and a lecturer at the CTU in Prague, Department of Computer Graphics and Interaction. His master thesis "User interface for in-door navigation of visually impaired people via mobile phone" was awarded with 2nd price in national Master thesis competition. In his research he is interested in development of navigation system NaviTerier for visually impaired people and also other interaction with such impaired users. Currently he is involved in the EU funded project AEGIS and Sun Center of Excellence for Accessibility, where he focuses on accessibility of rich internet applications.



Pavel Slavík is a full professor at the Department of Computer Science and Engineering at the CTU in Prague, where he received his PhD (1983), Associate Professor (1994), and Professor (2003) titles. His wide range of interests includes computer graphics, scientific visualization, graphic user interfaces, and interfaces for users with special needs. He is an author or co-author of more than 200 scientific papers and articles in journals and conference proceedings. He has an extensive record of participation in various research projects, including INCO projects (WISE/E 1995-96, Virtuos 1998 – 2000), or 5th Framework projects (ENORASI, 2000 – 2001, MUMMY 2001 – 2005), or 6th Framework projects (ELU, i2home), or 7th Framework projects (AEGIS, VitalMind).