DOCTOR OF SCIENCES

Methodology for automating web usability and accessibility evaluation by guideline

Beirekdar, Abdo

Award date:
2004

Awarding institution:
University of Namur

Facultés Universitaires
Notre-Dame de la Paix

A Methodology for Automating Guideline Review of Web Sites

Abdo Beirekdar

Thesis submitted in fulfillment of the requirements for the degree of Doctor of Sciences
(Computer Science Option)

- August 30th, 2004 -

Director:	Professor M. Noirhomme-Fraiture
Co-director:	Professor J. Vanderdonckt, Université Catholique de Louvain, Belgium
Jury:	Professor F. Bodart
	Professor J.-L. Hainaut (President)
	Professor Ch. Kolski, Université de Valenciennes, France
	Professor Ph. Palanque, Université Paul Sabatier - Toulouse III, France

Institut d'Informatique
NAMUR

Annex C

Web Guidelines Integrated by Major Existing Evaluation Tools

In this annex we list the guidelines evaluated by the tools reviewed in Chapter2.

1. W3C WCAG1.0

These guidelines explain how to make Web content accessible to people with disabilities. The guidelines are intended for all Web content developers (page authors and site designers) and for developers of authoring tools. The primary goal of these guidelines is to promote accessibility. However, following them will also make Web content more available to all users, whatever user agent they are using (e.g., desktop browser, voice browser, mobile phone, automobile-based personal computer, etc.) or constraints they may be operating under (e.g., noisy surroundings, under- or over-illuminated rooms, in a hands-free environment, etc.). Following these guidelines will also help people find information on the Web more quickly. These guidelines do not discourage content developers from using images, video, etc., but rather explain how to make multimedia content more accessible to a wide audience.

1. Provide equivalent alternatives to auditory and visual content.
2. Don't rely on color alone.
3. Use markup and style sheets and do so properly.
4. Clarify natural language usage
5. Create tables that transform gracefully.
6. Ensure that pages featuring new technologies transform gracefully.
7. Ensure user control of time-sensitive content changes.
8. Ensure direct accessibility of embedded user interfaces.
9. Design for device-independence.
10. Use interim solutions.
11. Use W3C technologies and guidelines.
12. Provide context and orientation information.
13. Provide clear navigation mechanisms.
14. Ensure that documents are clear and simple.

2. Section508 Guidelines

In 1998, Congress amended the Rehabilitation Act to require Federal agencies to make their electronic and information technology accessible to people with disabilities. Inaccessible technology interferes with an individual's ability to obtain and use information quickly and easily. Section 508 was enacted to eliminate barriers in information technology, to make available new opportunities for people with disabilities, and to encourage development of technologies that will help achieve these goals. The law applies to all Federal agencies when they develop, procure, maintain, or use electronic and information technology.

1. A text equivalent for every non-text element shall be provided (e.g., via "alt", "longdesc", or in element content).
2. Equivalent alternatives for any multimedia presentation shall be synchronized with the presentation.
3. Web pages shall be designed so that all information conveyed with color is also available without color, for example from context or markup.
4. Documents shall be organized so they are readable without requiring an associated style sheet.
5. Redundant text links shall be provided for each active region of a server-side image map.
6. Client-side image maps shall be provided instead of server-side image maps except where the regions cannot be defined with an available geometric shape.
7. Row and column headers shall be identified for data tables.
8. Markup shall be used to associate data cells and header cells for data tables that have two or more logical levels of row or column headers.
9. Frames shall be titled with text that facilitates frame identification and navigation.
10. Pages shall be designed to avoid causing the screen to flicker with a frequency greater than 2 Hz and lower than 55 Hz.
11. A text-only page, with equivalent information or functionality, shall be provided to make a web site comply with the provisions of this part, when compliance cannot be accomplished in any other way. The content of the text-only page shall be updated whenever the primary page changes.
12. When pages utilize scripting languages to display content, or to create interface elements, the information provided by the script shall be identified with functional text that can be read by assistive technology.
13. When a web page requires that an applet, plug-in or other application be present on the client system to interpret page content, the page must provide a link to a plug-in or applet.
14. When electronic forms are designed to be completed on-line, the form shall allow people using assistive technology to access the information, field elements, and functionality required for completion and submission of the form, including all directions and cues.
15. A method shall be provided that permits users to skip repetitive navigation links.
16. When a timed response is required, the user shall be alerted and given sufficient time to indicate more time is required.

3. LIFT

The last version of LIFT enables the evaluation of two sets of guidelines: W3C WACG1.0 and Section508 guidelines.

In fact, in it WAI, W3C proposes 14 guidelines at a high abstract level, and provided what they called techniques that can be used to check the respect/violation of these guidelines. These techniques correspond to what we called interpretation of the guideline. In its evaluation module, LIFT goes directly to these techniques and it only mentions to which W3C guideline the technique is related.

According to the help of the LIFT Accessibility extension to Macromedia DreamweaverMX[1], LIFT evaluates and can fix (automatically or manually) the following guidelines:

3.1 W3C WCAG1.0 Priority 1 Guidelines

The following table lists the Priority 1 guidelines evaluated by LIFT. In fact, LIFT recently proudly announced that its new version can evaluate Section508 guidelines in addition to W3C WCAG1.0, but when we examine the help of LIFT, we find that W3C Priority 1 guidelines and Section508 guidelines are almost the same (two guidelines are special to Section508 and two other are special to W3C) as the following table shows. **S**: Section508, **W**: W3C, **Man**: LIFT fixes the error manually, **Auto**: LIFT can fix the error automatically.

Guideline	Source	Fixing
Non spacer IMG with valid ALT	S, W1.1	Man
Spacer IMG with valid ALT	S, W1.1	Auto
Non spacer IMG with equivalent ALT	S, W1.1	Man
Non spacer IMG with valid LONGDESC	S, W1.1	Auto
No LONGDESC for spacer IMG	S, W1.1	Auto
Non spacer IMG needs LONGDESC	S, W1.1	Auto
IMPUT with valid ALT	S, W1.1	Auto
INPUT with equivalent ALT	S, W1.1	Man
SCRIPT with valid NOSCRIPT	S, W1.1	Auto
SCRIPT with equivalent NOSCRIPT	S, W1.1	Man
AREA with valid ALT	S, W1.1	Auto
AREA with equivalent ALT	S, W1.1	Man
Color is not essential	S, W2.1	Man
Style sheets should not be necessary	S, W6.1	Man
Links are needed for server-side image map	S, W1.2	Man
No server-side image maps should be used	S, W9.1	Man
Data tables should have headers	S, W5.1	Man
Cell of data table should refer to headers	S, W5.1	Man
Data tables should be defined by TABLE tag	S, W5.1	Man
Multiple headers should be marked in data tables	S, W5.2	Man
FRAME with valid TITLE	S, W12.1	Auto
IFRAME with valid TITLE	S, W12.1	Auto
Text only equivalent page may be needed	S, W6.2	Man
Scripts are accessible	S, W6.3,8.1	Man

[1] Dreamweaver MX is a trade mark of Macromedia Inc. (www.macromedia.com)

Link to plug-in is present	**S**	**Man**
APPLET with valid ALT	S, W1.1	Auto
APPLET with valid CONTENT	S, W1.1	Auto
APPLET with equivalent ALT	S, W1.1	Man
Form is accessible	**S**	**Man**
No javascript links are used	S, W6.5	Auto
No auto refresh is used	S, W7.4	Auto
No auto redirect is used	S, W7.5	Auto
GIFs do not cause the screen to flicker	S, W7.1	Man
Avoid causing the screen to flicker	S, W7.1	Man
Image OBJECT with valid CONTENT	S, W1.1	Auto
Image OBJECT with equivalent CONTENT	S, W1.1	Man
Audio/video OBJECT with valid CONTENT	S, W1.1	Man
Audio/video OBJECT with equivalent CONTENT	S, W1.1	Auto
OBJECT with valid CONTENT	S, W1.1	Auto
OBJECT with equivalent CONTENT	S, W1.1	Man
Multimedia with synchronized alternative	S, W1.4	Man
Multimedia with equivalent auto description	S, W1.3	Man
Use clear language for site's content	**W14.1**	**Man**
Clarify natural language usage	**W4.1**	**Man**
Link AUDIO with equivalent CONTENT	S, W1.1	Man

3.2 W3C WCAG1.0 Priority 2 Guidelines

LIFT enables also the evaluation of W3C WCAG1.0 Priority 2 guidelines. Again, these are the techniques proposed by W3C to evaluate priority 2 accessibility issues related to its set of guidelines. The following table lists these guidelines.

Guideline	Fixing
Ensure sufficient contrast between foreground and background colors	Man
Use markup rather than images [3.1]	Man
Document should be valid with respect to published grammars [3.2]	Man
Use style sheets [3.3]	Auto
Use percentage values for frame sizes [3.4]	Auto
Use percentage values for table sizes [3.4]	Auto

Use relative units in external CSS [3.4]	Man
Use relative CSS units in markup [3.4]	Auto
Use header elements according to specification [3.5]	Auto
Properly mark up lists and related items [3.6]	Man
Mark up quotations [3.7]	Man
Avoid using quotation markup for formatting effects [3.7]	Man
Layout tables should make sense when linearized [5.3]	Man
Avoid using structural markup for visual formatting [5.4]	Man
Use device-independent event handlers [6.4]	Auto
Avoid javascript links [6.5]	Auto
FRAMESET should have valid NOFRAMES [6.5]	Auto
Dynamic content should be accessible [6.5]	Man
Avoid causing content to blink [7.2]	Auto
Avoid objects that may cause the screen to blink [G 7.2]	Man
GIFs should not cause the screen to blink [7.2]	Man
GIFs should not cause movement [7.3]	Man
Objects should not cause movement [7.3]	Man
Avoid auto-refreshing pages [7.4]	Auto
Avoid to redirect pages by markup [7.5]	Auto
Jump menu should be device-independent [9.2]	Auto
Objects should have device-independent interface [9.2]	Man
Specify logical event handlers [10.1]	Auto
Inform users if new windows appear [10.1]	Man
Inform users if new pop-up windows appear [10.1]	Man
Properly position implicit labels in forms [10.2]	Man
Use last appropriate W3C technologies [11.1]	Man
Avoid deprecated features of W3C technologies [11.2]	Auto
Describe frames and their relationship [12.2]	Auto
Divide information into appropriate manageable groups [12.3]	Man
Add explicit labels to form controls [12.4]	Auto
Clearly identify the target of each link [13.1]	Man
Use meaningful labels for links [13.1]	Auto
Provide metadata to pages and sites [13.2]	Auto
Provide information about site organization [13.3]	Man

Use navigation mechanisms appropriately [13.4]	Man

4. DoctorHTML

The tool conducts test related to the following issues:

- **Document Structure:** This feature tests the overall document structure (excluding tables and forms, which are dealt with separately). The test looks for unclosed HTML codes which may cause problems on some browsers. It also checks to make sure some important tags are present, warning you if they are missing. When used in conjunction with "Show command hierarchy", this report can be helpful in hunting down extra or missing HTML tags.
- **Verify Hyperlinks:** This test is for finding dead hyperlinks on Web pages. It reports whether the URL is still present (though it may no longer be pointing to its original destination), or if the server returns an error. To make this feature work with a typical number of links on a page (about 30), the timeout for each link test is 10 seconds. This may cause some slow links to timeout, and you will have to check them manually. The report also informs you how large the destination URL is, so that you can check unusually small returns for short messages such as "This page has moved!"
- **Spelling:** This test looks for spelling errors in the document. It removes HTML directives and accented text before running the document through a spelling checker, eliminating most of the false alarms. You can also set up customized word lists to supplement the built-in (English) dictionary.
- **Image Syntax:** This test checks each image command for HEIGHT, WIDTH and ALT tags, and reports if they are absent. These tags are important for quick image loading and page formatting, as well as providing information for browsers lacking image support and users who have turned off image auto-loading.
- **Image Analysis:** This test examines all of the images in a Web page and determines a few important properties for each image. The most important information this feature provides is the bandwidth consumed by each image, and roughly how long it will take to download over a 14.4kbps modem (currently the most common speed for dial-up access users). Excessively long load times for individual images are highlighted in different shades of red. The program also reports the dimensions of the images (in pixels) and number of colors in the image (which has a direct bearing on how much bandwidth the image consumes).
- **Meta Tags:** This test checks the presence and syntax of META tags on a Web page. It also offers some suggestions for improving search engine placement.
- **Table Analysis:** This feature tests the table structure on the page. It specifically looks for unclosed <TR>, <TH> and <TD> tags inside a properly defined table (ie: one which has both an open and close <table> tag). It also reports on <TR>, <TH> and <TD> tags that appear outside of any properly defined table, since these may cause formatting errors on some browsers.
- **Form Structure:** For those sites which employ forms, this tool can check input types and variable names. The overall structure of Forms is examined (making sure there aren't form commands outside of any <Form>...</Form>). This test also checks to see if you've accidentally re-used form field names, have any empty <select> fields, etc.

- **Format HTML:** This module will take the HTML file and format it into an easily readable form. Table elements and rows are indented, etc. This formatting simplifies making manual changes to the HTML code.
- **Squish HTML:** This module will compress out unnecessary spaces and other data from the HTML page, making it as small (and thus as fast to download) as possible. The resulting HTML will be difficult to edit manually, but should not cause problems for Web browsers. "Squishing" is most effective on pages with lots of markup (e.g. lots of embedded tables), particularly if it was generated by a WYSIWYG HTML editor.
- **Frames Expansion:** For pages that have framesets defined, this offers a quick way to test the individual frame contents without having to type in the URLs. The test also lists all of the frames and their appropriate parameters as a quick reference. Finally, it offers a button that shows just the NOFRAMES content of the page, which is what the Doctor HTML report is actually testing.
- **Display Cookies:** Looking at the cookies sent by the Web server as the Web page is being loaded can be useful and instructive (especially since some users surf with cookies turned off). This tool shows all the cookies that arrive from both the page loading AND any that are sent with the images in the document.
- **Browser Support:** The browser test examines all HTML tags that occur on the Web page for compatibility with the most popular browsers and HTML standards.
- **Font Support:** The Font Support test checks all font faces specified on the Web page. These can come either from FONT tags or from cascading style sheets. This module looks to see if at least one of the fonts listed in each instance is available as a default font on Windows, Macintosh, and Unix computers. Making sure the font list includes options for each platform prevents the use of unexpected fonts on different computers and gives the designer greater control over the visual layout of the page.
- **Show HTML Hierarchy:** This test presents the HTML commands that are found in the document, with regular text removed. The source is indented to reflect inclusion in containers, which is helpful in hunting down extra commands in the code. This option is most useful when combined with one or more of the other structure tests. The outline is displayed in a scrolling <textarea>. If your browser is Javascript-enabled, then clicking on the button labeled "Show Printable Version" will display the outline in a separate window, which can then be printed for easy reference.

5. WebSat

WebSAT uses a set of heuristic rules to evaluate the use of tags in web page design. The rules are grouped into six categories as follows:

Accessibility

- Include "alternative text" for images, even when not used as links.
- Include "alternative text" for images used as links.
- Include "alternative text" when using java applets.
- Include text anchors or links for image maps.
- Include the "noframes" option when using frames.
- When specifying colors, use the RGB hexidecimal values. This helps to insure proper color recognition by any browser which does not recognized specified "color names".
- Combine the use of BG color and TEXT color attributes. Do not use them alone.

Form Use

- The form should include a functionality for returning the completed form.
- The form should include a button that is clearly labeled "Send" or "Submit" for delivering responses.
- The form should include a functionality for clearing or resetting the form.
- The form should include a button that is clearly labeled "Clear" or "Reset" for clearing current information from the form.

Performance

- The total amount of graphics should be 30K or less.
- Height and width specifications should be given for images.
- Images should be in either GIF or JPEG format.
- Banner sizes should be in the 468 X 60 pixels range.

Maintainability

- Relative links should be used when possible.
- Head tag information should be given.

Navigation

- Each page should contain at least one link.
- Links should be displayed in the default browser colors.
- Links should be descriptive of what the user will find upon visiting.
- Links should not be embedded.
- Links should not open a new browser window.

Readability

- Try to limit the density of the web page.
- Limit scrolling text, blinking text, and marquee style text on your web pages.
- Do not use horizontal lines across the web page.