Automatic Accessibility Evaluation of Dynamic Web Pages Generated Through XSLT

André Pimenta Freire
USP, University of Sao Paulo
P.O. Box 668
Sao Carlos-SP, Brazil
apfreire@grad.icmc.usp.br

Renata Pontin de Mattos Fortes
USP, University of Sao Paulo
P.O. Box 668
Sao Carlos-SP, Brazil
renata@icmc.usp.br

ABSTRACT

Much effort has been dedicated to develop software aids for authoring and evaluating Web pages using accessibility guidelines and standards. The evaluation of dynamic Web pages is a problem still unsolved in the field of automatic evaluation tools, since the current evaluators are only able to evaluate static Web pages. Stone and Dhiensa have addressed this problem, and proposed a method for evaluating the accessibility of dynamic Web pages using a generalized page which contains all possible outputs that can be generated by a script. In this paper, we present another approach for evaluating the accessibility of dynamic Web pages generated using XML and XSLT. The approach consists of analysing an XSLT using a structure descriptor such as DTD or XSD to determine the different types of XML documents that can be generated. The strength of the presented approach is given by the fact that the use of XML features enhances the results gathered from automatic accesibility evaluation of dynamic Web pages.

Categories and Subject Descriptors

H5.2 [**Information Interfaces and Presentation**]: User interfaces - screen design, graphical user interfaces (GUI), evaluation/methodology

General Terms

Design, Human Factors

Keywords

Web Accessibility Evaluation, XML, user interface

1. INTRODUCTION

The growth of the use of Internet using ubiquitous devices, and the use of the Web resources by people with disabilities have motivated the development of techniques to produce accessible Web pages. The main goal is that everyone can use and understand the content of the pages, despite disabilities or technologic restrictions.

Since Web Accessibility is the ability for a person using any user agent (software or hardware that retrieves and renders web content) to understand and fully interact with a website's content [9] [7], Web developers should often to be aware of how to make and maintain their web sites. However, it is not trivial to keep dynamic Web pages accessible.

Much effort has been spent to develop software aids for authoring and evaluating Web pages using accessibility guidelines and standards [4]. It includes authoring tools and automatic evaluation tools, among other techniques that help Web developers to produce accessible content.

Within the field of Accessibility Evaluation, current automatic evaluation tools are only able to validate and check the accessibility of static Web pages. The process of verification of dynamic Web pages has to be done more than once for each page, since there is more than one possibility of output, for each different input.

Stone and Dhiensa [8] have proposed a method for evaluating the accessibility of dynamic Web pages through the generation of a generalized page, which contains all possible outputs that can be generated by a script, and the evaluation is done over the generalized page.

The development of dynamic Web pages using XML (*eXtended Markup Language*) and XSLT *(XSL Transformations*) [11] has been increasing fast recently. We propose an approach for evaluating the accessibility of dynamic Web pages generated using XML and XSLT, by analysing the XSLT and using a structure descriptor such as DTD (*Document Type Definition*) or XSD (*XML Schema Definition*) to guide the possibilities of XML documents which can be generated.

This paper is organized as follows: In Section 2 the main concepts of Web Accessibility and Accessibility Evaluation are presented. In Section 3 discussions about automatic accessibility evaluation of dynamic Web pages are presented. In Section 4 we present our proposal for evaluating XSLT documents as a support for automatic evaluating the accessibility of dynamic Web pages using XML. In Section 5 the conclusions and future work are presented.

2. WEB ACCESSIBILITY AND ACCESSIBILITY EVALUATION

Software usability efforts have produced emerging standards addressing software accessibility for users with disabilities. Increasing recognition of the importance of software

W4A at WWW2005 10th May 2005, Chiba, Japan

accessibility and universal access to information technologies has been driven by a number of factors including: i) changing demographics and the "graying" of the population, ii) legislation in several countries like USA, Portugal and UK and iii) the rise of ubiquitous computing (e.g., Personal Digital Assistants (PDAs), Web-enabled cell phones, notebook computers) and the associated increasing diversity of people requiring access to information technology (IT) [6]. In Brazil, the legislation has started by government initiative last year and is under consolidation.

The idea behind Web Accessibility is based on more than just the implementation of standards; it embodies the idea that everyone has the right to be included in society, regardless disability, geographical location, language barriers, or any other factor. [9] [7].

Guidelines for accessible Web sites have been developed with the participation of representatives from many organizations around the world [4], and the most important of them, the WCAG 1.0 (*Web Content Accessibility Guidelines*) [10] has been published as a W3C recommendation. The guidelines include several topics to be used during the development of Web pages, such as providing alternative text for audio and visual content, providing semantic markup, etc.

The analysis of a Web page accessibility by means of guidelines, similarly to other inspection methods in usability/ accessibility assessment, requires observing, analysing and interpreting the Web page characteristics themselves [1].

Accessibility Evaluation of a Web site is more complex than simply validating markup since implementation of accessibility requirements is not only a matter of using valid markup, but also of issues such as the appropriateness of equivalent alternatives, and organization of informatino on a Web site [4].

Although automatic evaluation is not able to detect all accessibility problems in a Web site, it is still a powerful tool for helping developers to detect many problems related to markup. Various tools, such as Bobby [1] , LIFT [2] and WAVE [3] are able to verify the compliance of a Web site against the checkpoints of the WCAG 1.0. Brajnik [3] has published a work comparing some of these tools considering the aspects: *completeness*, *correctness* and *specificity*.

Automatic evaluation tools provide feedback to Web developers and maintainers, and many of them assist with the repair of the site [2]. One serious difficulty with these tools is about providing evaluation of Web pages dinamically generated server side. Some aspects of automatic evaluation of dynamic content and a proposal for solving this problem are discussed in Section 3.

3. AUTOMATIC EVALUATION OF DYNAMIC WEB PAGES

The general trend in web sites is for more and more pages to contain server-side scripting elements which are used to make the pages 'dynamic'. At present, when an automated accessibility checker checks the output from a scripted page it is in fact only checking one possible output from the script. Even if this page is identified as accessible, the very next visit to the same page could potentially result in output which is not accessible [8].

[1]http://bobby.watchfire.com
[2]http://www.usablenet.net
[3]http://wave.webaim.org

```
<html><body>
<?php
if(is_admin_user())
   echo "<p><img src='a.jpg'.../></p>";
else if (is_in_office_hour())
   echo "<p><img src='b.jpg'.../></p>";
else
   echo "<p><img src='c.jpg'.../></p>";
?>
</body></html>
```

Figure 1: Example of PHP script [8]

```
<html><body>
<choices>
   <choice><p><img src='a.jpg'.../></p>
   <choice><p><img src='b.jpg'.../></p>
   <choice><p><img src='c.jpg'.../></p>
</choices>
</body></html>
```

Figure 2: Example of a generalized code generated from the PHP code [8]

Current automatic evaluation tools are not able to verify the accessibility of dynamically generated web sites yet. Therefore, the evaluation of a single dynamic Web page has to be done by generating every possible different output (in HTML).

Stone and Dhiensa [8] have proposed a method to evaluate the validity and accessibility of dynamic Web pages generated by server side scripts, such as PHP, ASP and JSP.

The method proposed by Stone and Dhiensa [8] consists of obtaining a generalized output from a script which contains all possibilities and go on to show that a validity or accessibility tester can be relatively easily extended to check the extended output.

In Figure 1 an example of a piece of PHP script is presented and in Figure 2 the correspondent generalized code is shown. The variant parts of the code were codified as *choice* tags.

According to Stone and Dhiensa [8] this method addresses the problem of validation of dynamic web pages, since it is able to check every possible output that could be generated by dynamic Web pages generated by scripts. However, the method was not conceived for attend XML based Web applications.

In Section 4, our approach for evaluating dynamic Web pages generated through XSLT is presented.

4. AUTOMATIC EVALUATION OF XSLT

The use of XML (*eXtensible Markup Language*) and XSLT [11] for creating Web sites has been increasing recently, as developers have noticed the advantages in separating content and layout during the development and for providing multiple interfaces for the content.

The content production in Web applications that use these technologies works as follows: at each requisition one XML document is generated, and it is transformed using an XSLT by the browser at the client, or by the server application. The transformation by the XSLT occurs by matching certain

```
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
   <body>
   <p>
   <img src="img.png">
     <xsl:if test="/document/level
       = 'accessible'">
      <xsl:attribute name="alt">
        <xsl:value-of
            select="/document/alternative"/>
       </xsl:attribute>
     </xsl:if>
   </img>
   </p>
   </body>
</html>
</xsl:template>

<xsl:template match="description">
   <xsl:apply-templates>
</xsl:template>
</xsl:stylesheet>
```

Figure 3: XSLT of the First Example

XML elements and combining them with the content given in XSLT, generating an output.

Thus, the accesibility evaluation of this kind of application consists of a particular case of evaluating dynamic Web pages, that can use the features of XML do help the evaluation process.

The evaluation of XSLT can detect several accessility problems in the pages that will be generated using it, since the XSLT contains all information about the layout of the HTML.

XSLT Evaluation is supposed to perform the analysis of control structures, such as *if* and *for-each*, as the XSLT is able to handle these elements. To execute such analysis, it is necessary to use testing methods similar to conventional software testing methods, such as white-box testing, where the structure and flow of data under test are visible to the tester [5].

However, the input of a XSLT transformation are XML documents with a previously known structure defined by a DTD or by a XSD (*XML Schema Definition*). Therefore, the test of accessibility of XSLT containing control structures is facilitated by the use of a DTD or XSD as guideline to be considered in tests to verify the possible outputs that can be generated by the XSLT, according with the different types of XML documents that can be used as input.

A possible test strategy using an XSLT would be to verify its output using the method proposed by Stone and Dhiensa [8], since the XSLT stores all information about the HTML to be generated, and may be parsed (possibly with some modifications) in the same way that a *generalized document* is parsed.

On the other hand, the use of the definition of the XML document that is expected to be read by the XSLT can provide an evaluation method with better results, since it makes possible to detect even more problems, using the hability of test methods guided by the structure of XML documents.

An example of a simple accessibility problem that can be detected using test techniques on an XSLT using the correspondent XSD is the evaluation of conditional *if* statements. In Figure 3 is presented a XSLT which is supposed to check whether the element *level* is set to *accessible*. If the level is set to *accessible*, an *img* element receives an alternative text attribute. In Figure 4 it is possible to notice that the element *level* is an integer, so that is is not possible that any valid XML generated contains an element *level* with the value *accessible*.

```
<?xml version="1.0"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:element name="document">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="level"
type="xs:int"/>
      <xs:element name="alternative"
type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

</xs:schema>
```

Figure 4: XML Schema Definition of the First Example

```
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
   <body>
   <p>
     <img src="img.png">
<xsl:attribute name="alt">
       <xsl:apply-templates
            select="document/description"/>
</xsl:attribute>
     </img>
   </p>
   </body>
</html>
</xsl:template>

<xsl:template match="description">
   <xsl:apply-templates>
</xsl:template>
</xsl:stylesheet>
```

Figure 5: XSLT of the Second Example

Another example of accessibility problem that can be detected by this kind of evaluation is the reference for elements that can not exist. In Figure 5 is presented a XSLT which is supposed to read an element description and use it as an alternative text for an image. In Figure 6 the XSD correspondent to the XML document expected to be transformed by the XSLT is shown, and it is possible to notice that the structure of this XML does not have any *description* element under the *document* element, so this XSLT will never generate a HTML with an alternative text for the image, and will not be classified as accessible in any evaluation.

```
<?xml version="1.0"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:element name="document">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="name"
type="xs:string"/>
      <xs:element name="address"
type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

</xs:schema>
```

Figure 6: XML Schema Definition of the Second Example

The automatic evaluation of XSLT using information from DTD or XSD can provide more information about accesibility problems than the evaluation of generalized output generated by scripts using the of the structure of XML document that will be input of our proposed approach enables the implementation of simple test procedures.

5. CONCLUSIONS AND FUTURE WORK

The growth of the importance of Web Accessibility has enhanced many research areas related to the development of assistive technologies, in order to enable the access to Web pages to everyone, despite disabilities and technologies restrictions.

Several methods for authoring and evaluating Web pages using accessibility guidelines and standards have been developed [4], like authoring tools and automatic evaluation tools, but problems still remain unsolved, like the automatic accessibility evaluation of dynamic Web pages.

We have presented an approach for evaluating the accessibility of dynamic Web pages generated in a particular way. We addressed the accessibility evaluation of Web pages generated using XML(*eXtended Markup Language*) and XSLT (*XSL Transformations*). The evaluation consists of analysing the XSLT and in using an XML structure descriptor such as DTD (*Document Type Definition*) or XSD (*XML Schema Definition*) to guide the evaluation of all possibilities of XML documents that can be generated by the application.

Then, the strength of the presented approach is given by the fact that the use of XML features enhances the results gathered from automatic accesibility evaluation of dynamic Web pages.

As future work, we intend to validate this approach by implementing an automatic tool that evaluates the accessibility of dynamic Web pages generated by XSLT, adapting traditional software testing strategies.

We also intend to work on the formal specification of automatic accessibility evaluation tools, using concepts of evaluation of structured hyperdocuments.

6. ACKNOWLEDGMENTS

Thanks to FAPESP for funding this work.

7. REFERENCES

[1] J. Abascal, M. Arrue, I. Fajardo, N. Garay, and J. Tomas. The use of guidelines to automatically verify web accessibility. *Universal Access in the Information Society*, 3(1):71–79, 2004.

[2] C. Boldyreff. Determination and evaluation of web accessibility. In *Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE). Proceedings. Eleventh IEEE International Workshop*, pages 35–40, 2002.

[3] G. Brajnik. Comparing accessibility evaluation tools: a method for tool effectiveness. *Univers. Access Inf. Soc.*, 3(3):252–263, 2004.

[4] J. Brewer. Web accessibility highlights and trends. In *W4A: Proceedings of the international cross-disciplinary workshop on Web accessibility*, pages 51–55. ACM Press, 2004.

[5] R. S. Pressman. *Software Engineering: a practitioner's approach*. McGraw Hill, 2001.

[6] P. S. Reed, D. Gardner-Bonneau, and S. Isensee. Software accessibility standards and guidelines: progress, current status, and future developments. *Universal Access in the Information Society*, 3(1):30–37, 2004.

[7] B. Sierkowski. Achieving web accessibility. In *SIGUCCS '02: Proceedings of the 30th annual ACM SIGUCCS conference on User services*, pages 288–291. ACM Press, 2002.

[8] R. G. Stone and J. Dhiensa. Proving the validity and accessibility of dynamic web-pages. In *W4A: Proceedings of the international cross-disciplinary workshop on Web accessibility*, pages 45–49. ACM Press, 2004.

[9] J. Thatcher, P. Bohman, M. Burks, S. L. Henry, B. Regan, S. Swierenga, and M. Urban. *Constructing Accessible Web Sites*. Glasshaus, 2002.

[10] World Wide Web Consortium (W3C). Web Content Acessibility Guidelines 1.0, 1999. Available at http://www.w3.org/TR/WCAG10/.

[11] World Wide Web Consortium (W3C). XSL Transformations (XSLT), 1999. Available at http://www.w3c.org/TR/xslt.