Proceedings from the
Workshop on Web Accessibility and Metamodelling
April 14-16, 2005

Proceedings from the

Workshop on Web Accessibility and Metamodelling

April 14-16, 2005

Grimstad, Norway

Terje Gjøsæter and Mikael H. Snaprud (editors)

Summary

This publication presents the poster abstracts presented at the Workshop on Web Accessibility and Metamodelling, held at Agder University College, 14th-16th of April, 2005 in Grimstad Norway. The abstracts have been subject to review.

Skriftserien nr. 117

17 sider
kr 70,-
ISSN: 1503-5174
ISBN: 82-7117-550-5

Serviceboks 422, N-4604 Kristiansand

Design: Høgskolen i Agder

Keywords:

Web accessibility
Accessibility assessment
Metamodelling
Template identification
Web crawling

Contents

Preface

Terje Gjøsæter and Mikael Snaprud (editors)

This publication presents a collection of poster abstracts presented at the Workshop on Web Accessibility and Metamodelling. The workshop was held at the Agder University College in Grimstad, Norway on the 14th to the 16th of April 2005. The abstracts have been subject to review.

This workshop brought together experts in the field, as well as interested students. Participants from France, India, Italy, Germany, Denmark and Norway presented their work and development ideas. The slides from the talks are available on the webpage of the workshop:
http://osys.grm.hia.no/Workshop_on_Web_Accessibility_and_Metamodelling/

The organizers are most grateful for the generous support provided by The Norwegian Directorate of primary and secondary education, IT-Funk/The Research Council of Norway, and Agder University College, making this workshop an important starting point for further exciting co-operation.

Based on positive feedback from the participants, we plan to turn this into an annual workshop on web accessibility and metamodelling, and are looking forward to meeting the participants from this workshop as well as new participants in 2006.

How it started

The background for the workshop is an interesting story in itself. In the synergy field between the web accessibility monitoring project EIAO, and the SMILE metamodelling project, both international projects established at Agder University College, a third pilot project was started, attempting to apply metamodelling methods on web accessibility assessment.

This pilot project was called MEBACC, and a part of the proposed work for this pilot project was to host a small workshop discussing issues related to the project. As the pilot project progressed, the planned workshop grew to contain a peer reviewed poster

session, and it was decided to cooperate with the EIAO project to make this an even more interesting event than originally planned.

A program committee was established, consisting of Terje Gjøsæter of the MEBACC project and Mikael H. Snaprud from the EIAO project. Speakers were invited from Norway, further European countries, and India. A call for posters was published, and a group of referees was established to review the submitted poster abstracts. The following section of this publication is dedicated to the accepted poster abstracts.

The editors wish to thank Katharina Pätzold and Michael Eric Menk who participated in the organizing committee, the three anonymous referees, and all the participants who contributed to making the first Workshop on Web Accessibility and Metamodelling a successful event.

1 Adapting to feasible polling rate in an incremental web crawler using learning automata

Morten Goodwin Olsen
Faculty of Technology, Agder University College,
Grooseveien 36, N-4876 Grimstad, Norway
Email: mgolse99@student.hia.no

In this poster we will propose a model schema to detect the maximum possible number of updates on a fixed number of monitored web pages with limited available capacity. We present a model using connected fixed structure deterministic learning automata adapting to the polling frequencies most desirable for detecting as many updates as possible.

The proposed schema can be implemented as part of an incremental crawler monitoring any given number of web pages when it is feasible to poll and download only content which is out of date compared to the local repository.

The heart of the model involves connecting the polling frequencies of each monitored web page with a learning automaton. Each automaton increases the polling frequency when an update is detected and decreases the frequency when an assumed update did not occur. In this way the automata adapt the polling frequencies toward the frequencies of change of their corresponding web pages.

By connecting the automata to such a degree that they can only decrease the polling frequency whenever the maximal capacity is exceeded and only increase whenever the capacity is not exceeded, the schema will quickly adapt to using the exact available capacity.

We show, through experiments, that an incremental crawler with learning automata will utilize the available capacity much more than a traditional batch crawler. Our experiments show that the proposed schema will have up to twice as many polls toward modified pages than a batch crawler.

2 An Open architecture for large scale monitoring of web accessibility

Nils Ulltveit-Moe, Terje Gjøsæter and Mikael H. Snaprud
Faculty of Technology, Agder University College,
Grooseveien 36, N-4876 Grimstad, Norway
Email: nils@u-moe.no, Terje.Gjosater@hia.no and Mikael.Snaprud@hia.no

The EIAO (European Internet Accessibility Observatory) project will perform WCAG-based assessment of the accessibility of European web sites.

In the first release, we suggest an architecture with loosely coupled test suites that interfaces accessibility monitoring tools. The architecture is designed to incorporate existing and new accessibility monitoring tools into the observatory.

The system will consist of a controller, a repository database, a crawler, a data warehouse, a scheduler/dispatcher, a proxy, and one or more test suites.

We plan to use the HarvestMan crawler, Postgresql, a Squid proxy; and tools like HTML-Tidy and Mozilla may be used as basis for test suites. In this way, the architecture can benefit from available Open Source tools and also from propriatary test suites.

The controller starts the crawler, which loads the crawling specification from the repository, and starts crawling the web. Pages downloaded by the crawler are stored unmodified in the local repository together with metadata from the assessment.

After the crawling process has finished, the controller will start the main ROBACC scheduler/dispatcher, which distributes URLs to the test suites.

The scheduler/dispatcher dispatches a job and a URL to a test suite. The accessibility monitoring tool will perform a HTTP request towards the proxy, which fetches the given web page from the repository. The accessibility monitoring tool will then perform accessibility tests according to WCAG, and store the results as EARL metadata associated with the web page in the repository.

When all data from the test run is loaded into the data warehouse, the controller terminates.

With this architecture, we aim at creating a system that is both modular and scalable, and well suited for the task.

3 Distribution in large scale accessibility assessment systems

Ragnhild Varmedal
Department of computer science, Faculty of science
University of Tromsø, N-9037 Tromsø, Norway
Email: ragnhild@cs.uit.no

Accessibility assessment has traditionally been performed non-distributed and in small scale, either by the website owners to achieve better accessibility, or by a group or organisation to assess the accessibility of a selection of websites. These kind of assessments can easily be executed in a centralised way.

When the goal is to perform accessibility in large scale, the nature of the accessibility system changes. The web is itself a large distributed system, the websites that are subject to analysis are widely distributed, and the amount of data to collect and analyse is enormous.

Given this setting, the accessibility assessment system itself will gain from being distributed, and it might not even be possible to design such a system in a non-distributed way. Distributed systems provides challenges that the designers of non-distributed systems will not have to consider, but given a good design they can offer advantages that could not have been achieved without distribution.

One obvious advantage of a distributed accessibility assessment system is that the collecting and analysing of data can be executed close to the target websites. Another benefit can be distribution of workload over a set of nodes, and the possibility to scale the system to handle a even greater amount of work. With the proper design, the system can be robust against failure in parts of the system and against loss of data.

Of course it would be useful if the design allowed for a system that was not only scalable, but also extensible with new implementations of heterogeneous and autonomous measuring tools, possibly adding new and useful functionality addressing needs discovered after the design and implementation of the original tools

participating in the system. The key to this is a well defined interface for interaction between the tools and the distributed system.

4 A UML model of HTML for accessibility testing of web documents

Merete Skjelten Tveit, Terje Gjøsæter and Jan P. Nytun
Faculty of Technology, Agder University College,
Grooseveien 36, N-4876 Grimstad, Norway
Email: Merete.S.Tveit@hia.no, Terje.Gjosater@hia.no and Jan.P.Nytun@hia.no

Access to web content for all is crucial for building the information society. Information on the web should be accessible to all users, independent of disabilities or choice of web browser.

A pilot project has been established in cooperation with the Norwegian Directorate of Primary and Secondary Education. The aim of the project is to create a prototype for an Open Source tool for accessibility checking of web documents and web based teaching material.

An important part of the project is to define web document models representing the relevant standards for use in conformance testing. This poster will show how UML can be used for this purpose, a subset of the HTML 4.01 specification will be represented as a web document model in UML.

The idea is that the planned tool will take a set of web documents as input and instantiate them into web document model instances based on the web document model. If this instantiation is successful, the web documents are considered valid.

OCL-like constraints will be added to the web document model to model accessibility requirements. If there is a valid instance-of relationship between the model instance and the accessibility model, the tested documents are considered accessible.

The project is integrated with the ongoing research on metamodelling at Agder University College. The model elements and model instance elements will inhabit the two lowest levels of a metamodel architecture[1]. A basic representation defined in the mentioned research, which can be used in all levels of a metamodel architecture, will be considered for use in the project.

Accessibility requirements and web technology are constantly evolving. High level modelling of accessibility requirements can support more rapid generation of new test modules and improve the understanding of the accessibility barriers for web documents.

[1] J. P. Nytun, A. Prinz and A. Kunert: Representation of Levels and Instantiation in a Metamodelling Environment, NWUML 2004

5 Automatic template identification for accessibilty testing of web sites

Søren Øydna, Mikael H. Snaprud and Ole-Christoffer Granmo
Faculty of Technology, Agder University College,
Grooseveien 36, N-4876 Grimstad, Norway
Email: Soren.Oydna@hia.no, Mikael.Snaprud@hia.no and Ole.Granmo@hia.no

When performing accessibility testing of web sites, it is desirable to evaluate as few pages as possible without sacrificing the significance of the measurement. One way of accomplishing this is to identify the different templates a web site may use for web page generation. If we assume that pages based on the same template will have similar accessibility characteristics, only a subset of the pages from each template need to be used for testing a web site.
Also, assessing the template deficiencies will allow web masters to focus on adjustments that can improve larger portions of a web site.

The suggested approach for automatic template identification uses Bayesian networks. Web pages will be clustered by a Bayesian network based on their attributes and knowledge acquired on how these attributes relate to the different templates. Possible attributes includes tag structure, how CSS are used, link structure, colour usage and layout.

Initially standard values on how the attributes are related to the templates will be used. Since these values have not been adapted to the current templates, all attributes may have to be extracted to obtain correct clustering.
Each time a web page is added to a cluster the Bayesian network will gain knowledge on how the attributes are related to the templates. Because of this, fewer attributes need to be extracted as more knowledge is gained. Since each attribute extraction demands processing power, this will reduce computational load. With a traditional clustering technique all attributes would have had to be extracted for each page. The clustering algorithm itself also scales very well.

The poster will give an overview over the proposed operation of the automatic template identificator and indicate some results from initial experiments.