

Lightweight Mark-Up for Low Cost Semantics

Simon Harper and Sean Bechhofer
sharper , seanb [@cs . man . ac . uk]

Information Management Group, Dept of Computer Science,
University of Manchester, Manchester, UK.
<http://augmented.man.ac.uk>

Abstract. Visually impaired users are hindered in their efforts to access the largest repository of electronic information in the world, namely the World Wide Web (Web). A visually impaired user's information and presentation requirements are different from a sighted user in that they are highly egocentric and non-visual. These requirements can become problems in that 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. Encoding semantic information directly into pages can help to alleviate these problems and support users who wish to understand the meaning as opposed to the presentation and order of the information. If this is to happen in the "real world", however, authors must incur no "semantic overhead" when creating pages. In addition, resistance will be encountered to measures that threaten to compromise web design. In this paper we describe preliminary work towards an approach enabling a low-cost, lightweight solution to semantic encoding.

1 Introduction

We assert that the most preferential way to enhance visually impaired peoples access to information on web-pages is to encode the meaning of that information into the specific web-page it refers to. However, there are problems. Empirical evidence suggests that authors and designers will not separately create semantic mark up to sit with standard XHTML¹ because they see it as an unnecessary overhead.

Recently, we have seen a movement towards a separation of presentation, metadata (XHTML), and information. However, this has not been enough to support the unfettered access of visually impaired users. Consider, the excellent 'CSSZenGarden' (see Fig. 1). The site is a model of the state-of-the-art: the application of standards, separation of presentation and content, and visually stunning too. But, it is still reasonably inaccessible to visually impaired people. Inspect the site without an applied stylesheet (see Fig. 2). 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. Given this interaction paradigm we can see that visually impaired users are still at a disadvantage because they have no idea items are menus, what the page layout is, what the extent is. In effect, the implicit meaning contained in the visual presentation (see Fig. 1). is

¹ Extensible Hypertext Markup Language

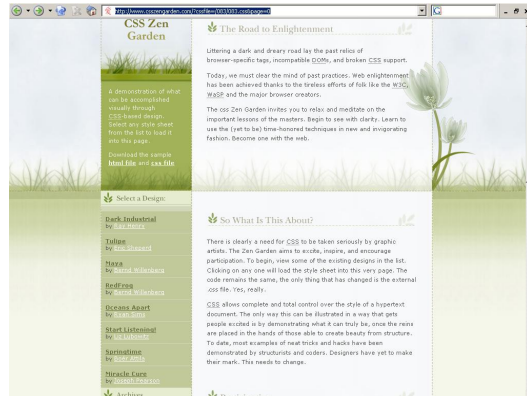


Fig. 1. Zen Garden with CSS 83

lost and any possibility of enhanced meaning is also not available as only authoring concepts (like footnote, heading, leftcolumn) are listed (see Fig. 2).

While authors and content creation engines still create non-standard CSS²-XHTML identifiers, they also often compound the problem by using linear paper based (book) metaphors such as: footer, header, bold, big, etc. This information can in fact be inferred from the coded style and presentation information contained within the CSS. This means the combination of identifier and presentation information together often represent a tautology.

Even when authoring concepts do look as though they have a meaning with regard to the information they are often mixed with un-descriptive qualifiers; and the problem is again compounded by the lack of an ontology in the event of there actually being some useful information to reason over. Therefore, the question which we faced and which this paper is dedicated to answering was this:

How can semantic information be built into general purpose web-pages such that the information is as accessible to visually impaired users as it is to sighted users, without compromising the page's design vision?

We based our question on a set of beliefs thus:

1. Visually impaired surfers need access to the meaning of information to assist in their cognition, perception, movement around that information, and to assist in the formulation of their world-view [4, 9]. This is the same for sighted users however pages are normally created with sighted users in mind.
2. Based on empirical and anecdotal evidence, authors and designers will not suffer a 'Semantic Overhead' when building pages.
3. A web page should be thought of an application, comprising functional elements and presentation / information elements, within an application (the browser).

² Cascading Style Sheet

4. Information should not need to be recreated (i.e. exist as XHTML for humans and RDF³ for agents) when the intended audience is human. The meaning should be seamless and be part of the data.
5. If we don't need to create explicit resources (RDF feeds etc) why should we?
6. Authoring concepts used as presentation identifiers are redundant when used with CSS as their presentational meaning is implicit in their technical definition.

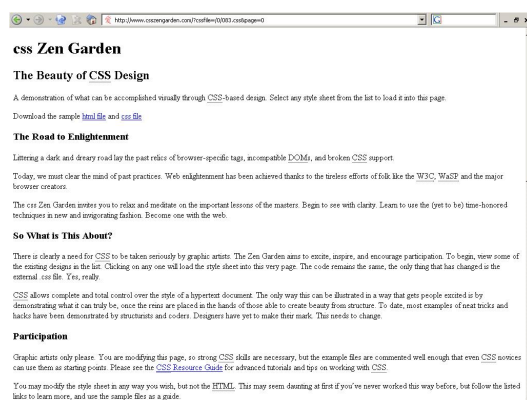


Fig. 2. Zen Garden without a CSS

This goal and set of beliefs led us to a simple, lightweight, and powerful solution. Create a grammar to represent the meaning of data within XHTML meta tags and encoded it into the data by leveraging the 'class' and 'id' attributes common to most XHTML elements. CSS presentation will be unaffected but semantics will be an implicit part of the data as opposed to an explicit duplicate representation (in say RDF(s) or N³ Notation). To achieve this we combine both XHTML elements that have meaning or that can be used to accurately infer meaning; and a bespoke grammar developed to enhance the limited XHTML syntax.

The focus of our system is to represent instances as information enclosed within meta elements along with concept and property identifiers as part of XHTML meta elements themselves. These elements can then be related to OWL Lite [8] ontologies defined in the normal way.

1.1 Synopsis

One of the goals of the Semantic Web vision is to make knowledge accessible to agents but with a strong human input and benefit. In this framework, our goal is to make

³ Resource Description Framework (Schema)

the role of the objects, that support visual accessibility through presentation, explicitly interpretable by humans (via web browsers) rather than just being visually interpretable. Therefore, it is necessary to associate metadata and semantics with XHTML objects (machine-readable vs machine-understandable). The rest of the paper can be summarised as follows:

Background We give an overview of how visually impaired people currently interact with web pages. We describe the problems associated with these methods and give an overview of current access paradigms and authoring concepts.

Related Work We present a small section on related work to place our contribution in context.

Low Cost Semantics We describe the concepts, rationale, and techniques behind our system focusing on the XHTML `abbr` / `acronym` elements and the `class` attribute. We show how these are referenced on XHTML pages and how our lightweight system can contribute to the accessibility of information via lightweight semantics.

Example As a preliminary case study we consider the simple ontology taken from ‘A Semantic Web Primer’ in an attempt to show how an ontology is represented using our methodology.

Why Does This Approach Aid Visually Impaired Users? We have identified the problem and suggested a solution but why do we think this is a useful solution?

Conclusion Finally, we focus on our conclusions from the work undertaken and look at future work including system evaluations.

2 Background

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 [5, 10]. 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 [4, 6] that this potentially complex and difficult access is further complicated, and becomes neither rapid or free, if the user is visually impaired⁴.

2.1 Current Access Paradigms

Visually impaired people usually access Web pages either by using screen readers or specialist browsers. If the Web pages are properly designed and laid out in a linear fashion, these assistive technologies can work satisfactorily. Some screen readers access the HTML / XHTML source code rather than solely reading the *screen*, which enables them to provide better support. However, not many pages are properly designed; the focus is usually on the visual presentation which makes audio interaction almost impossible.

⁴ Here used as a general term encompassing the WHO definition of both profoundly blind and partially sighted individuals [13].

Furthermore, chunking the page into several parts and presenting it in a nonlinear fashion is becoming popular which makes the provided functionalities of these assistive technologies insufficient. There are guidelines to aid the designers in creating accessible pages [1], unfortunately few designers follow these guidelines and therefore Web accessibility is still a problem.

Further problems also exist when trying to gain an overview of the page. Some screen readers, for instance Jaws [11], provide overview information when the user first accesses a page. This information often includes, for example, the number of headings in the page based on the “heading” tags in the source code. However, if the page is not appropriately designed, such information could be misleading.

2.2 The Problem with Authoring Concepts

Even when XHTML meta elements are used correctly and pages are created to standards and specifications, poor accessibility still persists. We believe this is because there are common misconceptions about what information is actually required by users. In our opinion this continued inaccessibility stems from the incorrect use of authoring concepts within the web-page.

Authoring concepts often hold information about the layout vocabularies used in transcoding and content management systems; but from a visual perspective. In this case, they do not consider the meaning of the objects in the page framework but are more interested in how the objects are presented in the Web landscape. The Web landscape is defined as the combination of the page and the agent (e.g, browser and assistive technologies such as screen readers). These concepts are more to do with the specific structures that can be used to define the overall layout of a page including for example, sections, summaries, abstracts, footers, etc. These constructs are usually implicit in the visual presentation of the page, and so many authors and transcoding systems seek to explicitly encode them in the underlying source code (e.g., HTML). However, this kind of terminology is less useful and therefore inaccessible in any other form of interaction (e.g., audio interaction through screen readers). Transcoders aim to define a vocabulary that is already widely used between the designers but not formally explained and defined, that is to say they try to make the domain knowledge explicit. However, they use the wrong paradigm, that of the linear and visual layout as opposed to the really useful information – the meaning of the actual instance of data itself.

Authors and systems need to move away from this paradigm of providing what they **THINK** users need and focus on what the creator actually **MEANS**. In this way visually impaired users can decide for themselves what is useful, and what is not.

3 Related Work

Adding semantics to an XHTML document is not a new concept. It has been thought about since the late 1990’s however concrete solutions were proposed as early as 2002. Tim Berners-Lee proposed embedding XML RDF in HTML documents as part of the tag project [3], however these documents would not validate as XHTML and so did not find favour among the community [12]. A version was created that did validate by

the inclusion of a small DTD using XHTML Modularisation. However, this was not deemed a good solution as unique extensions have to be created on a whim. In fact the work concluded that the RDF specification specifies how to understand the semantics (in terms of RDF triples) in an RDF document that contains only RDF, but does not explain how and when one can extract semantics from documents in other namespaces which contain embedded RDF. It goes on to say that the XHTML specification explains how to process XHTML namespace content, but gives no indication about how to process embedded RDF information [3]. Other methods have been proposed in which the object or script elements are used, however, the code becomes unreadable and therefore less workable although the RDF can be linked to in an external file [14]. The use of the XHTML `link` element has also been proposed, however the main problem with this method is that the RDF is not actually then embedded in the HTML source but in a separate file [14]. This file is then at the mercy of changes and synchronisation issues with the original and the amount of work needed to create the resource is the same as creating two separate and disjoint files – time and effort are not saved. Dan Connolly proposed a system called HyperRDF in which HTML is used as the conduit to use XSLT to transform information into RDF. However, HyperRDF cannot be validated since the head element does not allow an ID attribute [7]. Augmented Metadata for XHTML is an implementation that allows Dublin Core metadata to be incorporated in Web pages in a way that is compatible with today’s Web browsers. The basic premise is that one can take the profile attribute to be a global namespace prefix for all of the `rel / meta` and `name` attributes throughout the document. This approach is mainly for those authors that want to use a simple mechanism for producing RDF from their XHTML. It is ineffective from the point of view of anyone that wants to randomly extract RDF from XHTML, since one cannot tell whether the author wanted the assertions to be converted into the triples produced by the algorithm or not [2].

Finally, the most recent thinking on the subject comes in the form of GRDDL (Gleaning Resource Descriptions from Dialects of Languages). This work is being undertaken by the W3C Web Co-ordination Group and is a mechanism for encoding RDF statements in XHTML and XML. GRDDL shares some commonalities with HyperRDF and works on the principle that the HTML specification provides a mechanism for authors to use particular metadata vocabularies and thereby indicate the author’s intent to use those terms in accordance with the conventions of the community that originated the terms. Authors may wish to define additional link types not described in this specification. If they do so, they should use a `profile` to cite the conventions used to define the link types. GRDDL is one of these profiles which uses XSLT to transform a page to an RDF description.

3.1 Why GRDDL Doesn’t Work For Us

Our research centres around both the designer and the user. 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 resounding message we receive 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."

Many web designers move from print media to web design and this pre-gained experience in creating static designed artifacts forces them to see design as fixed and immovable once created. A designer creates and controls the development of what is in effect a piece of art and therefore once created should not be changed or violated. It can be difficult to convey that users often require web pages to adapt to their needs, and the fact that this sometimes goes beyond art.

We suggest that designers need a lightweight no-frills approach to include semantic information within XHTML documents; in effect the presence of the semantic information should be seamless indivisible and have a low cost design overhead.

4 Low Cost Semantics

Our system is in reality a process for associating ontology concepts with instances encoded within XHTML pages. Currently, presentation and meaning are separated as we can see in figure 3. The CSS and ontologies are mostly manual created while the instances, the XHTML, and the semantics associated with instances are created either manually or are automatically generated. The CSS and XHTML are assembled on the client and joined by the browser functionality while the RDF and ontology are used by either automated agents or RDF 'feed' readers (for use by humans). We suggest that this type of separation is both unhelpful, damaging, and counter to the Semantic Web vision. With Tim Berners-Lee's desire to describe resources (many on the web as standard XHTML documents) more fully the division between the web and the semantic web will increasingly become a hindrance. Although users can currently interact with web resources, and agents are starting to interact with semantic resources, surely progress should be made towards a joining of the two. We believe there should be just one web where semantics, presentation, and information are conjoined giving a holistic world-view.

Our system is a first step towards this. We suggest that meaning should be encoded within the elements of the XHTML and CSS along with ontologies which can be created as normal. Ontological concepts and properties are encoded into both the elements and attributes of the XHTML document and are used as identifiers within the CSS which link presentation to XHTML elements. Our system revolves around a software process (see Fig. 4) which converts an RDF-XHTML document into a series of instances and ontological descriptors for supply to the reasoner. Users view the document in a web browser as normal, however, browsers that are 'semantic-aware' can use the ontological information to provide more intelligent access to the instances of information than before. Currently, no browsers are 'semantic-aware' of our system except those with a system plug-in. However, all is not lost as RDF(s) can be generated by our process and inserted into the document such that RDF(s) aware browsers can take advantage of our system (as a 'Kludge').

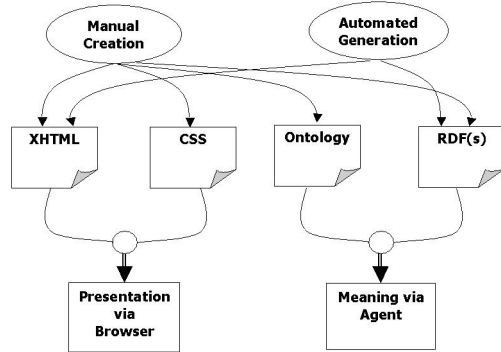


Fig. 3. As Things Currently Stand

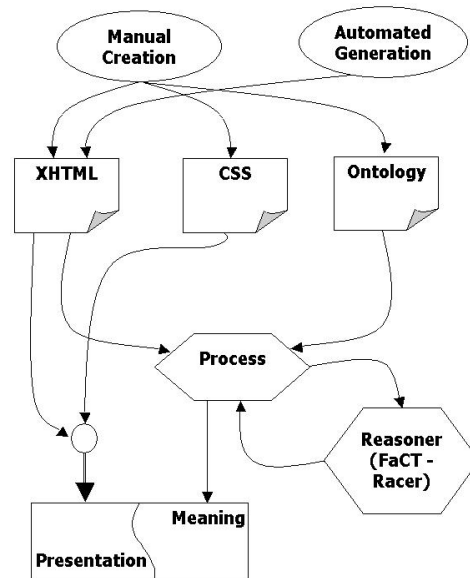


Fig. 4. Our Preliminary System

4.1 Encoding Ontologies in XHTML

Because we are suggesting a lightweight system our paradigm for encoding OWL Lite ontologies is simple, flexible, and without a semantic overhead. We use a trinity 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 (see Fig. 6).

Non Presentational XHTML Attributes We can leverage the implicit information contained in the names of XHTML elements if we have a corresponding ontology. Elements that are non-presentational (like `<address>`) can be used to encapsulate meaning within the page (see Fig. 6).

Individuals Unique individuals are defined by use of the anchor element where the `href` attribute is used to point to the URI or MAILTO of the unique item. If `http` / `mailto` are used then the link will be click-able. If `uri` is used then the link is not click-able (see Fig. 6).

We include namespaces in XHTML documents so that multiple ontologies can be used to describe one document. To implement this we use the `link` element of the XHTML header section.

```
<link rel="ontology" type="NAMESPACE" href="LOCATION"/>
```

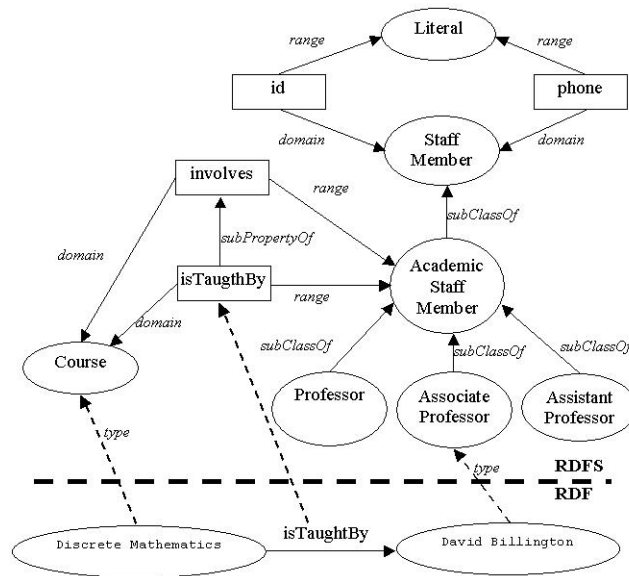


Fig. 5. RDF and RDFS Layers Taken from 'A Semantic Web Primer' Pg 84

```
<link rel="ontology" type="xmlns:owl"
href="http://www.w3c.org/2002/07/owl#" />
```

The first line represents the format the second an example. The `rel` attribute is always 'ontology' as this differentiates it from stylesheets and the like. Elements can be related to a namespace by using either the namespace identifier in the class attribute of the enclosing `div` element or by joining the namespace to the attribute name using an underline (`.`). The suggested approach provides a mechanism for encoding "lightweight" information. Of course this approach has its limitations – we can capture simple instantiation of atomic classes along with property assertions, but not richer assertions such as instantiation of arbitrary class expressions. We stress that this is not intended as a replacement for other representations but is a complementary mechanism. For example, we can still expect the class and property definitions in the ontology to be encoded using existing approaches such as RDF/XML.

Designers often want to adjust the visual design of a web-page without altering the actual meaning. We support theorise that this ad-hoc visualization can be handled by specialising ontological concepts with visual extensions if required.

5 Example

As a preliminary case study let us consider the simple ontology taken from 'A Semantic Web Primer' (in press) page 84 Figure 3.6 and recreated here for convenience as Fig. 5.

Let us now see how information culled from ‘David Billington’s’ Web page can be annotated (see Fig. 6) such that the semantics of the instance are available for inference following the ontology in Fig. 6. We can see that this information is just a general description of the course information. However, by adding a `div` element we enclose the information such that the enclosure implicitly relates any enclosed sub-elements. Secondly, we see that a `span` and anchor element are introduced to denote `Course` and `IsTaughtBy`. We can via the ontology now infer the conceptual range (using ABox reasoning via ‘Racer’) that discrete mathematics is taught by the associate professor David Billington, and what is more, so can assistive agents. This seems to represent what we want to say from a reasoning approach and when presented it is displayed correctly and with no additional overhead for the designer.

```
<div class="leftcolumn">
  <span class="course">Discrete Mathematics</span>, taught by
  <a class="IsTaughtBy" href="mailto:dbillington@uni.edu">
    David Billington</a>, is a second year course designed for Computer Science
    students who need a more formal mathematics training.
</div>
<div class="aboutnote">
  <a class="associate professor" href="mailto:dbillington@uni.edu">
    David Billington</a> is Associate Professor of Information Systems.
</div>
```

Fig. 6. XHTML Code

6 Why Does This Approach Aid Visually Impaired Users?

By knowing the meaning of the information that is being encountered visually impaired users can perform their own triage on that information. As we have previously mentioned, web pages are read from top left to bottom right. If there is a lot of information on the page then the user can get lost, disoriented, or at least frustrated with their progress through this information. By presenting the meaning of the information using standard transcoding methods, users can choose which information is important to them, not the visual designer.

7 Conclusions

Our system suggests a method of encoding lightweight mark-up into webpages to incur a low cost semantic benefit. With the meat of the information design being abstracted from the graphic / web designer the system has given a taster of how semantics can be represented within web-pages. Additionally, we also show how this can be achieved without incurring a significant overhead with regard to marking-up that semantic information and have it validate to XHTML 1.0 strict.. We propose that the inclusion

of semantic information directly into the XHTML is the only way to assist visually impaired users access web pages while not increasing or compromising the creation activity of authors and designers. Indeed we show the first stage in a more elaborate system to enable semantic information to be freely accessible by all users.

References

1. Web content accessibility guidelines 1.0, 1999. <http://www.w3.org/TR/1999/WAI-WEBCONTENT/>.
2. M. Altheim and S. B. Palmer. Augmented Metadata in XHTML, 2002. <http://infomesh.net/2002/augmeta/> - valid 2004.
3. T. Berners-Lee. RDF in HTML, 2002. <http://www.w3.org/2002/04/htmlrdf> - valid 2004.
4. 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.
5. 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.
6. A. Chieko and C. Lewis. Home page reader: IBM's talking web browser. In *Closing the Gap Conference Proceedings*, 1998.
7. D. Connolly. HyperRDF: Using XHTML Authoring Tools with XSLT to produce RDF Schemas, 2000. <http://www.w3.org/2000/07/hs78/> - valid 2004.
8. M. Dean and G. Schreiber. OWL Web Ontology Language Reference. W3C Recommendation, World Wide Web Consortium, 2004. <http://www.w3.org/TR/owl-ref/>.
9. A. G. Dodds. The mental maps of the blind. 76:5–12, January 1982.
10. 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.
11. Henter-Joyce, Inc. *Jaws*. <http://www.hj.com>.
12. N. Kew. Why Validate?, 2002. <http://lists.w3.org/Archives/Public/www-validator/2001Sep/0126.html> - valid 2004.
13. V. RNIB. A short guide to blindness. Booklet, Feb 1996. <http://www.rnib.org.uk>.
14. Sean B. Palmer. RDF in HTML: Approaches, 2002. <http://infomesh.net/2002/rdfinhtml/> - valid 2004.