SADIe: Semantic Annotation for Accessibility

Sean Bechhofer, Simon Harper, and Darren Lunn

School of Computer Science, University of Manchester,
Kilburn Building, Oxford Road,Manchester, M13 9PL, UK
{sean.bechhofer,simon.harper,darren.lunn}@manchester.ac.uk

Abstract. Visually impaired users are hindered in their efforts to access the largest repository of electronic information in the world – the World Wide Web (Web). The web is visually-centric with regard to presentation and information order / layout, this can (and does) hinder users who need presentation-agnostic access to information. Transcoding can help to make information more accessible via a restructuring of pages. We describe an approach based on annotation of web pages, encoding semantic information that can then be used by tools in order to manipulate and present web pages in a form that provides easier access to content. Annotations are made directly to style sheet information, allowing the annotation of large numbers of similar pages with little effort.

1 Introduction

Access to, and movement around, complex hypermedia environments, of which the web is the most obvious example, has long been considered an important and major issue in the Web design and usability field [**?**,**?**]. The commonly used slang phrase 'surfing the web' implies rapid and free access, pointing to its importance among designers and users alike. It has also been long established [**?**,**?**] that this potentially complex and difficult access is further complicated, and becomes neither rapid nor free, if the user is visually impaired[1].

Annotation of web pages provides a mechanism to enhance visually impaired peoples' access to information on web-pages through an encoding of the meaning of that information. Annotations can then be consumed by tools that restructure or reorganise pages in order to pull out salient information. However, when working in the real world, there are issues we must face. Empirical evidence suggests that authors and designers will **not** create separate semantic mark up to sit with standard XHTML[2] because they see it as an unnecessary overhead. In addition, designers will not compromise their desire to produce "beautiful and effective" web sites.

Recent moves towards a separation of presentation, metadata and information such as Cascading Style Sheets (CSS) [**?**], can help to alleviate some of the problems, but there are still many issues to be addressed. Sites such as CSSZenGarden[3] are models of the state of the art, but still remain relatively inaccessible

[1] Here used as a general term encompassing the WHO definition of both profoundly blind and partially sighted individuals [**?**].

[2] Extensible Hypertext Markup Language

[3] http://www.csszengarden.com/

to visually impaired people, however, as the information is rendered in an order defined by the *designer* and not in the order required by the *user*.

Visually impaired users interact with these systems in a 'serial' (audio) manner as opposed to a 'parallel' (visual) manner. Content is read from top left to bottom right, there is no scanning and progress through information is slow. Visually impaired users are at a disadvantage because they have no idea which items are menus, what the page layout is, what the extent is, and where the focus of the information is. Even when CSS concepts *do* look as though they have a meaning with regard to the information there is no way of relating this due to the lack of machine interpretable semantics. At this point, we can turn our attention to advances and developments in the Semantic Web.

Before doing so, we must stress that a key consideration for us is the support of designers. We wish to support the designer because in doing this we make sure our target user group are supported by the designers' creation. In our conversations with designers [**?**,**?**] the message we hear given is:

> *"If there is any kind of overhead above the normal concept creation then we are less likely to implement it. If our design is compromised in any way we will not implement. We create beautiful and effective sites, we're not information architects."*

We suggest that designers need a lightweight no-frills [**?**] approach to include semantic information relating to the role of document elements within XHTML documents; thus we need to ensure that any technical solutions proposed should incur a minimal costs in design overhead. We consider this to be "semantics" as it exposes additional information about page elements that would otherwise be implicit – for example a menu is an element that should be treated in a particular way by a client browser. CSS information may describe how to render the element in an appropriate way, but tells us nothing about the intended interpretation (and thus semantics) of the element.

The Semantic Web vision [**?**] is of a Web in which the underlying semantics of resources are made explicit using representations that are amenable to machine processing. The consideration of the problem outlined above leads us to the question:

> *Can semantic information be exposed in general purpose web-pages such that the information within the page can be transformed into a version as accessible to visually impaired users as it is to sighted users, without compromising the page's design vision?*

Our proposed approach, known as SADIe, can be summarised as follows. We provide an ontology that describes the meaning of elements found within XHTML meta tags and associate this with the data found in pages through an annotation of CSS style sheets. In this way, CSS presentation will be unaffected but semantics will be an explicit part of the data. We can then provide tools that consume this information, manipulating the documents and providing appropriate presentations to the user. A characteristic of our approach which is worth highlighting is that – in contrast to the majority of Semantic Web work concerning semantic annotation – we are *not* here concerned directly with annotation of domain content, but rather in exposing semantics relating to the presentation of material and the document structure. In addition, there is novelty in the attempt to annotate the CSS style sheet rather than the individual documents. Although this may not allow us to provide detailed annotations of individual document elements in particular documents, the broad-brush approach results in a low-pain, high-gain situation. As we see in our example discussed throughout the paper, annotation of a single CSS style sheet can result in the ability to transcode large numbers of pages that share the CSS presentation. Annotation via the CSS also allows us to deal with legacy sites.

The needs of visually impaired users accessing pages via audio are similar in a number of ways to those using mobile or small-screen devices – for example, only a small portion of the page is viewable at any point. Thus, although our primary motivation for this work is in supporting the needs of visually impaired users, we see potential benefit in the support of small-screen and mobile devices.

Earlier work[4] puts forward the basic ideas behind our approach [**?**]. Here, we expand on those ideas, providing a more detailed description of our prototype implementation along with an evaluation. The remaining sections of the paper are structured as follows. We provide a brief overview of background and context. This is followed by a description of the technical approach being taken by SADIe, along with examples. We present results from a preliminary *technical* evaluation, showing the viability of our approach, and conclude with discussion and pointers to future directions.

2 Background and Context

An overview of related work and technology is given in [**?**]. A brief summary is given here. Our work draws on a number of strands, including the Semantic Web, encoding semantics in documents, transcoding, and annotation.

A variety of techniques have been proposed for embedding XML/RDF information in HTML documents. This includes work from the TAG project [**?**,**?**], the use of the XHTML `link` element [**?**], the HyperRDF system [**?**], Augmented Metadata for XHTML [**?**] and the W3C Web Co-ordination Group's work on GRDDL [**?**].

None of these methods prove ideal for our purposes, some due to problems with validation (TAG, XHTML `link`, and HyperRDF). GRDDL is about embedding extra information through *modification* of that document. We are interested in associating additional information with documents, but not through an embedding – rather we aim to make use of existing information already present and *expose* it in a more explicit fashion. This is similar to the Deep Annotation [**?**] approach proposed by Volz et. al., where annotation of a logical schema can lead to annotation of resources or web pages that are dynamically generated from a database.

Transcoding is a technology used to adapt Web content so that it can be viewed on any of the increasingly diverse devices found on today's market.

[4] Going under the name of LLIS

Transcoding normally involves: (1) Syntactic changes like shrinking or removing images [?]; (2) Semantic rearrangements and fragmentation of pages based on the meaning of a section [?,?]; (3) Annotation of the page created by a reader [?]; and (4) Generated annotations created by the content management system [?]. In **Semantic Transcoding**, the semantics provide the machine understandability and knowledge reasoning and the transcoding provides the transformation technique. Current systems are at present however limited to page analysis [?] where a page built after a set template can be analysed and transformed by semantic or semantic like technologies.

The goal of **annotations** for Web content transcoding is to provide better support either for audio rendering, and thus for visually impaired users, or for visual rendering in small screen devices. Various proxy-based systems to transcode Web pages based on external annotations for visually impaired users have been proposed [?,?]. The main focus is on extracting visually fragmented groupings, their roles and importance. They do not support deep understanding and analysis of pages, and in consequence the supported transcoding is somewhat constrained. DANTE [?] uses an ontology known as WAfA, providing a representation of knowledge about mobility of visually impaired people. Annotations made on pages then drive a page transformation process. The DANTE approach annotates individual page fragments using XPointer which results in a rather brittle solution. Annotation at the level of the stylesheet (as proposed here) should, provide a solution which is more resilient to change. Other work centres on small-screen devices and proposes a system to transcode an HTML document by fragmenting it into several documents [?]. The transcoding is based on an external annotation framework. Annotation in the **Semantic Web** context [?] has tended to focus on providing annotations on documents in order to improve search/retrieval or integration. The focus is thus on identifying particular concept instances that are described by web pages. Here though, as introduced in Section **??** we are providing an explicit description of the meaning or intended interpretation of the structure of the document, rather than the objects in the world that the document is talking about.

Each of the transformations described above are fraught with problems with regard to the acceptability of the resulting generation. This is especially the case when sighted users as well as visually impaired users wish to use the same page. Automatic transcoding based on removing parts of the page results in too much information loss and manual transcoding is near impossible when applied to dynamic web sites. Most systems use their own bespoke proxy-servers or client side interfaces and these systems require a greater setup cost in-terms of user time. Finally, some systems require bespoke automatic annotation by a content generator and so are not usable by every user and all systems.

3 System Description

From the preceding discussion, we can identify the following requirements for our system.

- Semantic descriptions of element roles

– Non-destructive, unobtrusive annotation of pages
– Transcoding based on descriptions

The approach taken in our prototype system can be loosely described as follows. An upper level ontology provides basic notions that encapsulate the role of document elements. In the current implementatino, this is largely a taxonomy consisting of elements such as menu or header. In addition, an element can be characterised as a removableCSSComponent – one which can be removed without significantly impacting on the information carried within the document or given a priority that express how important the element is considered to be. This upper level ontology is defined in isolation from a particular site, providing an abstraction over the document structure. For a particular CSS stylesheet, we provide an extension of the ontology giving the particular characteristics of the elements appearing in that stylesheet. We can consider this extension to be an annotation of the stylesheet elements – it provides information telling us, for example, whether particular elements in the stylesheet can be considered to be removable or important. Figure **??** shows an example of a site-specific ontology extension.

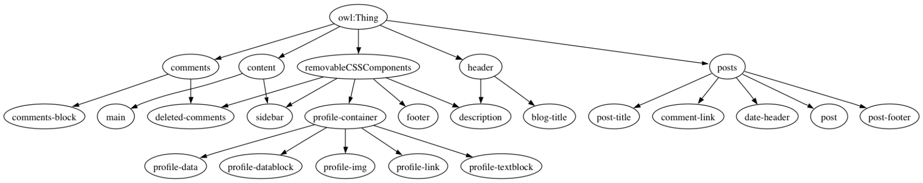


Fig. 1. `blogger.com` ontology fragment

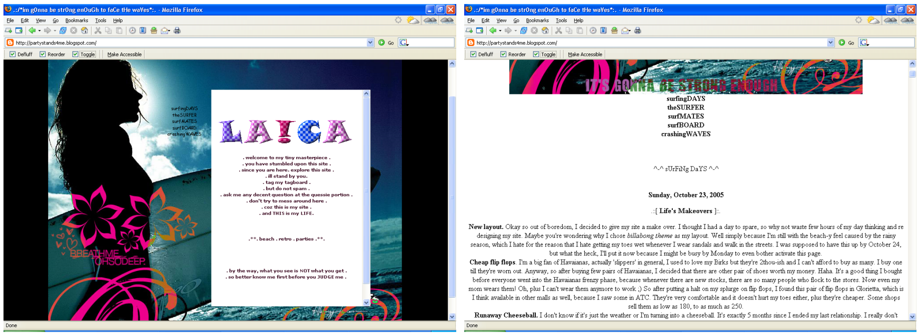
Annotation of the CSS elements allows us to make our assertions about the meaning of the document structure at an appropriate level. A CSS stylesheet often contains inherent "semantic" information about the implicit intended function of the elements, but which is not necessarily presented in a manner which is amenable to machine processing. For example, `blogger.com` (see below) provides elements such as `comment` and `profile`. This is, we feel, a clear example of a problem that Semantic Web technology and approaches are intended to represent – there is (currently) no explicit characterisation of the semantics of these tags, and they are thus opaque to understanding by machine. By providing a mapping from these elements to a shared upper level ontology of document elements, we can provide the opportunity for applications to manipulate documents in appropriate ways. The SADIe application then uses the ontology to determine appropriate transformations to be made to a source document.

Our prototype is delivered as a Mozilla toolbar extension called SADIe (see Fig. **??**) which has three types of functionality; *De-Fluff*, *ReOrder*, and *Toggle Menu*. *De-fluff* removes all the information that is removable based on its location in the ontology not in the CSS or XHTML. *ReOrder* rearranges the page so that the most important pieces of information are moved to the top of the document based on the values assigned to the elements in the ontology. Finally, *Toggle*

Menu moves menu items from their current location to the top of the DOM (as a child of the DOM body). In the current prototype, requests and operations are pre-configured and anchored to checkboxes on the toolbar (see Fig **??**), with checkboxes for the functionalities described above and a button to execute the SADIe transformations. When transformation is selected, appropriate requests are sent to the Ontology Service. In de-fluffing, for example, all of the removable items are requested. The Service complies and the SADIe parses the Document Object Model (DOM) looking for removable components and discarding them.

As an illustrative example, we consider a blogging site `blogger.com`, a legacy site for which we have created a sample ontology, and show how our application can transform pages into more accessible forms. Blogs are fast becoming ubiquitous on the web, with sites such as `blogger.com` providing easy mechanisms allowing users to publish their thoughts or opinions on a wide range of subjects. As many users are neither interested nor competent in issues surrounding web design or the use of markup languages, `blogger.com` provides standard mechanisms for marking up and structuring pages. CSS stylesheets are used to control the presentation. In this way, a large number of pages can be delivered with almost identical underlying structure, but with widely differing "look and feel" in terms of the colour, fonts, layout etc. It is this similarity in structure that we exploit – by providing a mechanism that allows us to annotate at the CSS level. A single annotation is then applicable to a large number of pages. One key feature is that because we do not annotate or modify the actual XHTML document our system does not force developers into costly and time consuming re-engineering to achieve backward compatibility. We extended our SADIe Ontology with web logging terms and from these created a specific ontology for `blogger.com` (see earlier Fig. **??**). The ontology was created in OWL using the Protégé tool; it comprises a small set of concepts and sub-concepts derived from the `blogger.com` CSS Template. Some of these concepts were described as being removable, and a measure of importance assigned using integer values. A fragment of the ontology is shown in Fig. **??**. The hierarchical (subclass) relationships for the class removableCSSComponents have been inferred using a reasoner and show that deleted-comment, description, footer, profile-container, and sidebar can all be removed.

Interestingly our ontology contains two concepts (recently and archive-list) which have no CSS entry but which *are* used as CSS-class identifiers in blogger. Thus there is no extra presentational information associated with elements using these identifiers. These two concepts enclose the recent posts list and the archive month lists and so, in fact, act like menus into previous postings. Axioms asserting that the concepts recently and archive-list are subclasses of menu are added to the ontology. As we will see below our application can then treat recently and archive-list as kinds of menus and perform appropriate operations up on them. Again, this is an example of the explicit specification of the information content of the document.



.:/*im g0nna be str0ng en0uGh to faCe tHe waVes*:. - Mozilla Firefox
File
Edit
View
Go
Bookmarks
Tools
Help
Go
Done

http://partystands4me.blogspot.com/

☑ Defluff ☑ Reorder ☑ Toggle Make Accessible

surfingDAYS
theSURFER
surfMATES
surfBOARD
crashingWAVES

LA!CA

. welcome to my tiny masterpiece .
. you have stumbled upon this site .
. since you are here. explore this site .
. ill stand by you.
. tag my tagboard .
. but do not spam .
. ask me any decent question at the quessie portion .
. don't try to mess around here .
. coz this is my site .
. and THIS is my LIFE.

.**. beach . retro . parties .**.

. by the way, what you see is NOT what you get .
. so better know me first before you JUDGE me .

IT'S GONNA BE STRONG ENOUGH

surfingDAYS
theSURFER
surfMATES
surfBOARD
crashingWAVES

^-^ sUrFiNg DaYS ^-^

Sunday, October 23, 2005

.::[Life's Makeovers]::.

New layout. Okay so out of boredom, I decided to give my site a make over. I thought I had a day to spare, so why not waste few hours of my day thinking and re designing my site. Maybe you're wondering why I chose *billabong theme* as my layout. Well simply because I'm still with the beach-y feel caused by the rainy season, which I hate for the reason that I hate getting my toes wet whenever I wear sandals and walk in the streets. I was supposed to have this up by October 24, but what the heck, I'll put it now because I might be busy by Monday to even bother activate this page.
Cheap flip flops. I'm a big fan of Havaianas, actually 'slippers' in general, I used to love my Birks but they're 2thou-ish and I can't afford to buy as many. I buy one till they're worn out. Anyway, so after buying few pairs of Havaianas, I decided that there are other pair of shoes worth my money. Haha. It's a good thing I bought before everyone went into the Havaianas frenzy phase, because whenever there are new stocks, there are so many people who flock to the stores. Now even my mom wears them! Oh, plus I can't wear them anymore to work ;) So after putting a halt on my splurge on flip flops, I found this pair of flip flops in Glorietta, which is I think available in other malls as well, because I saw some in ATC. They're very comfortable and it doesn't hurt my toes either, plus they're cheaper. Some shops sell them as low as 180, to as much as 250.
Runaway Cheeseball. I don't know if it's just the weather or I'm turning into a cheeseball. It's exactly 5 months since I ended my last relationship. I really don't

Fig. 2. Transcoding a Blog (see `http://partystands4me.blogspot.com/` for the original)

Figure **??** illustrates the tool in action. To the left we see the original page before transcoding. In this case, the blog contents are relatively inaccessible, even to sighted users. After transcoding (on the right), the blog entries are exposed.

When an XHTML document arrives in the Mozilla browser with a SADIe toolbar the application first determines whether there is an ontology associated with the document (see Section **??**). If such an ontology is present it is retrieved much like Mozilla retrieves the CSS document. The ontology is then passed to an Ontology Service which is used to provide functionality relating to the ontology such as classification (e.g. what are all the removable elements?).

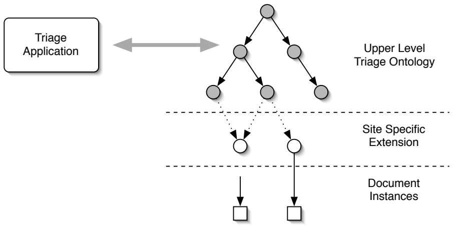


Fig. 3. Ontology and Site-specific extensions

In this way all pages created using `blogger.com` (close to a million blogs) can be modified by using this one simple ontology and tool. The particular ontology is specific to the site. However, the upper level definitions which are used by the tool in order to determine which elements to be removed are generic – integrating an additional site into the system simply requires the definition of a mapping from the CSS elements of the site into the base SADIe ontology. Any site's pages can be de-fluffed as long as the elements that are removable are identified. We do not need to hard-wire the information about the CSS elements into the application – this is encoded in the ontology which is then used by the application (see Figure **??**). The upper level ontology describes concepts that are relevant to the process – for example menu – and the application's behaviour is specified in terms of these concepts. Site specific extensions describe their CSS elements in terms of these upper level concepts (e.g. recently as discussed in the example).

The approach is non-intrusive and works hand-in-hand with existing technologies used to control presentation. Users view the document in a web browser as normal. Browsers that are 'semantic-aware', however, can use the extra information to provide more intelligent access to the information than before.

In additoin, as we do not annotate or modify the actual XHTML document our system does not force developers into costly and time consuming re-engineering to achieve backward compatibility.

We are suggesting a simple and flexible system without a significant semantic overhead. To achieve this we use a group of techniques to encode semantics directly into a page:

Class and ID Attributes XHTML `class` or `id` attributes are used to encode a piece of semantic information in the form of a concept-class or property into a defined piece of XHTML delimited by the closing element identifier. This is normally achieved by using the `div` and `span` elements to conjoin both the presentation style (CSS) and the semantic meaning (ontology) to the user.

Ontology Our approach involves an annotation on CSS elements in order to describe their properties. The identification of the ontology to use may be done in a number of ways. These follow the methods laid down by the originators of CSS in order to link stylesheets to XHTML pages, e.g. through an explicit XHTML `<link>` element, a direct encoding of the ontology in the page or by searching for an ontology in the root directory of the web site;

The SADIe application parses the XHTML DOM and the document is then viewed as a knowledge base – instances are elements from the document such as `<span>` or `<div>` elements with their associated classes being taken from CSS `id` or `class` attributes (see Fig. **??** - '`ID / CLASS` Results'). Information about the classes in the ontology is then used to determine the actions to take. For instance, if we wanted to remove all the concepts (and therefore CSS-blocks) that are removable, then this involves a query for those elements classified as removable in the ontology. We can here make use of the concept hierarchy, potentially providing descriptions of the document elements that are organised and classified using inference. Inference may be brought into play here – for example, it may be that we use a number of characteristics to determine whether an element should be considered as removable.

Similarly, as discussed above, concepts such as recently and archive-list are classified as kinds of menu. As SADIe knows how to process menu concepts (from the SADIe Ontology), when it encounters an archive-list, this can be handled using appropriate mechanisms – say moving it to the top of the document or back to its original position. A key point to note here is that the reordering of the DOM in general does *not* change the visual appearance as the CSS takes care of the layout. It does however move the information in the XHTML document and changes are noticeable if the style information is removed. This is exactly the outcome we hoped for because access technologies access the XHTML DOM as presented and often exclude the style and placement information.

Building a transformable web site is now a relatively straightforward activity. XHTML pages and the CSS are already built as part of the standard site creation. The addition required is the identification of the ontology that assists in

Category	Name	URL	CSS	Failure	Entry Point
Corporate Sites	Microsoft Corporation	http://www.microsoft.com/	Mixed	2	Success
	Digital Designs	http://www.digitaldesign.us	Pure	0	Success
	Stagecoach Buses	http://www.stagecoachbus.com/	Pure	0	Success
	British Nuclear Fuels	http://www.bnfl.com/	Pure	0	Success
	Epson Corporation	http://www.epson.co.jp/e/	Mixed	1	Success
Content & Media	Blogger	http://www.blogger.com/	Pure	0	Success
	The Mac Observer	http://www.macobserver.com/	Pure	0	Success
	New Musical Express	http://www.nme.com/	Mixed	5	Failure
	BBC News	http://news.bbc.co.uk/	Mixed	2	Failure
	CNN International	http://edition.cnn.com/	Mixed	1	Failure
Search Engines	Google	http://www.google.co.uk/	None	5	Failure
	Yahoo	http://uk.yahoo.com/	Mixed	0	Success
	Ask Jeeves	http://www.askjeeves.co.uk/	Mixed	0	Success
	MSN Search	http://search.msn.com/	Pure	0	Success
	HotBot	http://www.hotbot.co.uk/	Pure	0	Success
Directories	Google Directory	http://directory.google.co.uk/	None	5	Failure
	Yahoo Directory	http://uk.dir.yahoo.com/	None	5	Failure
	This Is Our Year	http://www.thisisouryear.com/	Mixed	2	Success
	HotSheet	http://www.hotsheet.com/	Pure	0	Success
	HaaBaa Web Directory	http://www.haabaa.com/	Mixed	0	Success
Portals	AOL UK	http://www.aol.co.uk/	Mixed	0	Success
	MSN UK	http://www.msn.co.uk/	Mixed	2	Success
	Wanadoo	http://www.wanadoo.co.uk/	Mixed	4	Success
	Virgin Net	http://www.virgin.net/	Mixed	4	Success
	Tiscali UK	http://www.tiscali.co.uk/	Pure	0	Success
E-stores	Play	http://www.play.com/	Mixed	0	Success
	Amazon UK	http://www.amazon.co.uk/	None	5	Failure
	Tiny	http://www.tiny.com/	Mixed	1	Success
	Tesco	http://www.tesco.com/	Mixed	1	Success
	Red Letter Days	http://www.redletterdays.co.uk/	Mixed	1	Success
Virtual Hosting	Bravenet	http://www.bravenet.com/	Mixed	1	Success
	InMotion Hosting	http://www.inmotionhosting.com/	None	5	Failure
	Path Host	http://www.pathhost.net/	Mixed	0	Success
	Honest Web Host	http://www.honestwebhost.com/	Mixed	5	Failure
	Netwalker Internet Services	http://www.netwalker.biz/	Mixed	0	Success
Universities	University of Manchester	http://www.manchester.ac.uk/	Mixed	0	Success
	University of York	http://www.york.ac.uk/	Mixed	0	Success
	University of Sheffield	http://www.shef.ac.uk/	Mixed	1	Success
	University of Oxford	http://www.ox.ac.uk/	Mixed	0	Success
	University of Southampton	http://www.soton.ac.uk/	Mixed	1	Success

Table 1. SADIe Evaluation Results

the transformation task. As discussed above, this could be either via a `<link>`, a direct encoding, or the inclusion of the ontology in a standard location.

4 Evaluation

In order to explore the viability of our proposed approach, we conducted a small *technical* evaluation. We are chiefly interested here in evaluating the first part of our objective: *Can semantic information be exposed in general purpose web-pages such that the information within the page can be transformed?* Thus for the purposes of this evaluation, we make an assumption that the proposed transformations such as removal of unnecessary items of reordering of menus *are* useful operations in improving accessibility[5].

The W3C's Web Accessibility Initiative (WAI) provides strategies and guidelines that web designers can use to make the Web accessible to people with disabilities. These guidelines are targeted at designers using current technology and techniques, such CSS and XHTML. The main focus of our approach is not web site design, but some of the principles in the guidelines can be applied when evaluating SADIe. The W3C guidelines [?] include steps such as:

1. Select a sample of different kinds of pages from the Web site to be reviewed. This must include all the pages that people are likely to use to enter the site.

[5] Of course, such a claim is open to challenge, and we intend to pursue further User Evaluations in order to investigate this.

2. Use a graphical user interface browser and examine a selection of pages while adjusting the browser settings.
3. Use a voice browser or a text browser and examine the Web site while checking if equivalent information available through the voice or text browser is available through the GUI browser and that the information presented is in a meaningful order if read serially.

Choose Sample Web Pages. Amitay et. al. [**?**] propose that while web sites are different visually, if a web site's role is taken into account, then there are some similarities. By using web site roles, they produced eight categories that can be used for classifying web sites. These categories are Corporate Sites, Content and Media, Search Engines, Web Hierarchies and Directories, Portals, E-Stores, Virtual Hosting and Universities. By using these eight categories, we can gain some confidence that our evaluation uses a reasonable sample of the kinds of web sites that potential users of SADIe may access.

Five web sites from each category were selected, giving a total of 40 sites in the sample. The W3C guidelines specify that when evaluating a web site the entry point should be tested, as this is the first page the users will access. Therefore, the samples include the site entry point (usually index.html) of the web site, plus 4 other randomly chosen pages on the web site. This gave us a total of 5 pages per web site. With 40 web sites, we examined 200 web pages in total.

Apply SADIe to Each Page. We applied De-fluff, Reorder and Toggle to the page and observed the results.

Evaluate Results of SADIe for Each Page. Success of the transcoding was determined by observation of the resulting page. Taking into account what SADIe was designed to do, we asked the following questions of a transcoded page:

1. Have all obstacles marked removable been removed?
2. Are there multiple columns on the page?
3. Has all formatting that uses tables been removed?
4. Is there anything that breaks up the flow of text?
5. Are all blocks of text aligned vertically as opposed to horizontally?
6. Are all navigation links at the top of the page?

A positive answer to all these questions, was considered to indicate a successful transcoding by SADIe. A negative answer to any question was considered a failure. This assessment was performed by one of the authors.

Determine Web Site Success. In determining a web site's success or failure, we used the entry point to determine if the site succeeded or failed, following the WAI philosophy. If we can make a page that most people use accessible, then that is more important for the site than providing access to a page that few people will ever read.

Having established a framework for the evaluation, it was then applied to a sample of web pages. The sample sites were obtained by taking the first five web sites from each of the eight IBM categories were used.

Table **??** shows the results of the SADIe evaluation. The 40 web sites and their categories are noted as well as how many of the web pages on the site

CSS Type	Site Sample	Site Failures	Sample Error (%)	True Error Range (%)
Pure	9	0	0	0 - 0
Mixed	26	4	15	2 – 28
None	5	5	100	100 – 100
All	40	9	23	11 – 35
Pure/Mixed	35	4	11	1 – 21

Table 2. SADIe Web Site Evaluation Summary

failed the SADIe evaluation and if the entry point was a success or not. We also note how the presentation of the site was achieved. Pages using only CSS are designated Pure. None indicates no CSS usage. Mixed was for those sites that use CSS for formatting fonts and colours and headings etc, but use tables for layout purposes.

Table **??** shows a summary of results. The results are broken down to show the success rate of the various classes of CSS usage. These three categories are then further summarised. The Pure/Mixed CSS Type is the combined results of only those web sites that used Pure CSS for presentation and those that used a mixture of CSS and tables for presentation. We factor out the web sites that used no CSS as our design rationale is to make use of document structure as encapsulated through the use of CSS and XHTML. If there is no CSS then by design, we are unlikely to be able to transcode the page[6].

The column "Site Failure" indicates how many entry points in the sample failed to be correctly transcoded. The sample error is the proportion of web sites from the sample that failed. The True Error Range provides a range in which the true error lies for that class of web site (using a 95% confidence interval). From Table **??**, we can see that all the sites that used no CSS for presentation failed. This was expected – SADIe relies upon the CSS to capture the structure of the web page. If there is no CSS, there is nothing for SADIe to use for transcoding.

Discounting the sites that used no CSS, we consider that SADIe obtained reasonable results. All sites that used pure CSS were successfully transcoded. When the sites that used mixed presentation are included, the error rate increases. This is partly due to problems in separating columns of text. We observed that a common approach adopted by these mixed sites was to give the entire table a CSS class value, which SADIe could use, but not give the elements within the cells of the table a Class or ID value. So while SADIe could remove or reorder the table as a whole, the contents within the table were inaccessible to SADIe and so remained in columns. This in turn meant the screen reader would be unable to read the text properly and the page was deemed a failure. However, there were still a large number of web pages that were successful that mixed CSS and tables for presentation. Table **??** shows that the error rate for this category was 11%, with the true error lying in the range of 1% and 21%.

While these results are encouraging, they must be taken with a pinch of salt as we are making several assumptions. The first is that our confidence values assume that web site design follows a Normal Distribution. Secondly, we are assuming that our sample is an accurate reflection of the web pages that are available on the Web. Amitay et. al's proposal of categories based roles provides

[6] Clearly this is a limitation here, but we surmise that both the number and relative proportion of sites that use CSS and XHTML is likely to continue to increase.

a good guidance for selection. However, it is difficult to say that choosing only 5 web sites for each category, which we did, could accurately reflect that category when the number is so small and the selection was not purely random. Recall that we are basing success and failure on the structure and content of page after transcoding. While we can make a value judgement that the transcoded page will be more accessible, based on research in the field, a true user evaluation will be needed before we can be sure of SADIe's success.

While these assumptions need to be addressed, the initial results are promising. As Table **??** shows, the combined error rate when we tested web pages that used pure and mixed presentation was only 11%. While we are not claiming that SADIe can successfully transcode 89% of all web sites and make them accessible, this initial result does provide a good basis for continued investigation of the SADIe approach.

5 Conclusions and Further Work

We have described the first stage in a more elaborate system that will increase free access to information for all users. By knowing more about the intended meaning of the information that is being encountered visually impaired users can perform their own transformations on that information.

Transcoding can help to make information more accessible via a restructuring of pages. Unnecessary items that introduce clutter can be removed, while important items can be promoted to a position on the page where they are encountered earlier by assistive technologies such as screen readers. Doing this in a principled manner, however, requires that the implicit semantics of the document be made explicit. We have described an approach based on annotation of web pages, encoding semantic information that can then be used by tools in order to manipulate and present web pages in a form that provides easier access to content. The annotations use an ontology describing the basic semantic units found in the pages as described in style sheets. Annotations are made directly to style sheet information, allowing the annotation of large numbers of similar pages with little effort.

The approach is minimal in the overhead presented to the site designer. No constraints are made on the ways in which the layout and presentation of the site can be produced. This is one of our key requirements – as discussed, designers will ignore, or at the very least fight against, initiatives that compromise their work. Rather we make use of the fact that CSS elements are identified in the document – in a large number of cases, these elements do, in fact, already correspond to "meaningful" units of information. In addition, the approach makes no impact on the validation of XHTML documents.

An alternative approach might have been to use the underlying XML structure of the XHTML documents and then apply basic XSL technology to transcode. We see at least two problems with this. First, the current number of resources that are actually marked up using valid XHTML is small [**?**]. While browsers continue to be successful in handling badly formatted HTML, there is little incentive for authors to rectify this. Of course, our approach requires HTML+CSS, but our investigations (see Section **??**) suggest that the proportion of sites using CSS is significant enough to merit this requirement – CSS does not necessarily require valid HTML in order to allow the production of good-looking web pages. The second problem is that even if the documents are valid, the underlying XML structure is not sufficient to carry the required information. The XML document will have structure in the form of `h1` or `p` or possibly even `div` and `span` elements, but these alone are not sufficient to represent the various roles played by elements in a page – this richer detail is usually encoded in the style sheet.

The current prototype is still very much at the level of a proof-of-concept demonstrator and will benefit from further refinement. We plan to extend the upper level ontology to include more concepts covering document constructs along with the specification of further transcoding operations. Site-specific extensions of the ontology are currently produced manually – investigations of the automation or semi-automation of this process are also planned. Finally, we need further user evaluations of the tool to determine how effective it really is in increasing accessibility.

In summary, we propose that the inclusion of semantic information directly into XHTML is an effective way to assist visually impaired users in accessing web pages while not increasing or compromising the creation activity of authors and designers. By knowing the meaning of the information that is being encountered visually impaired users can perform their own transformations on that information.

References

1. M. Altheim and S. B. Palmer. Augmented Metadata in XHTML, 2002. `http://infomesh.net/2002/augmeta/`.
2. E. Amitay, D. Carmel, A. Darlow, R. Lempel, and A. Soffer. The connectivity sonar: Detecting site functionality by structural patterns. ACM Press, 2003.
3. C. Asakawa and H. Takagi. Annotation-based transcoding for nonvisual web access. In *Proceedings of the Fourth International ACM Conference on Assistive Technologies*, pages 172–179. ACM Press, 2000.
4. T. Berners-Lee. *Weaving the Web*. Orion Business Books, 1999.
5. T. Berners-Lee. RDF in HTML, 2002. `http://www.w3.org/2002/04/htmlrdf`.
6. B. Bos, T. Çelik, I. Hickson, and H. W. Lie. Cascading Style Sheets, level 2 revision 1 CSS 2.1 Specification. Candidate recommendation, W3C, February 2004. `http://www.w3.org/TR/CSS21/`.
7. M. Brambring. Mobility and orientation processes of the blind. In D. H. Warren and E. R. Strelow, editors, *Electronic Spatial Sensing for the Blind*, pages 493–508, USA, 1984. Dordrecht, Lancaster, Nijhoff.
8. O. Buyukkokten, H. G. Molina, A. Paepcke, and T. Winograd. Power browser: Efficient web browsing for PDAs. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 430–437. ACM Press, 2000.
9. C. Chen. Structuring and visualising the www by generalised similarity analysis. In *Proceedings of the 8th ACM Conference on Hypertext and Hypermedia*, New York, USA, 1997. ACM Press.

10. A. Chieko and C. Lewis. Home page reader: IBM's talking web browser. In *Closing the Gap Conference Proceedings*, 1998.
11. Codix.net;. *Textualize*;. http://codix.net/solutions/products/textualise/index.html.
12. D. Connolly. HyperRDF: Using XHTML Authoring Tools with XSLT to produce RDF Schemas, 2000. http://www.w3.org/2000/07/hs78/.
13. R. Furuta. Hypertext paths and the www: Experiences with walden's paths. In *Proceedings of the 8th ACM Conference on Hypertext and Hypermedia*, New York, USA, 1997. ACM Press.
14. S. Handschuh and S. Staab, editors. *Annotation for the Semantic Web*, volume 96 of *Frontiers in Artifical Intelligence and Applications*. IOS Press, 2003.
15. S. Harper and S. Bechhofer. Semantic Triage for Accessibility. *IBM Systems Journal*, 44(3):637–648, 2005.
16. S. Harper, Y. Yesilada, and C. Goble. Proceedings of the International Cross-Disciplinary Workshop on Web Accessibility. W4A, ACM Press, May 2004.
17. D. Hazaël-Massieux and D. Connolly. Gleaning Resource Descriptions from Dialects of Languages (GRDDL). W3c team submission, World Wide Web Consortium, May 2005. http://www.w3.org/TeamSubmission/grddl/.
18. M. Hori, G. Kondoh, K. Ono, S. ichi Hirose, and S. Singhal. Annotation-based web content transcoding. In *In Proceedings of 9th International World Wide Web Conference*, 2000.
19. N. Kew. Why Validate?, 2002. http://lists.w3.org/Archives/Public/www-validator/2001Sep/0126.html.
20. V. Mirabella, S. Kimani, and T. Catarci. A no-frills approach for accessible web-based learning material. In *Proceedings of W4A 2004*, pages 19–27. ACM Press, 2004.
21. W. Myers. *BETSIE:BBC Education Text to Speech Internet Enhancer*. British Broadcasting Corporation (BBC) Education. http://www.bbc.co.uk/education/betsie/.
22. Palmer, Sean B. RDF in HTML: Approaches, 2002. http://infomesh.net/2002/rdfinhtml/.
23. V. RNIB. A short guide to blindness. Booklet, Feb 1996. http://www.rnib.org.uk.
24. L. Seeman. The semantic web, web accessibility, and device independence. In Harper et al. [**?**], pages 67–73.
25. V. Y. S. Shan Chen, Dan Hong. An experimental study on validation problems with existing html webpages. In *International Conference on Internet Computing ICOMP 2005*, pages 373–379, 2005.
26. Simon Harper and Yeliz Yesilada and Carole Goble. Workshop Report: W4A - International Cross Disciplinary Workshop on Web Accessibility 2004. In *SIGCAPH Comput. Phys. Handicap.*, number 76, pages 2–20. ACM Press, November 2004.
27. H. Takagi and C. Asakawa. Transcoding proxy for nonvisual web access. In *Proceedings of the Fourth International ACM Conference on Assistive Technologies*, pages 164–171. ACM Press, 2000.
28. R. Volz, S. Handschuh, S. Staab, L. Stojanovic, and N. Stojanovic:. Unveiling the hidden bride: deep annotation for mapping and migrating legacy data to the Semantic Web. *Journal of Web Semantics*, 1(2):187–206, February 2004.
29. World Wide Web Consortium, http://www.w3.org/WAI/eval/Overview.html. *Web Accessibility Initiative*.
30. Y. Yesilada, S. Harper, C. Goble, and R. Stevens. Dante annotation and transformation of web pages for visually impaired users. In *The Thirteenth International World Wide Web Conference*, 2004.