Tree-Map Visualisation for Web Accessibility

John Bailey, Elizabeth Burd
Department of Computer Science
University Of Durham
Durham DH1 3LE
{j.o.bailey, liz.burd}@durham.ac.uk

Abstract

Ensuring a website is accessible for diverse sets of users is a crucial and in some circumstances a legal requirement. It can also be a time consuming and frustrating fire fighting exercise. This research paper describes a visual evaluation of the accessibility of a website using Tree-Maps. Its goal is to highlight areas of a site that need attention. This is achieved by generating accessibility metrics and then plotting these onto a visualisation. The research identified that Tree-Maps can efficiently represent the accessibility of a website. Further work on the visualisation of changes in accessibility is suggested and discussed.

1. Introduction

The web is changing the way society interacts with information[13]. Today many companies use the web as a means of interacting with their clients and customers. For these clients to be able to access the information, particularly those with disabilities, the interface to this information is crucial. This paper describes a tool for the assessment of web accessibility and investigates how it may be used to aid the management of websites. The tool is used to produce a visual representation of a website which can then indicate the accessibility of the site's pages. This paper now describes web accessibility and how it can be evaluated. It then goes on to review Tree-Maps and how they can be applied to a website's accessibility through the use of maintenance scenarios. The effectiveness of this visualisation is then evaluated and finally ideas for future work presented.

2. Web Accessibility

Accessibility is defined by Webster's dictionary as being "capable of being used or seen". In terms of the World Wide Web this means that an accessible website is one which can be seen and used by as many different user groups as possible. Boldyreff underlines the growing importance of the Web but warns that, "access to the web and its many applications cannot be taken for granted"[3]. The World Wide Web Consortium (W3C) has produced a set of web accessibility guidelines[6]. Each guideline contains a prioritised list of checkpoints that web authors can use to ensure their web content is accessible to the widest possible audience. Recently passed legislation[8] ensuring the accessibility of on-line learning material within education has raised the profile of web accessibility within British academia. A similar law (SECTION 08) passed in the USA enforces accessibility for all software and information systems within federal organization[ ]. Ensuring full compliance with accessibility standards for a large website is extremely time consuming and expensive.

3. Web Maintenance

Maintaining a website involves not only updating content but also the presentation and structure. To facilitate this maintenance tools have been developed that semi-automate the accessibility evaluation process. Evaluation tools examine web pages for machine readable accessibility barriers and report back to the maintainer. Results from these tools can be used as an indicator for overall accessibility. For example, if a web page contains basic accessibility barriers it is inaccessible to users affected by those barriers.

Although there are now tools which evaluate individual pages, little work has been carried out in assessing accessibility for an entire website. This paper proposes that such an overview allows an organisation to monitor and react to potential accessibility problems. To provide this overview the paper will present a novel application of Tree-Maps to visualize a website's accessibility. This is described in the following section.

4. Tree-Map Visualisation

Tree-Maps are a visualisation developed to present hierarchical data in a space efficient manner[12]. They are two dimensional visualisations which display nested, coloured, rectangular nodes inside a fixed space. The dimensions and colour of the nodes convey details about the data they represent. This means they make maximum use of their available display space. A Tree-Map can represent website as a hierarchy by presenting each page as a node nested within either a physical (file system) or logical hierarchy (information structure). As links between pages are not the focus of this paper it was possible to represent a website using its file system hierarchy. Navigational issues are already considered within the W3C guidelines and by ensuring that valid HTML is used. There are a significant number of advantages of the use of Tree-Maps for visualising web pages, which are described in the following.

Effective Space Utilization

By presenting the data densely and in a fixed space there are no problems with either navigating through a large area or the disorientation often associated with graph based visualisations[4]. Users cannot get "lost" within the Tree-Map as they view the entire visualization on a single screen.

Interactivity

To assist the maintenance of a website the user must be able to interact with what they see on screen. For example, users should be able to zoom in and out of different sections and by doing so reveal an appropriate level of accessibility detail.

Comprehension

Users must be able to make sense of the visualisation quickly and easily. Johnson states that Tree-Maps "must facilitate the rapid extraction of information with low perceptual and cognitive loads"[4]. Tree-Maps encode information by changing the colour and dimensions of each node. Dimensions and colour are both preattentive graphical features[ ]. Preattentive features are those that picked up immediately by users without the need to analyse the visualisation. By providing an overview of a website it is possible to view changes on a larger scale, and hence intervene before problems become critical.

A visualisation is only as useful if the underlying data is accurate. Section describes how accessibility metrics were generated from a website and then presented as a Tree-Map.

5. Tree-Map Generation

In order to accurately represent the level of accessibility of a web page a software tool called Access Valet [1] was used to evaluate each web page against the W3C guidelines [6]. Access Valet generates an XML based report containing all instances of breeches of W3C guidelines. Along with each instance it gives a confidence rating which reflects how confident the tool is that a guideline has been breeched. For instance, if an image has no alternative text then the tool is certain that guideline 1.0 has been broken. An example where the tool cannot be certain would be whether clear and appropriate language had been used. .

5.1. Metrics chosen

Metrics were derived from the reports generated by Access Valet. Each instance of a W3C guideline being broken was recorded and used to produce an overall accessibility score. The number of invalid / deprecated HTML tags within the document was used to represent the validity of the HTML. This is a useful indicator of poor accessibility as accessibility relies on correct use of HTML.

5.2. Tool development

A combination of PHP and Java was used in the creation of the tool. There were four main stages:

1. Visiting each page following (spidering) links within the page for the entire website and then building an XML document containing the hierarchical structure.
2. Running evaluation tool on each page and processing the report
3. Generating an accessibility metric and adding them to the XML document from Stage 1
4. Converting the XML document in Stage 1 to the format required by the Tree-Map software

Further details of this process will now be given.

Stage 1

Pages were spidered by parsing and following all available hyperlinks this process started with the homepage and continued ignored external website links. Only HTML based pages were included. This included dynamic pages providing they produced their output as HTML.

Stage 2

For each individual webpage an accessibility report was produced by Access Valet. This report is produced in the W3C Evaluation and Report Language (EaRL)[7]. It reports all valid and invalid HTML used within the page. Following this it includes all failed accessibility guideline tests and where exactly within a page's source the barrier occurs. Access Valet also concludes whether the page complies with the specified guidelines.

Stage 3

Two accessibility metrics were generated from the report based on the problems detected by Access Valet. There are two simple equations. *Equation I* takes the potential accessibility problems detected by Access Valet and computes an overall accessibility metric. *Equation II* collects all the invalid or deprecated HTML errors found by Access Valet and produces a metric for the validity of the HTML used. The details of these equations follow.

Equation I
At each confidence level (contained in the Conf set) the number of accessibility problems (B_c) found is multiplied by the simple weight (W_c) for that confidence. The higher the confidence rating the greater the weight. Weights were chosen after a series of preliminary tuning experiments. This is then divided by the total number of HTML attributes (Attribs) and elements (Elements) contained within the webpage. This ensures that pages of various sizes can be fairly compared. The sum of each confidence score is the Overall Accessibility metric (OAM).

$$OAM = \sum_{\forall c \in Conf} \left(\frac{B_c.W_c}{Attribs + Elements}\right)$$

$$Conf = \{(Certain, 10), (High, 8), (Low, 4), (None, 1)\}$$

Equation II
There are four elements to Equation II. Invalid and deprecated HTML tags (IEM and DEM) and along with invalid and deprecated attributes (IAM and DAM) within HTML tags. These give an Overall HTML Element and Attributes Metric (OEAM). As with equation I, weightings add more significance to elements than attributes and reflect that invalid tags or attributes are more serious than deprecated tags or attributes. Deprecated markup was considered less problematic than invalid markup. This is because at the time of creation the markup might have been valid and so it is reasonable to expect modern browsers will be able to cope gracefully.

$$IEM = \frac{InvalidElements.10}{TotalElements}$$

$$DEM = \frac{DeprecatedElements.5}{TotalElements}$$

$$IAM = \frac{InvalidAttribs.4}{TotalElements}$$

$$DAM = \frac{DeprecatedAttribs.2}{TotalElements}$$

$$OEAM = IEM + DEM + IAM + DAM$$

Stage 4
To create the Tree-Map an open source application called JTreeMap was used. The XML document produced in Stage 1 combined with the added accessibility information was converted into the XML format required by JTreeMap. JTreeMap then parsed the XML file and created the visualisations that are presented in the Tree-Maps and Discussion section.

5.3. Interpreting the Tree-Maps

Each node in the Tree-Map represents a web page. Each node's colour saturation was determined by equation I and its dimensions by equation II. Equation I identifies how accessible a page is according to the W3C standards. An increase in the accessibility metric relates to an increase in the number of accessibility barriers. The ordinal nature of the metric makes it well suited to a colour saturation mapping. Equation II denotes the quality of the HTML. Nodes are grouped together in "branches" which correspond to the directories their page is located in. Borders and labels of each branch are shown on the top level of the tree.

The more saturated (or brighter) a node is, the less accessible its corresponding web page. The larger the node, in relation to others in its branch, the more inaccessible mark-up contained within its corresponding web page. So then if follows that a large bright node is probably more critical than a small dark node.

6. Web Maintenance Scenarios

To investigate whether Tree-Maps provide an effective overview of changes in a website's accessibility four scenarios were created based on typical website maintenance activities. The website of the Department of Computer Science at the University of Durham was chosen because source code and maintainers of this website were readily available. Also the site's size (approx 200 pages) was also ideal for the experimental nature of the software tool used.

6.1. Scenario One: Script Maintenance

Websites rarely remain unchanged during their life cycle and regularly undergo acts of maintenance[14]. Typically such maintenance will be small corrective changes and could be carried out to either dynamic web applications or static web pages[2]. This scenario simulated an act of bad maintenance on a dynamic web page. Specifically the staff details page (get_info2.php) will be modified. The script was changed so it no longer produced ALT attributes inside image tags in the HTML.

6.2. Scenario Two: Addition of a new module

Websites constantly evolve and grow[10]. To cope with a constant demand for extra features and functionality, web maintainers usually make use of preexisting software modules to meet the new requirements. Accessible design in the pages produced by third party products is just as important as pages created by the organization itself. This scenario simulates the act of installing such a software module. The module was an open source e-shop (phpShop) product and contains dynamic pages.

6.3. Scenario Three: Accessibility Repair work

When accessibility problems are found, website maintainers may choose may use an accessibility repair tool such as A-PROMPT[1]. Scenario three investigates whether the visualisation accurately represents any improvement after the pages contained within the undergraduate prospectus were repaired.

6.4. Scenario Four: Convert and publish documents from commercial software

Internal documents are rarely created in HTML. Organisations may wish to convert documents from their original format into HTML for publication on the Internet. Documents containing details of taught postgraduate courses were converted from Word format to HTML web pages. In this scenario all conversions will be carried out using the software which created the original document.

7. Tree-Maps and Discussion

Original Website Tree-Map

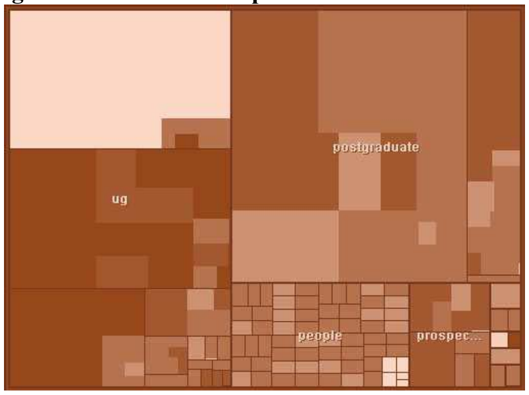

Figure 1. Tree-Map of Computer Science website before maintenance

Figure 1 shows the Tree-Map of the Department's website before any modifications have been made. There are two very interesting areas that appear to need attention. Firstly in the top half of the "ug" (undergraduate section) there is a large "bright" section. In this section there are three very large HTML files that have been converted from word processor and desktop publishing formats. The second area that attracts the eye is in the bottom right hand corner of "people". Upon closer inspection they appear to be clones (copies). Each script lists the staff members in each of the sub-divisions of the department. The developer has copied and pasted a single script and embedded a sub division variable within each copy. The remaining nodes in the "people" section are generated from "get_info2.php" and are staff information pages. They appear as one of two shades, brighter nodes represent pages that have a photograph on them. This is because Access Valet identifies *all* images as at least a low confidence potential accessibility problem (i.e. any image may flicker).

Scenario One Tree-Map

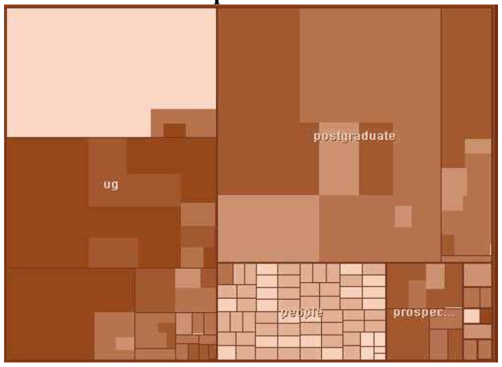

Figure 2. Tree-Map of Computer Science website after staff page maintenance

It is clear from Figure 2 that the accessibility metrics of the pages in the "people" section have changed. The shades of the nodes in this section are much brighter than in Figure 1. This indicates that the accessibility of these pages has decreased. As all images within the page had the "alt" attribute removed even pages without staff photographs were affected.

Scenario Two Tree-Map

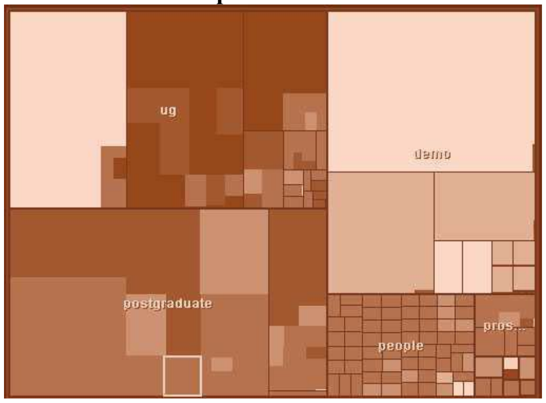

Figure 3. Tree-Map of Computer Science website after addition of phpShop (demo) module

Scenario Two has produced a Tree-Map significantly different from that in Figure 1. The phpShop module created many web pages and so added a large number of nodes to the Tree-Map. Because Tree-Maps are constrained within a fixed space the other nodes' dimensions have been altered to fit onto the screen. The section labeled "demo" contains the additional nodes belonging to phpShop. According to the Tree-Map it is clear that the pages generated by phpShop have accessibility problems. There is a significant mass of bright and large nodes within the "demo" section of the Tree-Map.

Scenario Three Tree-Map

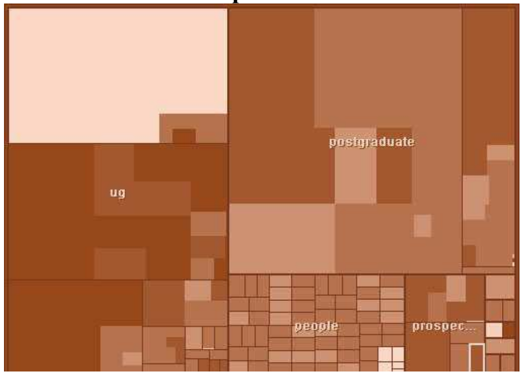

Figure 4. Tree-Map of Computer Science website after accessibility repairs to prospectus section

In Scenario Three there appears to be no difference in the accessibility of the department's website. Figure 4 is identical to Figure 1 and hence the "repairs" carried out on the web pages had no noticeable effect according to the Tree-Map. This could have been expected as the website was recently updated with particular attention paid to the prospectus and accessibility.

Scenario Four Tree-Map

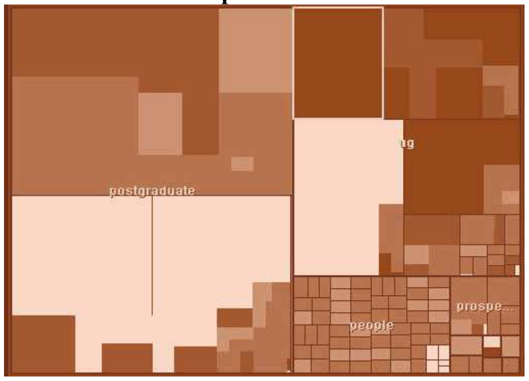

Figure 5. Tree-Map of Computer Science website after word processor documents have been converted to HTML in the postgraduate section

Figure shows the effect of adding documents to a website that have been converted from a bespoke format into HTML without any treatment. The large areas of bright nodes in the postgraduate section correspond to the newly converted web pages and their ancillary pages. For example the slide show software created a new sub-directory and one web page per slide. In this case it also created a frameset to allow navigation. Frames are known to be potential barriers to accessibility.

. Evaluation

.1. Detection of Accessibility Errors

The Tree-Map visualisation illustrates and draws attention to accessibility problems. Comparing Figure 1 to Figure 2 it is immediately obvious that there is an accessibility problem present in the people section of the website after the modification to the script. The use of preattentive features such as colour saturation and dimension ensures that time is saved when looking for accessibility problems. In this instance the hue of the nodes remained constant with only the saturation was changed. This allowed a mapping from the ordinal accessibility metric. As such the eye is drawn to the most inaccessible web pages. In the case of the departmental website these were web pages converted from word processor documents. All the figures show that examples of these web pages are located in the undergraduate ("ug") section. It is also very clear that in Scenario Four (see Figure) inaccessible web pages have been added to the postgraduate section.

.2. Limitations of Tool / Visualisation

One important limitation to the use of a visualisation to manage accessibility is that it is not accessible itself. In its current form the tool could not be used by a blind user. A text based equivalent therefore is proposed which would take the form of a nested tree, similar to the XML document processed by the visualisation tool. Voice based software readers could then navigate through the hierarchy with attributes read out loud to the user.

Another limitation is that it is impossible without comparing previous visualisations to gauge whether accessibility has improved or decreased. A solution would be to provide a comparative Tree-Map show the differences in accessibility between accessibility ratings taken over a period of time. For example, an organisation might wish to see if accessibility has been improved since initiating a new IT strategy.

.3. Potential Usage Example

The high level view of accessibility enables management to monitor the evolution of a website. If the same sections of the website are constantly performing badly, in terms of accessibility, the administration may wish to study the web practices and procedures and recommend new tools and techniques used to create the web content.

9. Related Work

Macias[11] has developed a toolkit KAI which analyses web pages, classifies and then presents them to a user in an appropriate form. Macias[11] introduces an XML based Blind Mark-up Language (BML) which "guarantees that the Web page constructed is accessible"[11]. The KAI aims to automatically enhance accessibility from the web browsers point of view it does not help organizations improve their web content creation techniques. Access Valet's evaluation tool was used in the creation of the Tree-Maps it also provides an entire reporting system. Reporting is text based and can be customised with the use of style sheets.

10. Conclusion

This paper has applied a Tree-Map visualisation to the management of accessibility within a website. Such an overview will allow managers to observe the current state of a website's accessibility as well as monitoring its evolution.
It is clear from the results of the scenarios that problems and changes in the accessibility can be quickly and easily observed. Visualisations are ideal for presentations and reports to non-technical audiences. The availability of this additional information allows an organisation to resolve problems before they propagate throughout the website. It also could be used to spot persistent offenders and identify if more training or better practices/tools are required. Potential staff exchanges can also be identified. For example, if one division produces accessible web pages they may be asked to help other divisions that may be struggling.

11. Future Work

Future work will be carried out to reduce the number of false positives highlighted by the tool. This would involve a pilot scheme with feedback from website maintainers used to tune the metrics used in Stage 3 of the tool's visualisation process. The switch from a snap shot based visualisation to one that shows accessibility changes over a period of time would be more useful for monitoring the system's development and determining future maintenance.

12. References

1. Adaptive Technology Resource Centre, U. o. T., *A-Prompt Web Accessibility Verifier Website*. 2004. http://aprompt.snow.utoronto.ca/
2. Athula Ginige, S. M., *Guest Editors' Introduction: Web Engineering - An Introduction.* IEEE Multimedia, 2001. (1): p. 14-18.
3. Boldyreff, C., *Determination and Evaluation of Web Accessibility.* Proceedings of IEEE 11th Intl. Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2002: p. 36 - 41.
4. Brian Johnson, B. S., *Tree-Maps: A Space-Filling Approach to the Visualization of Hierarchical Information Structures.* Proceedings of IEEE Visualization 1, 1 1: p. 284 - 2 1.

 . Christopher G. Healey, K. S. B., James T. Enns, *High-speed visual estimation using preattentive processing.* ACM Transactions on Computer-Human Interaction (TOCHI), 1 6. **3**(2): p. 107 - 13 .

6. Consortium, W. W. W., *Web Content Accessibility Guidelines 1.0*. 1 . http://www.w3.org/TR/WCAG10/
7. Consortium, W. W. W., *Evaluation and Report Language (EARL) 1.0*. 2002. http://www.w3.org/TR/EARL10/
8. Government, B., *Special Educational Needs and Disability Act 2001*. 2001. http://www.hmso.gov.uk/acts/acts2001/20010010.htm

 . Government, U., *Section 508 Website*. 2004. http://www.section 08.gov/

10. Lowe, D., *Engineering the Web - Web Engineering or Web Gardening* WebNet Journal, 1 . **1**(1).
11. Mercedes Macias, F. S., *Improving Web Accessibility for Visually Handicapped People Using KAI.* Proceedings of the 3rd International Workshop on Web Site Evolution, 2001: p. 4 - . http://csdl.computer.org/comp/proceedings/wse/2001/13 /00/13 004 abs.htm
12. Schneiderman, B., *Tree visualization with tree-maps: 2-d space-filling approach.* ACM Transactions on Graphics (TOG), 1 2. **11** (1): p. 2 - .
13. Sharp, J. W., *The Internet. Changing the Way Cancer Survivors Obtain Information.* Cancer Pract, 1 . **7**(8 September 01, 1): p. 266-26 .
14. Verbyla, J., *The Seven Habits of Effective Web Managers*, in *3rd Australian World Wide Web Conference*. 1 7: Lismore, Australia. http://ausweb.scu.edu.au/proceedings/verbyla/

1 . WebThing, *Accessibility Valet*. 2004. http://valet.webthing.com/access/