TOWARDS A WEB ACCESSIBILITY MONITOR

M.H. Snaprud*, C.S. Jensen**, N. Ulltveit-Moe*, J.P. Nytun*, M.E. Rafoshei-Klev*, A. Sawicka*, and Ø. Hanssen*
*Agder University College/Faculty of Engineering and Science
Grooseveien 36, N-4876, Grimstad, Norway, mikael.snaprud@hia.no
**Aalborg University/Department of Computer Science
Fredrik Bajers Vej 7E, DK-9220 Aalborg Øst, Denmark, csj@cs.auc.dk

A tool for the assessment and monitoring of web content accessibility is proposed. The experimental prototype utilises an Internet robot and stores the collected accessibility data in a data warehouse for further analysis. The evaluation is partly based on the Web Accessibility guidelines from W3C.

Keywords: **Web, WAI, accessibility, robot, data warehouse, mobility.**

Introduction

As the information society develops, the “digital divide” grows. On the other hand, appropriate uses of new technologies can bridge the divide and enable new user groups to access information and participate in communication. In particular, people with disabilities may benefit from the development.

We propose a tool to monitor the accessibility to content on the Internet. Using data warehouse techniques, we can discover how some accessibility properties depend on variables such as publishing tools, operating system, script language, geographical location, etc.

The Web Content Accessibility Guidelines (WCAG) [1] of the World Wide Web Consortium (W3C) have been adopted by eEurope2002 and are recommended for all public information in all member states. The guidelines, promoting multiple representation of content, are useful in many contexts. For example, a car driver “reading” electronic mail can make use of tools developed for persons who are blind. Benchmarking web pages can also be useful for ranking results from a search engine for a blind person.

Several automatic validators, such as the HTML validator from the W3C [1] or A-prompt [2], are already in use. However, the use of data warehouse technology in this context seems to be a new approach. An evaluation and report language (EARL) [1] is being developed by the W3C.

Design of an Accessibility robot

We are developing a robot, ROBACC, that collects accessibility properties from web sites.

As a first example property, we chose the “alt-tag” of images. The WCAG 1.0 contains 14 guidelines, the first of which concerns alternatives to images and audio.

Simple counting of all images with alt-tags is not useful. First, an existing alt-tag may not contain any useful information. Second, some images are used only as style elements and are generally ignored by text browsers. Therefore, we need a mechanism that automatically identifies informative alt-tags.

To achieve this, we suggest to match the alt-tag against a list of “useless” alt-tags maintained for each natural language. Examples of useless alt-tags include “click here,” “200kb,” and “image.gif.”

To separate images used only as style elements from those carrying information, we count images just a few pixels high or wide as style elements.

Additional properties are also stored in the data warehouse, including caching information, the content length (size in kb), and the content language. The accessibility information will be stored along with a timestamp, web-locations (URIs), the geographical location of the web server, descriptions of the software used by the web site, the ROBACC software version, categorisations of site owner organisations, etc. The database design is based on a star-schema as shown in Figure 1.

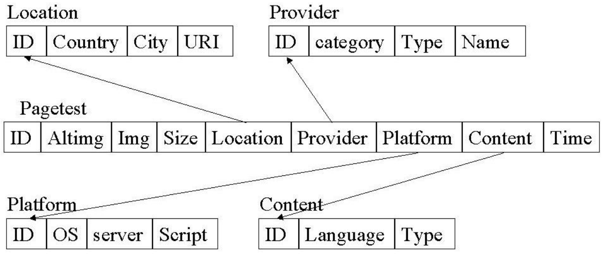
Location
ID | Country | City | URI
Provider
ID | category | Type | Name
Pagetest
ID | Altimg | Img | Size | Location | Provider | Platform | Content | Time
Platform
ID | OS | server | Script
Content
ID | Language | Type

Figure 1. Star schema for the ROBACC data warehouse.

The dimensions of the star schema describe Location (physical and virtual address of the site), Provider (properties of the organisation providing content), Platform (properties of software on the web site), and Content (properties of the web content). The strength of using a data warehouse will be more obvious as the amount of data increases. Still, the data extraction procedure used currently is useful already for smaller data sets. For more on data warehousing, see [3].

By storing all information in a data warehouse, we can compare web sites and perform in-depth analyses of the evolution of online accessibility over time. For example, it is possible to analyse the average accessibility of web sites based on a certain technical platform, or compare the web accessibility of the main universities in different countries.

Some information needs to be entered manually, including the following: The specific URIs to check, the name of the site provider, the site category (e.g., public, private, commercial, education), the site type (e.g., employment office, personal homepage, university), and the implementation language (e.g., Java (server/client), PHP, ASP, CGI).

ROBACC uses a Python script for fetching accessibility information from HTML and the HTTP header of a given URI. Data is extracted from the raw data and entered into the data warehouse regularly by SQL queries. The Python script fetches the URIs of sites to validate from a database, and it uses threads to simultaneously retrieve the information from the sites.

Results

Our initial experiments with a prototype of ROBACC confirm that the selected software platform is suitable for collecting and storing accessibility information from web sites. Currently, ROBACC does not follow links recursively.

We have selected 15 web sites that present different European countries as an example. We intended to locate the official page presenting each country (such as denmark.dk or spain.es). Since the naming policies differ from country to country, some countries are represented by pages about tourism, whereas others are represented by pages presenting government authorities. The principle underlying the survey method is unaffected by this.

In the ranking, the British site scored best on the alt image ratio. However, the same site still has some 180 HTML errors. Every fourth site does not declare any content length. None of the sites use the content-language tag to negotiate which natural language to return to the requesting browser. In the HTTP, several tags can be used to return content on a form requested by the user. However, the web servers must be set up to support them. As the work with Composite Capability/Preference Profiles (CC/PP) develops [1], this may be used more systematically. Adapting the content to special needs is e.g. dealt with in [4].

Discussion

The current prototype version of ROBACC assesses some web site properties. However, to effectively include new properties, as technology evolves, the architecture of ROBACC needs to be revised.

One problem with some methods for collecting accessibility data is that the metrics may time collection closely to the evaluation. This happens if we have a number of defined levels of accessibility for mobile devices, to assign to a web site, such as no access, poor access, sufficient access, and good access. Then the result of our data collection can hardly be used later for other purposes.

By storing instead raw data, we can assess accessibility from different perspectives and customise our analyses to special needs. For example, colour options can be crucial for a visually impaired person, whereas content length may be of interest to a mobile device. A weight for each property can define a user profile.

An improved tool will be based on software components. In this way, new accessibility properties can be plugged in without further code changes. A challenge here is to adapt the data warehouse to new accessibility properties while still making use of old measurements. As additional instances of ROBACC are installed, distributed queries, retrieving accessibility information from multiple databases, may be used.

The metrics for assessing web accessibility needs to be extended and refined. For example, a readability index could be useful as a numeric estimation for Guideline 14 of the WCAG.

A future version should include a graphical web interface to allow URI entry and support for online data analyses based on the data warehouse. Another option is to connect a spreadsheet to the data warehouse.

A further direction of development is to analyse the web site by means of techniques such as production rules, Bayesian networks, or pattern matching. The OCAWA tool [5] utilises a rule-based approach. Some previous work on integrating hypertext with artificial intelligence also suggests possible enhancements [6].

Conclusions

A proof of concept version of the ROBACC tool demonstrates a way to collect and store web accessibility data for further analysis. Future work will include more of the 14 guidelines from the WCAG and refinement of the metrics for assessing web-accessibility.

By collecting, analysing, and making available real information on accessibility, we may improve the awareness of accessibility and the understanding of critical issues in accessible web design.

Better use of already available technologies can dramatically improve the accessibility to content on the web both for people with special needs and for mobile devices. ROBACC can be used to monitor to what extent the potential is actually realised.

References

[1] WAI: http://www.w3.org/WAI

[2] A-prompt: http://www.aprompt.ca/

[3] T. B. Pedersen, C. S. Jensen: Multidimensional Database Technology. IEEE Computer 34(12): pp 40-46. 2001.

[4] V. L. Hanson, Web Access for Elderly Citizens, ACM Workshop on Universal Accessibility of Ubiquitous Computing, pp 14-18, 2001

[5] D. Chêne, M. Hoël, Web Site Accessibility Auditing Tool for Visually Deficient Persons OCAWA, pp. 27-32, ICCHP, 2002

[6] H. Kaindl, M. Snaprud, Hypertext and Structured Object Representation: A Unifying View, Proceedings of Hypertext '91, pp. 345-348. ACM, 1991.