This item was submitted to [Loughborough's Research Repository](#) by the author.
Items in Figshare are protected by copyright, with all rights reserved, unless otherwise indicated.

Profile-based web document delivery

PLEASE CITE THE PUBLISHED VERSION

PUBLISHER

© ACM

LICENCE

CC BY-NC-ND 4.0

REPOSITORY RECORD

Stone, R.G., Jatinder Dhiensa, and Colin H.C. Machin. 2019. "Profile-based Web Document Delivery".
figshare. <https://hdl.handle.net/2134/2467>.

This item was submitted to Loughborough's Institutional Repository by the author and is made available under the following Creative Commons Licence conditions.

For the full text of this licence, please go to:
<http://creativecommons.org/licenses/by-nc-nd/2.5/>

Profile-Based Web Document Delivery

Roger Stone
Loughborough University
Loughborough
LE11 3TU, England
+44(0)1509 222686

R.G.Stone@lboro.ac.uk

Jatinder Dhiensa
Loughborough University
Loughborough
LE11 3TU, England
+44(0)1509 222681

J.Dhiensa@lboro.ac.uk

Colin Machin
Loughborough University
Loughborough
LE11 3TU, England
+44(0)1509 222683

C.H.C.Machin@lboro.ac.uk

Stone, R.G., Dhiensa, J. and Machin, C.H.C., "Profile-based Web Document Delivery", Proceedings of the 2006 ACM Symposium on Document Engineering, Brailsford, D.F. (ed.), ACM Press, Amsterdam, 2006, pp. 215-217, ISBN: 1-59593-515-0

ABSTRACT

This work originated by considering the needs of visually impaired users but may have wider application. A profile captures some key descriptors or preferences of a user and their browsing device. Individual users may maintain any number of profiles which they can edit for use in different situations, for different tasks or with different devices. A profile is described in terms of essentiality and proficiency. Essentiality is used to control the quantity of information that is transmitted and proficiency is used to control the format. Various levels of essentiality are introduced into a document by the technique known as microformatting. Proficiency (for the visually impaired) includes a description of minimum acceptable font size, preferred font face and preferred text and background colours. A key feature of the proficiency profile is the accessibility component which captures the user's tolerance of accessibility issues in a document, for example the presence of images or the markup of tables. The document delivery tool works as a kind of [filter](#) to reduce the content to the level of essentiality requested, to make the various presentation changes and to warn of accessibility issues as specified in the user's profile. Encouraging preliminary results have been obtained from testing the prototype with subjects from the local RNIB college.

Categories and Subject Descriptors

H.5.2 Information Interfaces and Presentation: User Interfaces

H.5.4 Hypertext/Hypermedia, User issues

I.7.4 Electronic Publishing

K.4.2 Computers and Society: Social Issues - Assistive technologies for persons with disabilities

General Terms

Human Factors

Keywords

Web-documents, Visual Impairment, Usability, Web Accessibility.

1. INTRODUCTION

There are two apparently opposing views of disability. According to one view everyone should be treated the same and according to the other view each individual is different, has different needs and these needs should be met on an individual basis. The web accessibility guidelines (WCAG) [5] are designed to enable all web pages to be made accessible to all and to this end some web authors use automated checking tools to ensure that they have followed the guidelines [e.g. 1, 7, 14]. However, many (most?) of the web pages on the internet have not been checked against these guidelines at all. When an individual user surfs the web they will not normally know in advance whether or not a page that they land on is accessible. On the relatively few web pages that carry a logo of accessibility, the actual logo is usually placed at the foot of a page and is not normally seen until after the page has been browsed. So it would seem that the accessibility information is absent or arriving too late. Now imagine a browsing device which runs an accessibility check on every page before delivery to the user. This would allow the user to be pre-warned of any accessibility issues but could make the browsing experience very slow if the checks take a noticeable amount of time to run. Also the error reports from the accessibility scan will contain messages about the full range of checks that were meant to provide accessibility for users with all kinds of different disabilities. An individual user would be better served by an accessibility check tailored to their own disability. Given a profile of the user it should be possible to subject the document to a much reduced accessibility check, specific to the user, in an acceptably short time and thus warn the user of any accessibility issues specific to the user before presenting the document itself.

The idea of a user profile then is to enable the system to deliver a surfing experience that is as specifically tailored to the individual as possible. In general the profile is made up of parameters which can be set by the user to control various aspects of the browsing experience. So far we have discussed parameters to do with specific accessibility checks but there are many other possibilities.

Although we refer to our tool as a [filter](#), the name is really only appropriate to the essentiality component. [The tool could also be called a proxy server except that its aim is to alter the pages which it handles to the benefit of the user.](#) It shares the concept of personal profiles with adaptive hypertext systems but our system is not adaptive. In summary the tool is responsible for rendering a chosen document according to the profile of the user. The remainder of this document gives some more details about our approach to profiles, our concepts of essentiality and proficiency, our approach to accessibility and some implementation details.

There have been many attempts to address the issue of making web content more accessible. The Web Adaptation Technology is a prime example that offers users with varying disabilities the ability to customise the visual interface to meet their needs [6]. This offers the user the opportunity to change the visual content 'on the fly' but does not introduce the idea of profiles. Other work recognises the usefulness of stored profiles. The W3C Device Independence Working Group have recommended an RDF-based structure and vocabulary for profiles which they call Composite Capability/Preference Profiles (CC/PP) [9] although our simple prototype has no need of such a sophisticated system.

2. PROFILES

At its simplest a profile is just a list of requirements or preferences. Any kind of requirement or preference could be considered, but in respect of web documents a stylesheet containing necessary or preferred styles is typical c.f. the web accessibility toolbar [11]. The profile can conveniently be stored in a database table linked to the user. The user should be able to create a profile from scratch or by choosing from a range of typical profiles. The user should have the ability to edit their own profile. A user should be able to have several profiles as appropriate to different tasks (e.g. directed searching, general browsing) or different physical states (e.g. alert, tired).

On starting a session the user would begin at the [filter](#) homepage and choose a profile (or create a new one). Then they would choose their first URL to browse. The page, as retrieved by the server, would be altered by the [filter](#) by leaving out non-essential information and by swapping to preferred font sizes, text colours, etc. before being received and rendered by the browser.

The [filter](#) homepage is situated on an authenticating site so the server will have ready access to the details of the user and thus be able to lookup their profile(s). When the user roams across the internet the name (or username) information will not normally be available. For this reason the [filter](#) has been developed so as to rewrite hyperlinks in documents so that they carry the required information. A hyperlink of the form

```
<a href='domain/page'>
```

is changed to the form

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Conference'04, Month 1–2, 2004, City, State, Country.
Copyright 2004 ACM 1-58113-000-0/00/0004...\$5.00.

```
<a href='filter?url=domain/page&usr=u27&prof=p3'>
```

By this means, after the user starts their browsing at the [filter's](#) homepage, they can then apparently roam freely. However each click on a link is routed through the [filter](#) and each new page is rendered according to the user profile.

3. ESSSENTIALITY

It is a common observation that many popular web pages are crammed with information. Their usability is in question because they are cluttered and have complex page structures [8]. Typically some central information is surrounded by navigation information and advertising, often in a multicolumn format. When a user's ability to receive large volumes of information is hampered (e.g. by having to listen to it being read by a screen reader) it is important to be able to get to the essential information quickly. It is possible to add markup to a web page to indicate levels of essentiality for sections of information. Rather than inventing new tags e.g. `<essen level='10'>...</essen>` and by their introduction making an HTML document non-standard, the same effect can be achieved without disturbing the document by the technique called microformatting [10]. Microformatting in this sense is a way of adding extra semantic information into an HTML document without compromising standards compliance or the accessibility of the document.

In our trial implementation, ten levels of essentiality were distinguished (1-10). Level 1 includes everything and level 10 identifies only the most essential information. A section of information would be given a degree of essentiality 5 by adding the attribute `class='ess5'` to the enclosing tag. Where the section of information is not already enclosed in a tag, or the tag is there but already had a class attribute, an extra span or div tag is added as appropriate.

Any particular profile will contain an essentiality preference which will be a number in the range 1-10. All information in a page with lower essentiality will be excluded. This is currently achieved by stylesheets using the `display:none` facility. Although the [filter](#) could physically remove content so that it is not available to the browser, it is useful to have the browser merely conceal content via the `display:none` facility. This allows a facility to be built in to the foot of every page using Javascript whereby the essentiality can be instantly switched to any of the levels via 10 links. Thus it is possible to override the profile essentiality setting and instantly view the page at any level of essentiality with a client-side operation.

4. PROFICIENCY

Proficiency is a combination measure of the (dis)ability of the user and the device that is being utilised to browse the web. For example a braille device might be unable to cope with images or the user might not be able to distinguish text below a certain font size. In the trial implementation eight style components were captured in the proficiency part of the profile (an image switch, preferred font face and size, colours of text and background, colours of link, hover and visited text). When editing their proficiency profile the user is given full freedom to [change](#) these settings using an HTML form. When choosing a font, a choice can be made from a menu of common font families but any font available on the user's system may be [entered](#). When choosing a

colour, a choice can be made from a fixed palette of forty named colours or a colour can be entered in #RRGGBB format.

5. ACCESSIBILITY

One of the new features of the draft version of version two of the Web Content Accessibility Guidelines (WCAG2) [3] is that for each accessibility check a list is given of specific disabilities or disability groups that should benefit if a document passes that particular accessibility check. In principle this means that a user with a specific disability would be most interested in just those accessibility checks which relate to their disability. This leads to the idea of including specific accessibility checks in the user's profile. When creating an accessibility profile a user would first be able to choose a prepared profile based on a generic disability (like visual impairment) and then edit the profile by adding or removing specific checks as they saw fit. Thus if the accessibility checks in regard to tables (caption, titles, headers, etc) are in the generic disability profile that the user chooses initially but the user knows (or discovers) that they can cope with tables that do not pass the accessibility checks, the user might remove such checks from their profile(s). In practice not all the accessibility issues defined in WCAG (or WCAG2) are amenable to machine checking, so the profile will just be based on those capable of machine checking.

6. IMPLEMENTATION

An essentiality and proficiency [filter](#) has been created that works with user profiles which are stored in a database table. The system has been implemented on an authenticating intranet so that the server has knowledge of the user and can thus access their profile(s). The essentiality and proficiency parts of the [filter](#) work by rewriting the page to introduce the styles requested in the profile. The accessibility part of the [filter](#) runs an accessibility check based on the profile and, if accessibility issues are found, a report is presented to the user before the web page concerned is rendered. The [filter](#) implementation language is PHP with a MySQL database for the profiles. **For the prototype, each profile occupies one table record with each feature of the profile occupying one field.** The page rewriting uses the regular expression pattern matching facilities built in to PHP.

The advent of XML-aware programming languages in various paradigms (Query languages, Functional languages and Procedural languages), has made the coding of searching for patterns in XML 'tree' structures very much more direct and researchers are beginning to report the building of accessibility checkers as DIY projects [4, 12]. Of course not all web HTML documents are XML documents (XHTML) but the trend is in that direction and also there exist applications to convert HTML documents to XML (e.g. JTiny [13]). The accessibility check built into the [filter](#) happens to be written in the functional language CDuce [2] but could easily have been written in (say) XSLT [15].

User trials have been conducted with nine volunteer subjects, with seven from the local RNIB college (Royal National Institute for the Blind). The experiment comprised the set-up of a personal profile, a short session of general surfing and finally a search for some pre-specified information in two websites which had been marked up **by ourselves** for essentiality. The websites were copies of real websites for a local tourist attraction and a **national**

building society. The experiment occupied about one hour per subject.

The trial showed up some awkward accessibility issues with the profile editor [16]. It was built so that if the user chose a certain setting (perhaps a text colour) then as soon as the selection was made the editor page was refreshed to show the effect. This seemed correct and necessary behaviour for sighted and partially-sighted users but was really unwelcome behaviour for those dependent on a screen reader as they had to re-navigate to the point they had just reached before they could continue.

Once the profile had been set up the users were all able to complete the tasks and at the end seemed to be enthusiastic about the [filter](#) and what it could do for them.

7. CONCLUSIONS

The idea has been presented that a document browsing system should render documents according to a user profile. This can be seen as an extension of the idea of the user working with browser preferences but this method is independent of the browser being used. Although a variety of different features figure in the profiles we have created, it may be that there are other issues that could be dealt with via a profile and tackled within the same framework. Initial results from a modest user trial are encouraging. The proficiency and accessibility aspects of the [filter](#) will work with arbitrary pages and can be used freely to browse the internet. The essentiality part of the [filter](#) clearly only works on specially marked up pages.

8. REFERENCES

- [1] A-Prompt: Web Accessibility Verifier, <http://aprompt.snow.utoronto.ca/>
- [2] Benzaken, V., et.al., CDuce: An XML-Centric General-Purpose Language, Proc. ACM International Conference on Functional Programming, 2003, <http://www.cduce.org/>
- [3] Caldwell, B., et.al., WCAG 2.0, Web Content Accessibility Guidelines 2.0, Web Accessibility Initiative (WAI), WWW Consortium, <http://www.w3.org/TR/WCAG20/>, 2005
- [4] Centeno, V.L., et.al., Web Accessibility Evaluation Tools: a survey and some improvements, 1st Int'l Workshop on Automated Specification and Verification of Web Sites 2005, [\[http://www.sciencedirect.com/science/journal/15710661\]](http://www.sciencedirect.com/science/journal/15710661)
- [5] Chisholm, W., Vanderheiden, G., Web Content Accessibility Guidelines 1.0, Web Accessibility Initiative (WAI), World Wide Web Consortium, <http://www.w3.org/TR/WAI-WEBCONTENT/>, 1999
- [6] Hanson, V. L. (2004). The User Experience: Designs and Adaptations. International Cross Disciplinary Workshop on Web Accessibility, World Wide Web Conference (WWW2004), New York City, USA, ACM.
- [7] HiSoftware: "CynthiaSays" Portal, <http://www.contentquality.com/>
- [8] Hudson, W. (2004). Inclusive design: accessibility guidelines only part of the picture. Interactions. 11: 55-56.

- [9] Klyne, G., et.al., "Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0", <http://www.w3.org/TR/CCPP-struct-vocab/>
- [10] Meyer, E., "Microformats", <http://meyerweb.com/eric/thoughts/2005/06/21/microformatsorg/>
- [11] Rainger, P., User Preferences Toolbar, TechDis, http://www.techdis.ac.uk/index.php?p=1_20051905100544
- [12] Takata, Y., Nakamura, T. & Seki, H., Accessibility Verification of WWW Documents by an Automatic Guideline Verification Tool, Proceedings of the 37th Hawaii International Conference on System Sciences, 2004, <http://csdl2.computer.org/comp/proceedings/hicss/2004/2056/04/205640098a.pdf>
- [13] Tripp, A, Giustina, et.al., JTidy Home, <http://sourceforge.net/projects/jtidy>
- [14] WebXact, web page testing service, <http://webxact.watchfire.com/>
- [15] XSLT - XSL Transformation language, <http://www.w3.org/TR/xslt>
- [16] https://co-public.lboro.ac.uk/corgs/essen/profile_ed.html