# Extracting Content from Accessible Web Pages

Suhit Gupta
Dept. of Comp. Sci./450 CS Bldg
500 W. 120th Street
New York, NY 10027
001-212-939-7184
suhit@cs.columbia.edu

Gail Kaiser
Dept. of Comp. Sci./450 CS Bldg
500 W. 120th Street
New York, NY 10027
001-212-939-7081
kaiser@cs.columbia.edu

## ABSTRACT
Web pages often contain clutter (such as ads, unnecessary animations and extraneous links) around the body of an article, which distracts a user from actual content. This can be especially inconvenient for blind and visually impaired users. The W3C's Web Accessibility Initiative (WAI) has defined a set of guidelines to make web pages more compatible with tools built specifically for persons with disabilities. While this initiative has put forth an excellent set of principles, unfortunately many websites continue to be inaccessible as well as cluttered. In order to address the clutter problem, we have developed a framework that employs a host of heuristics in the form of tunable filters for the purpose of *content extraction*. Our hypothesis is that automatically filtering out selected elements from websites will leave the base content that users are interested in and, as a side-effect, render them more accessible. Although our heuristics are intuition-based, rather than derived from the W3C accessibility guidelines, we imagined however that they would have little impact on web pages that are fully compliant with the accessibility guidelines. We were wrong: some (technically) accessible web pages still include significant clutter. This paper discusses our content extraction framework and its application to accessible web pages.

## Categories and Subject Descriptors
I.7.4 [**Document and Text Processing**]: Electronic Publishing; H.3.5 [**Information Storage and Retrieval**]: Online Information Services – *Web-based Services*

## General Terms
Human Factors, Algorithms, Standardization.

## Keywords
DOM trees, content extraction, reformatting, HTML, context, accessibility, speech rendering.

## 1.INTRODUCTION
Computer users are spending more and more time on the Internet in today's world of online shopping and banking; meanwhile, web pages are getting more complex in design and content. However, while web technologies are constantly being improved, both in terms of "friendly" user interfaces and functionality, the gap between the usability of the web for persons with vs. without disabilities grows ever wider [1]. Many of these technologies were designed to better the web experience for sighted users, including script- and flash-driven animation, pop-ups, banners, and of course images. While some users may find these features effective, they may make the website less accessible to users with disabilities. It is for this reason that the W3C's Web Accessibility Initiative has created a set of guidelines [2] to assist web developers in creating sites that are accessible to all.

The advantages of universal web accessibility go beyond legal requirements and the usefulness of a website to persons with disabilities. Applying universal design concepts to a website makes it accessible to all site visitors, including those visiting with graphics turned off or with mobile devices. There are several common steps that can be taken to make one's site more accessible. For example, screen readers, in-car browsers and users connected through slow dial-up connections that turn off images rely on ALT text, in place of images; therefore, adding ALT tags to all images is a worthwhile step. Furthermore, while Javascript allows a web designer to create a wonderfully dynamic site, one's site should not be completely dependent on it being turned on. Tables should also be laid out with care as the cell layout within the table typically determines the order in which text is rendered by screen readers and handheld browsers. Another approach to building an accessible website is to use Cascading Style Sheets (CSS) to specify layout, separating design from content. But none of these techniques inherently eliminates clutter: Users with disabilities still have to read or listen through all the navigation menus and other non-content until they reach the desired content.


W4A at WWW2005, 10th May 2005, Chiba, Japan

We developed Crunch (http://www.psl.cs.columbia.edu/crunch) as a web proxy, usable with essentially all browsers, for the purpose of content extraction (or clutter reduction) as described further in the following section. We believed that preprocessing web pages with Crunch would make inaccessible web pages more accessible, but initially imagined that Crunch would have little to do for fully accessible web pages. However, when we tested our heuristic framework on sites that *are* compatible with the WAI accessibility guidelines, we found that content extraction remained a valuable tool. For example, Figures 1-3 show an accessible website (tested using [3] [4]) before and after Crunch. Notice that Crunch removes most of the header material, shown graphically in Figure 1 and with ALT text in Figure 2, and starts the main article much closer to the top of the page in Figure 3.


**Figure 1 – From NASA Research's WAI compliant website**

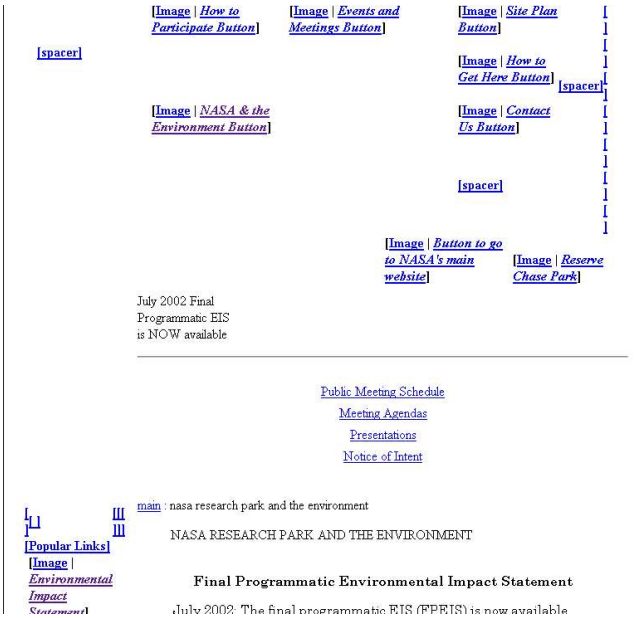
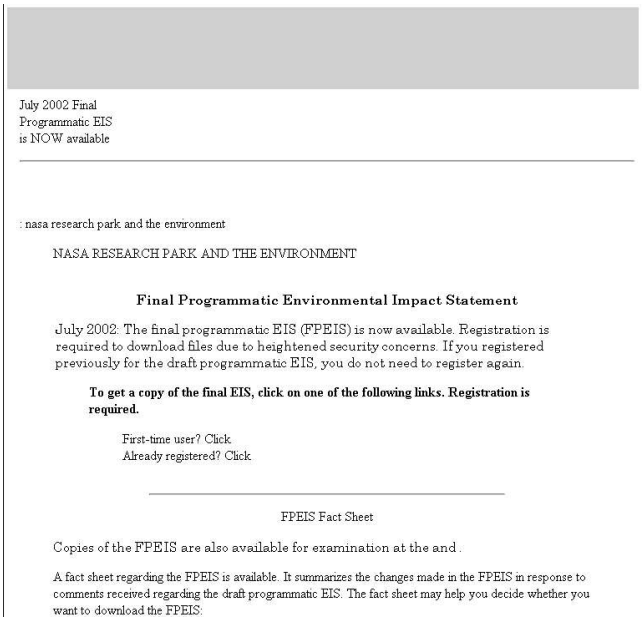



**Figure 2 – Same web page with ALT text replacing images**

## 2.CRUNCH

Crunch operates as a web proxy, sending the web browser's URL request to the appropriate web server, and then applying its heuristic filters to the retrieved web page before returning the content extracted from that web page to the browser (or other HTTP client). The first step in Crunch's analysis of a web page is to pass it through a conventional HTML parser, which corrects the markup and creates a Document Object Model (DOM) tree. DOM (http://www.w3.org/DOM) is a W3C standard for creating and manipulating in-memory representations of HTML (and XML) documents. Crunch's heuristics manipulate the DOM representation in terms of tree transformation and pruning operations, rather than working with HTML text. This enables Crunch to perform its analysis at multiple granularities walking up and down the tree, both on large logical units similar to Buyukkokten *et al.*'s "Semantic Textual Units" [5] and on smaller units such as specific links. Crunch generates the corresponding HTML (or optionally plain text) as its last step.



July 2002 Final
Programmatic EIS
is NOW available

: nasa research park and the environment

NASA RESEARCH PARK AND THE ENVIRONMENT

**Final Programmatic Environmental Impact Statement**

July 2002: The final programmatic EIS (FPEIS) is now available. Registration is required to download files due to heightened security concerns. If you registered previously for the draft programmatic EIS, you do not need to register again.

**To get a copy of the final EIS, click on one of the following links. Registration is required.**

First-time user? Click
Already registered? Click

FPEIS Fact Sheet

Copies of the FPEIS are also available for examination at the and .

A fact sheet regarding the FPEIS is available. It summarizes the changes made in the FPEIS in response to comments received regarding the draft programmatic EIS. The fact sheet may help you decide whether you want to download the FPEIS:

**Figure 3 – NASA web page through Crunch**

Our design of Crunch's heuristics was largely intuitive, inspired by related work, and based on our own informal perusal of a wide variety of clutter-laden websites. We did not attempt to model either author or user tasks, nor their corresponding context or intentions, but we believe any non-intrusive approach to doing so would also likely be heuristic and thus also imprecise: There is no "one size fits all" algorithm that could accurately extract the content desired by an arbitrary user from an arbitrary web page during an arbitrary visit to that web page. Crunch heuristics replace images with their ALT text, find clusters of links that might constitute navigational menus, table cells devoid of apparent content that might exist primarily for layout spacing purposes, etc. (see [6][7] for details). Another example is shown in Figures 4 and 5.

One of the limitations of our framework is that Crunch could potentially remove items from the web page that the user may be interested in, and may present content that the user is not particularly interested in. This problem is partially addressed by providing an administrative console, whereby an individual user can adjust the "settings" of each heuristic (e.g., on/off, link to non-link text ratio threshold for table cell removal), in order to produce what that user deems the "best" results for a given web page. But this rather tedious manual tweaking model is not appropriate for most users. Therefore Crunch also automates analogous tweaking, by employing another set of heuristics to try to determine whether the DOM-pruning collection of heuristics mentioned above are properly narrowing in on the content. In particular, Crunch employs a multi-pass filtering architecture, where the DOM tree produced by each stage of filtering is compared to the DOM tree input to that stage. If "too much" or "too little" has been removed, again determined heuristically, Crunch backs out of that pass, adjusts the heuristic settings, and retries the pass. See [7] for details.


**Figure 4 – CNN front page**

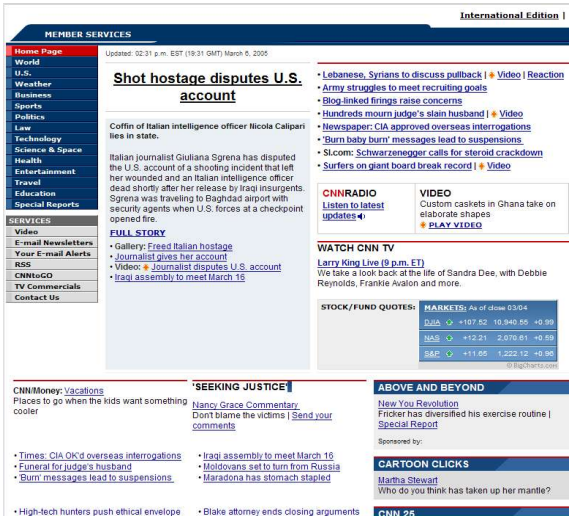



**Figure 5 – CNN front page through Crunch**

In addition, since we have found that what seem to us the "best" settings tend to differ significantly across different "genres" of websites (e.g., news vs. shopping – shopping sites tend to consist almost entirely of what the heuristics consider ads, but in this case they should not be removed), but be virtually identical within a genre, we have included in the Crunch proxy a utility that automatically classifies a website by genre. A large number of websites were clustered in advance offline, using their web search engine snippets as well as their front page text. Then when Crunch retrieves a page from a previously unknown website, that website is efficiently incrementally clustered to find the closest cluster. If the Crunch proxy's database contains previously determined "best known settings" for at least one member of that cluster, the heuristic settings from the closest such member are applied to the pages of the new website. Details are given in [8].

# 3.ACCESSIBILITY

We have tested Crunch extensively on a large corpus of popular websites and our results to date are promising. A small IRB-

approved usability evaluation with a blind volunteer is discussed in [7], and we have informally "shown" Crunch, via before and after screen reader renderings, to other blind web users. One of our volunteers suggested that we test Crunch on websites that are compliant with the W3C's Web Accessibility Guidelines [2].

We found that many websites offer text-only versions of their entire sites. With the huge number of inaccessible websites out there, any attempt to make a website universally accessible is highly commendable. However, we also found that text-only versions do not necessarily offer true accessibility. This could be for a number of reasons. The two most common reasons we found are non-descriptive link text and inaccessible forms. Visually impaired web users might browse a text-only website by tabbing from one link to the next; however, link text such as 'click here' and 'more', which tend to feature in a text-only version derived from a graphics-heavy version, is not particularly meaningful or helpful. In forms, prompt text is not always properly assigned to each form item. Further, users with disabilities still have to read or listen through all the navigation menus and other non-content until they reach the desired content.

While Crunch does not directly address these problems, as it does not add data to websites but only extracts from what is already there, we believe our framework can provide a valuable service for accessible as well as inaccessible websites. The accessibility guidelines are not intended to force text-only versions of sites, but instead suggest guides to the creation and layout of the various elements on a web page so that a multitude of web browser platforms, and their users, can fully access them.

Crunch is intended to remove the extraneous elements of a web page, and enable user's browsers (and other HTTP clients) to receive only what the heuristics deem to be content. In the case of an accessible site, as shown in Figures 6 and 7, Crunch zeros in on what it seems likely that the user is interested in. While the overall site is accessible, Crunch's heuristic filters still apply, and usefully extract content for more immediate access by a screen reader or screen magnifier. In some accessible cases, as in the example shown in Figure 8, Crunch finds nothing to remove - except of course it could replace the images with the ALT text, which can also be done by most browsers.

# 4.CONCLUSION

In order to improve the web experience for persons with disabilities, the W3C's Web Accessibility Initiative has created a set of guidelines for web page design. It is unfortunate that many websites are not (yet) compliant. We have developed a web proxy called Crunch that utilizes a collection of heuristics, essentially tunable filters operating on the DOM representation of the HTML web page, in its effort to extract core content from those web pages – with, among other goals, that the resulting web page be accessible even if the original was not. In this position paper, we present our preliminary findings that Crunch can also be useful for extracting content from sites that are deemed accessible, that is, compatible with the WAI guidelines. While no heuristic approach could ever perfectly define the user's (or author's) intent, Crunch's current set of heuristics performs well, and the framework also supports extension with new heuristic plugins (see [7] for plugin interfaces and example plugins using heuristics developed by others). To aid future content extraction and accessibility tools to perform even better, we would suggest revising accessibility guidelines to include tags that explicitly indicate (e.g., bracket) the core content on each web page.

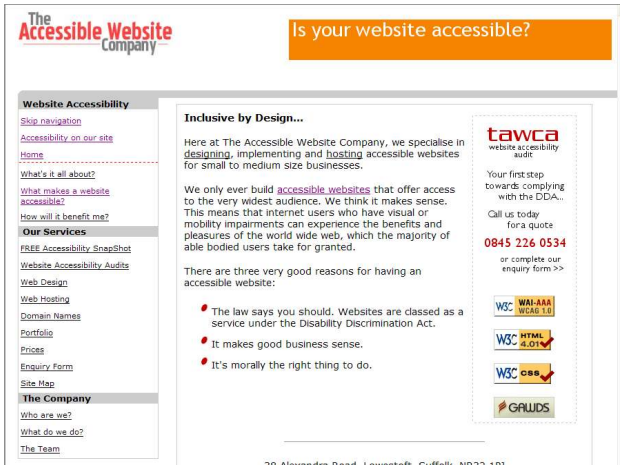
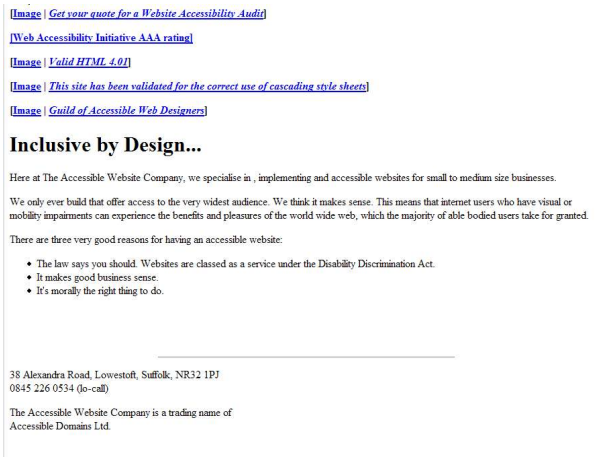



**Figure 6 – Accessible Website Company's front page**


[Image | *Get your quote for a Website Accessibility Audit*]

[Web Accessibility Initiative AAA rating]

[Image | *Valid HTML 4.01*]

[Image | *This site has been validated for the correct use of cascading style sheets*]

[Image | *Guild of Accessible Web Designers*]

**Inclusive by Design...**

Here at The Accessible Website Company, we specialise in , implementing and accessible websites for small to medium size businesses.

We only ever build that offer access to the very widest audience. We think it makes sense. This means that internet users who have visual or mobility impairments can experience the benefits and pleasures of the world wide web, which the majority of able bodied users take for granted.

There are three very good reasons for having an accessible website:

- The law says you should. Websites are classed as a service under the Disability Discrimination Act.
- It makes good business sense.
- It's morally the right thing to do.

38 Alexandra Road, Lowestoft, Suffolk, NR32 1PJ
0845 226 0534 (lo-call)

The Accessible Website Company is a trading name of Accessible Domains Ltd.

**Figure 7 – Same accessible site through Crunch**

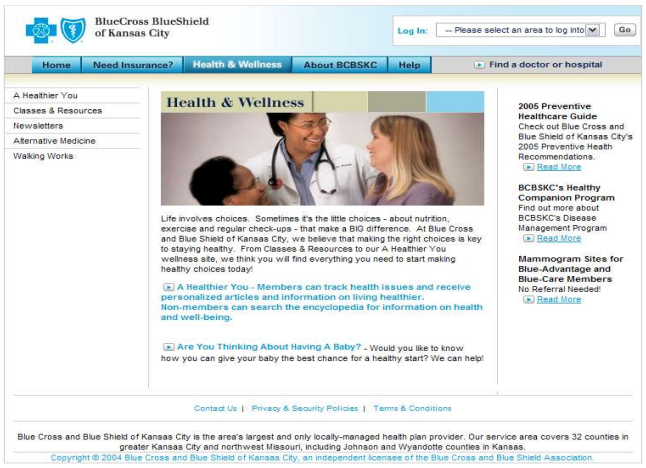



**Figure 8 – Crunch leaves some sites alone**

## 5.ACKNOWLEDGMENTS

The Programming Systems Laboratory is funded in part by National Science Foundation grants CNS-0426623, CCR-0203876 and EIA-0202063, and in part by Microsoft Research.

## 6.APPENDIX

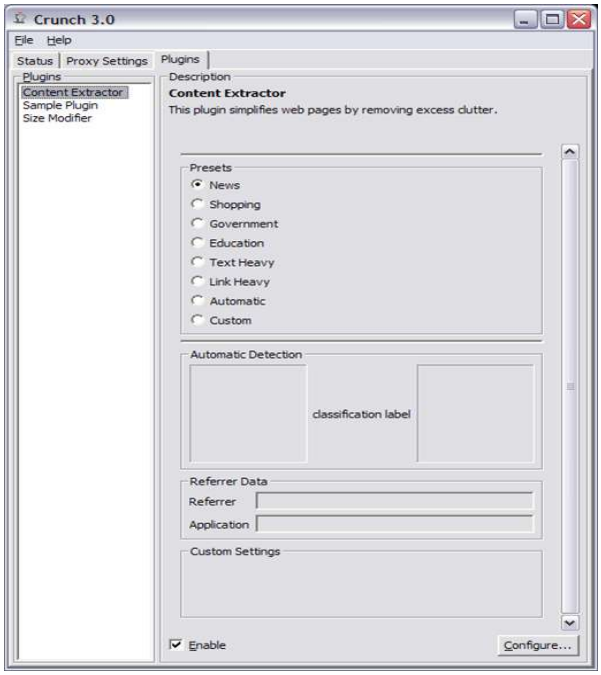



**Figure 9 – Crunch administrative interface**

## 7.REFERENCES

[1] Michael F. Chiang, Roy G. Cole, Suhit Gupta, Gail E. Kaiser, Justin B. Starren, "Computer and World Wide Web Accessibility by Visually Disabled Patients: Problems and Solutions", to appear in *Survey of Ophthalmology*, 2005.

[2] W3C Accessibility Guidelines 2.0 - http://www.w3.org/TR/WCAG20/.

[3] Joint Education and Outreach project of ICDRI, The Internet Society Disability and Special Needs Chapter, and HiSoftware - http://www.contentquality.com/Default.asp.

[4] W3C Accessibility Initiative accessibility evaluation tools - http://www.w3.org/WAI/ER/existingtools.html.

[5] Orkut Buyukkokten, H. Garcia-Molina, A. Paepcke, "Seeing the Whole in Parts: Text Summarization for Web Browsing on Handheld Devices", *World Wide Web Conf.*, 2001.

[6] Suhit Gupta, Gail Kaiser, David Neistadt, Peter Grimm, "DOM-based Content Extraction of HTML Documents", *World Wide Web Conf.*, 2003.

[7] Suhit Gupta; Gail E Kaiser, Peter Grimm, Michael F Chiang, Justin Starren, "Automating Content Extraction of HTML Documents", to appear in *World Wide Web Journal*, 2005.

[8] Suhit Gupta, Gail Kaiser, Salvatore Stolfo, "Extracting Context to Improve Accuracy for HTML Content Extraction", Columbia University TR CUCS-045-04, 2004 - http://www1.cs.columbia.edu/~library/TR-repository/reports/reports-2004/cucs-045-04.pdf.