

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

Opening up access to online documents using essentiality tracks

PLEASE CITE THE PUBLISHED VERSION

PUBLISHER

© ACM

VERSION

AM (Accepted Manuscript)

LICENCE

CC BY-NC-ND 4.0

REPOSITORY RECORD

Atkinson, Matthew T., Jatinder Dhiensa, and Colin H.C. Machin. 2019. "Opening up Access to Online Documents Using Essentiality Tracks". figshare. <https://hdl.handle.net/2134/4474>.

This item was submitted to Loughborough's Institutional Repository (<https://dspace.lboro.ac.uk/>) by the author and is made available under the following Creative Commons Licence conditions.



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

Opening up Access to Online Documents using Essentiality Tracks

Matthew T. Atkinson
Research School of
Informatics
Loughborough University
Loughborough, Leicestershire,
LE11 3TU
England, UK

M.T.Atkinson@lboro.ac.uk

Jatinder Dhiensa
Research School of
Informatics
Loughborough University
Loughborough, Leicestershire,
LE11 3TU
England, UK

J.Dhiensa@lboro.ac.uk

Colin H. C. Machin
Department of Computer
Science
Loughborough University
Loughborough, Leicestershire,
LE11 3TU
England, UK

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

ABSTRACT

This paper discusses extensions to the previously developed “essentiality and proficiency” approach to increasing usability and accessibility of websites. The existing approach is introduced, as is a new application in the processing of DocBook XML documents. The current principles are extended to make them more appropriate for increasing the usability of long documents. Techniques for allowing organisations to efficiently disseminate information based on the proposed application are discussed – increasing productivity for both non-disabled and disabled users.

Categories and Subject Descriptors

H.3.5 [Information Storage and Retrieval]: Online Information Services—*Web-based services*; H.5.4 [Information Interfaces and Presentation (e.g., HCI)]: Hypertext/hypermedia—*User issues*; I.7.2 [Document and Text Processing]: Document Preparation—*Hypertext/hypermedia*

General Terms

Documentation, Human factors, Standardization

Keywords

Accessibility, DocBook, Essentiality, Proficiency, Usability, Web, XML

1. INTRODUCTION

A significant amount of research has been carried out into providing access to websites for computer users with disabilities (such as sight loss). Some of this research [1, 2] focuses on finding ways to provide the user with only the information they need, in the format that is most suitable for them. This “universal design” [3, 4] style of approach

is applicable to providing access for non-disabled users who are, for example, using embedded devices or simply do not have time to view anything but the essential information a website has to offer.

We present some new applications of and extensions to this research that are relevant to both on and offline document access (via integration with the DocBook XML typesetting system). Utilising this approach will allow corporations and educational institutions to provide websites, training material and manuals that can be automatically filtered and rendered to meet the needs of users in various roles. This will negate both the cost of transcribing such materials into alternative formats and that of maintaining the transcribed versions. It will also allow new methods of enhancing productivity for non-disabled users.

2. CHALLENGES AND CURRENT WORK

Dhiensa et al [5] point out that the problems of web access (which is equally applicable to offline electronic documents in web formats) is three-fold:

- Information exclusion (the fact that either people do not have access to computers or, even when they do, the cost of assistive technology is too great or the sites they wish to access are not standards-compliant).
- Though standards and legislation are now in place, they do not guarantee usability [6].
- “Information Overload”. The problem, and potential ways to cope are described in more detail in [7] and [8].

Research on solving these problems has been targeted at different points within the lifecycle of websites. Mo-hamad et al [6] presents a solution that highlights standards-compliance and accessibility problems so that they can be fixed by site developers. Hanson and Richards [2] modify existing sites on-the-fly – enabling users to have sites rendered in their chosen format.

The Disability Rights Commission Report [9] shows that websites are 35% easier to use for *everyone* if they are accessible¹. This tells us that accessibility is a valid metric

¹i.e. compliant with accessibility [10] and other web standards

by which we can estimate the usability of websites for all people. This implies that organisations adapting the ideas presented here should be able to improve the productivity of all people wishing to access their website and web-based documents.

3. PRIMER ON ESSENTIALITY AND PROFICIENCY

To solve the problems discussed above, work carried out by Dhiensa [1] lead to the creation of the “Essentiality and Proficiency Tool”. The basic concepts of this tool are:

- A web page contains information that has varying levels of importance for its visitors.
- The author of a web page should attribute such levels of *essentiality* to sections of the page. Essentiality ratings (integer values from 1-10) surround page elements and can be nested (e.g. more essential elements/text can be found within less essential ones).
- When the page is rendered for a user, content is filtered based on its essentiality level in relation to the level that the user wishes to view the page at. Only content with an equal, or higher, essentiality rating is displayed.
- Users have varying preferences and needs for how they view the page – it should be rendered according to the *proficiency* of the output device and/or disability of the user.²
- Profiles can be used to store settings (essentiality levels, formatting preferences) for users of the system.
- The tool is designed to act as a proxy service, modifying pages as they are requested and transforming them according to the user’s profile. Later we will discuss the challenges associated with this approach and how they are inherently overcome in the problem domain this paper is concerned with.
- The two ideas of essentiality and proficiency are highlighted as separate factors involved in making sites accessible. These ideas fit well with DocBook XML documents, as we will discover later.

The rest of the work in this paper adapts and subsequently builds on these concepts.

3.1 Usage of Essentiality and Proficiency

The tool requires input in the form of a profile and essentiality markup on the pages it is to filter and transform. Therefore the process involved with its use is split into separate activities for page authors and users of the tool.

3.1.1 For Authors

Authors are required to indicate the essentiality levels of elements within the pages they create. Originally, tags had to be added manually to the source code of pages. An example web page with some essentiality markup is shown in figure 1.

²Rendering transformations may include the removal of pictures for blind users, use of high-contrast colours for the vision-impaired and truncation of pages for display on mobile/embedded devices.

```
<html>
  <head>
    <title>Test Page</title>
  </head>
  <body>
    <h1>Essentiality Tag Example</h1>
    <p>This paragraph is not tagged.</p>
    <essn level="6">
      <p>Both this paragraph...</p>
      <p>...and this one are marked as
        level 6.</p>
    </essn>
    <essn level="8">
      <p>This one, however, is even more
        important.</p>
    </essn>
  </body>
</html>
```

Figure 1: A sample web page with essentiality tags.

Recently a Mozilla Firefox plugin was created [11] that allows this task to be completed via a GUI interface. Tags are still added to the document’s source, but this process is hidden from the author – making it usable by a wider range of authors.

The GUI interface allows authors to obtain feedback on their markup in the following ways:

- Parts of their pages marked up as essential are highlighted. The colour varies according to essentiality rating.
- They can request that the page be displayed at a given essentiality level. All information marked up at this level, and below is then displayed.

It should be noted that the adoption of essentiality tags by the World Wide Web Consortium (W3C) is an ultimate goal of the project. Before this happens, however, the use of microformatting³ has been used to ensure that pages are still regarded as valid. This technique will not be required by the process for marking up DocBook XML documents.

3.1.2 For Users

The system is designed to make its use as transparent as possible. In the current prototype, the user visits a web page where they can select and tailor their profile. The profile records their preferred levels of essentiality and proficiency settings.

From the profile page, the user can enter a URL to visit in a textbox. The system then retrieves and transforms that page, finally presenting it to the user. For more information, please consult the previous work [1].

3.2 Benefits

This system enables content producers to maintain only one version of their website – there is no longer a need to

³This is the practise of marking up the essentiality levels of document elements using their CSS “class” property, instead of using a dedicated essentiality tag. This ensures that the page is still regarded as valid and (ironically?) passes accessibility checks.

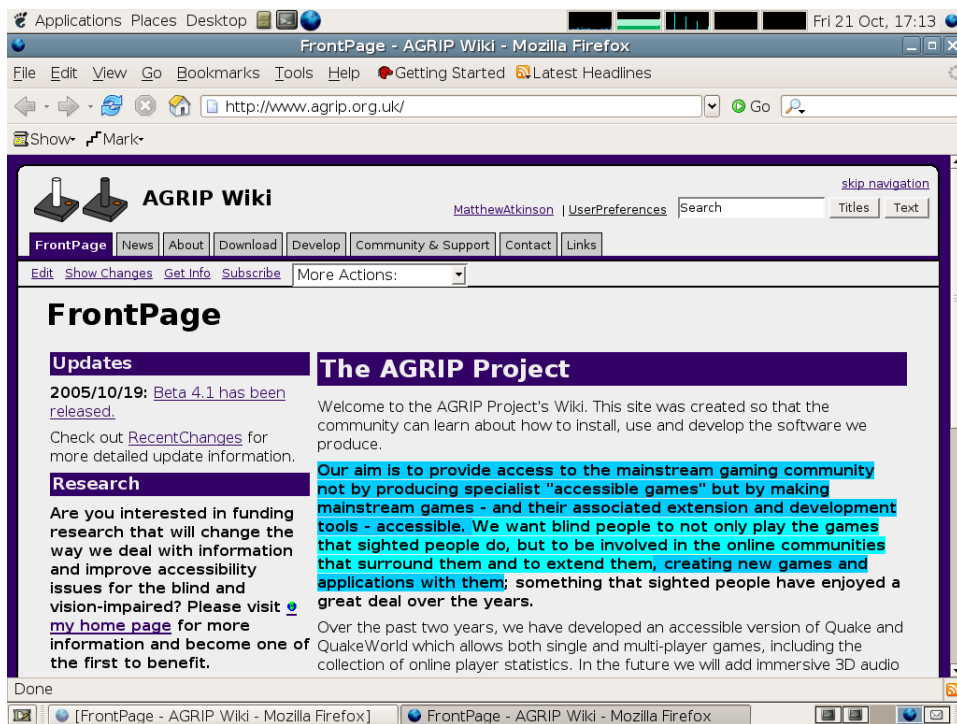


Figure 2: Screenshot of Essentiality Editor, with its toolbar buttons (“Show” and “Mark”) displayed, and some text that has been marked as level 6 (darker text background) which includes some nested text at level 8 (lighter text background).

create separate “accessible” or text-only variants. The Essentiality and Proficiency Tool automatically adapts sites to users’ needs with more flexibility than such alternative versions have historically provided – and at significantly lower cost.

Additionally, the work of Cheng provides a friendly, cross-platform, GUI-based method for authors to mark up their content.

The extensions and techniques proposed in later sections of this paper inherit the process model and general approach used by Dhiensa et al, but implement them in the context of automatic translation of documents written in the DocBook XML typesetting system.

3.3 Limitations of the Current System

Two main limitations of the current system are currently being investigated. These are:

Dynamic Content — A great deal of web content is generated dynamically by scripts (which may include the output from database lookups). A way to mark up this content with the appropriate essentiality levels should be devised.⁴

⁴It is theorised by the authors that there are two main type of script-generated information: *facts* gleaned from a database and *prose* gleaned from a content management system (which could also be resident in a database, but this is incidental). Any prose may be treated in the same way as the current static content is (i.e. must be marked up as it is written). Facts (such as a price list) may contain entirely essential information and thus another way of navigating them must be developed. Research into these issues is ongoing.

Scalability across User Types — Some documents (especially corporate guidelines, user/developer manuals and “procedure documentation”⁵) are aimed at groups of people with a number of different roles. These documents inevitably contain information of varying importance to people in these differing roles. There is currently no way to denote this difference in interest.

This paper is mainly concerned with the latter of these limitations, which will be revisited shortly.

4. DOCBOOK XML

So far the work described has been applied to make websites accessible. However, it could equally be applied to documents that use web standard formats. DocBook XML is a Document Type Definition (DTD) and set of output filters designed for the creation of technical documentation. It is employed by many companies and institutions in the creation of their internal and external documentation.

The ethos of DocBook is to:

- Provide separation of content from formatting.
- Be as extensible as possible, allowing users to customise both output filters and the DocBook DTD.

⁵“Procedure documentation” refers to the type of documentation that is created within corporations to describe business processes, health and safety guidelines, software test procedures and so on. It often targets roles at different levels within an organisation, so finding relevant information in them can be difficult.

```

<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE book PUBLIC
  "-//OASIS//DTD DocBook XML V4.4//EN"
  "http://www.oasis-open.org/docbook/xml/
  4.4/docbookx.dtd">
<book lang="en">
  <bookinfo>
    <title>Very Simple Book</title>
    <keywordset>
      <keyword>Example</keyword>
    </keywordset>
  </bookinfo>
  <part id="intro">
    <title>Introduction</title>
    <chapter id="intro-background">
      <title>Background</title>
      <para>Some text...</para>
      <section>
        <title>Nested Section</title>
        <para>Some more text...</para>
      </section>
    </chapter>
  </part>
</book>

```

Figure 3: A very simple DocBook XML Document. The ‘\’ is used as a line continuation character.

What makes DocBook particularly suitable, as far as Essentiality and Proficiency is concerned, is its close integration with web and on-the-fly translation technologies.

Previous work ([12] and the use of DocBook for creating accessible lecture notes) found that DocBook provided a means to generate output in a number of useful formats from one source file. Most notably: the effort required to customise the existing XSLT code that produced (X)HTML output, in order to improve accessibility, was minimal. PDF and RTF output can also be produced easily.

5. APPLYING ESSENTIALITY TO DOCBOOK

Before we propose extensions to the existing approach, we will describe how it was implemented for DocBook documents. This demonstrates how useful the principles can be when applied in this way and provides a stepping stone for extending them. A very simple DocBook document is shown in figure 3. “Essentiality” is currently a one-dimensional grading of importance, as shown in figure 1.

In a simplistic essentiality-filtering algorithm, the essentiality level of the `<essn>` tag is checked against the user-defined filter level. The idea is that all elements which have an equal or higher rating to that selected by the user should be displayed (i.e. we create a high-pass filter). The content bounded by the `<essn>` tags should be displayed only if this condition is met.

However, we must beware of the situation where a more essential piece of information is nested inside a less essential section. If the essentiality level of a given element is not sufficient to get through the filter, but it contains a nested element that *would* get through the filter, we must display the nested element.

The following DocBook source could produce misleading output:

```

<essn level="7">
  <para>Under no circumstances must any
  employee:</para>
</essn>
<essn level="5">
  <itemizedlist>
    <listitem>
      <para>
        Stay in the building after the alarm
        has sounded. <essn level="9">Leave the
        building immediately if there is a
        fire.</essn>
      </para>
    </listitem>
    . . .
  </itemizedlist>
</essn>

```

Sample output from the above source (if user-specified essentiality level is 6-9):

```

<p>Under no circumstances must any employee:</p>
<ul>
  <li>Leave the building immediately if there
  is a fire.</li>
  . . .
</ul>

```

Figure 4: An example of the importance of context.

5.1 The Problem of Context

A further issue is that of information being taken out of context due to the way it has been filtered. Consider the situation where one vitally important sentence has been displayed that was in the middle of a less important paragraph. If the parent paragraph is filtered but the sentence is displayed (due to the situation described above), then it could be taken out of context and have potentially serious ramifications – see figure 4 for an example.

There are two ways to ensure this does not happen:

- Authors should ensure essential information is marked up with the appropriate contextual information (e.g. the entire bullet point should have been at level 9 in our example).
- As the above may not always be appropriate, the algorithm should ensure that the output makes breaks in context clear to the user.

6. PROPOSED EXTENSIONS TO ESSENTIALITY MARKUP

Adapting the ideas of essentiality and proficiency to the world of DocBook documentation could further improve the accessibility and usability of such documents. However, the scalability of the current system can limit its usefulness, especially when long documents, aimed at people in multiple roles, are concerned.

```

<essn track="manager" level="3">
  <p>...</p>
</essn>
<essn track="developer" level="4">
  <p>...</p>
</essn>

```

Figure 5: Simple example of Essentiality Tracks.

```

ProcessEssnTag(tag) is
begin
  if tag track = global track
  if tag level >= global level
    display elements inside tag
  else
    if nested tag
      print 'Context Break'
      ProcessEssnTag(nested tag)
      print 'Context Break Finished'
    end if
  end if
else
  if nested tag
    ProcessEssnTag(nested tag)
  end if
end if
end

```

Figure 6: Simple algorithm for selecting elements to process from the correct essentiality track. Also includes code to add context break warnings where necessary (see 5.1).

6.1 Essentiality Tracks

We propose extending the current tags to create a number of *tracks* through the document; each denoting how important the information is for people of different roles and/or groups of people. Figure 5 shows a simple example of the tags that may be used for this.

In the case that a certain element, or set of elements may be of interest to multiple roles, they can be marked up with nested `<essn>` tags. This does not affect the formatting of the document and is semantically equivalent to marking the essentiality up in parallel⁶

Detecting if a sub-tree of an `<essn>` tag should be processed in a multitrack document is slightly more involved than for the single-track documents discussed above. An example algorithm is given in figure 6. The algorithm contains a test to ensure that the `<essn>` tag is concerned with the track that the user chose and includes extra code to deal with contextual breaks as described in section 5.1.

6.2 Track Coordination

The extensions proposed above have been informally tested, however further improvements may be necessary before the system could be deployed for very large documents. This section lists some further improvements that should be considered in future work.

⁶Suggestions on how this situation could be made more intuitive are given later.

Error Checking — By requiring authors (or their authoring tools) to declare all tracks for the document in its header, the `<essn>` tags may be validated to ensure they refer to a track that exists.

Track Grouping — It may be useful to group tracks at certain points to make the markup more readable and ensure it is easy to update in the future. One way to provide grouping of essentiality tracks extends the above suggestion to include a section in the document header that links tracks to their groups.

Track Relationships — Providing a means to express how tracks are related could aid marking up and revising the markup in long documents. Allowing authors to use expressions which link the ratings between different tracks could allow them to more effectively partition the document. For example, a simple “invert” rule could be used to ensure that anything marked up as very important for developers should be marked as of little importance for users. This feature could introduce too much complexity, however, so its usefulness will need to be tested.

Overview — Future essentiality editors could include a visual “map” of where different tracks can be found. This would present the author with a quick overview of how the document is structured.

Figure 7 gives an example of how the first two suggestions above may be expressed in terms of document markup. It presents a more intuitive way of marking up an element or sub-tree’s essentiality across different tracks – simple `<essn>` tags have to be nested; `<grade>`s can be applied in parallel.

6.3 Applying Essentiality Tracks to DocBook in the Real World

As discussed, DocBook is designed with customisation in mind. It is trivial to edit the XSLT code so that truly accessible (X)HTML output is produced⁷ so this will not be discussed further.

Following basic accessibility improvements, proficiency may be implemented by further enhancing the stylesheets to enlarge the fonts, choose user and/or device-compatible colours and reorder navigation links according to the values specified in the user’s profile.

Adding support for our extended essentiality tags involves editing the DocBook DTD⁸. As the DTD itself is an XML document this is also a relatively easy task. However, it should be borne in mind that adding elements to a DTD (as opposed to taking them away) results in incompatibility with the original format because the new standard is a superset of the original one. This is not a problem when an organisation uses a format internally, but it is still important to submit these changes for inclusion in future versions of the standard if they are to be widely promoted and used.

⁷Replacing all layout tables with CSS2, as the AGRIP Documentation Project have done, for example.

⁸In the experiments conducted for this paper, the DTD was not edited and the “-novalid” option was passed to the XML processor to disable validity checks. This allows quick tests to be made but is not as robust as editing the DTD.

```

<book>
  <bookinfo>
    <title>Essentiality Tracks Example</title>
    ...
  </bookinfo>
  <essninfo>
    <tracks>
      <track id="manager" display="Manager"/>
      <track id="dev" display="Developer"/>
      <track id="user" display="User"/>
      <track id="qa" display="Quality Assurance"/>
      <track id="test" display="Tester"/>
    </tracks>
    <groups>
      <group id="testgroup">
        <member>test</member>
        <member>qa</member>
      </group>
    </groups>
  </essninfo>
  ...
  <chapter>
    <title>...</title>
    ...
    <essn track="manager" level="3">
      <para>...</para>
    </essn>
    <essn>
      <essngrades>
        <grade track="manager" level="4"/>
        <grade track="developer" level="9"/>
        <grade group="testgroup" level="6"/>
      </essngrades>
      <section>
        <title>...</title>
        <para>...</para>
        <essn>
          <essngrades>
            <grade track="manager" level="6"/>
            <grade track="user" level="3"/>
          </essngrades>
          <para>...</para>
        </essn>
        <para>...</para>
        <para>...</para>
      </section>
    </essn>
    ...
  </chapter>
  ...
</book>

```

Figure 7: Example of some more practical DocBook DTD Extensions. Being able to group tracks could make marking up long documents considerably easier. This scheme also provides some built-in error-checking.

Browser Menu and Toolbar	
Document Retrieval Form	
Your role	<input type="text"/>
Essentiality	<input type="text"/>
Profile	<input type="text"/>
Custom font	<input type="text"/>
Custom colours	<input type="text"/>

Figure 8: Sample webform used to select essentiality track and level.

6.4 Online Transformation

As discussed, many different output formats can be generated from a DocBook document. This may be done both on and offline, via the use of XSLT and other related technologies⁹. As an example of where on-the-fly processing may be useful, consider an organisation publishing the documentation for some software it has developed for its employees.

In such an example, people in many different job roles could be interested in the documentation; ranging from managers to users and developers. There would likely be a number of essentiality tracks in the source files of the document. It is very likely that users would have different proficiency requirements too. If we consider the combination of essentiality track, filter level and proficiency requirements, we realise that a vast amount of output would have to be developed to cater for the needs of everyone in advance.

An efficient way to disseminate this documentation would be to have readers fill in a (very) short webform such as that shown in figure 8. This would select the parameters for the transformation, which would then occur in real time, generating (X)HTML (or PDF/RTF) output for the reader's chosen essentiality track, level and proficiency profile. There may be good reasons for having this generation take place either on a server running Apache Cocoon or within the user's web browser.

6.5 Challenges of the Proxy Approach and How they are Alleviated

Though they promise great benefits (almost no modification necessary on client machines, centralised administration and upgrades), traditional proxy services may run into problems. Some popular criticisms (including those highlighted by Hanson and Richards [2] and Mohamad et al [6]) are:

- The method they use to transform pages may be overridden by features of the page (such as the use of embedded CSS instead of external stylesheets).
- Bandwidth constraints may slow the system down.
- Such systems often have to “undo” the inaccessible work done when the site was created; this could be inefficient (at least from a design elegance point of view).

⁹There is an older SGML standard and DSSSL stylesheets, but we do not consider them here

- Proxies have to be tolerant of non-standards-compliant sites.
- Most web sites are not annotated to provide extra accessibility information.

These problems are not present (or are averted) within the proposed DocBook transformation system for the following reasons:

- All style information is embodied in the XSLT and external CSS stylesheets (in the case of generated (X)HTML).
- The system is envisaged to be deployed within a web service such as Apache Cocoon [13] (which could cache the results) on an organisation's LAN. However, there is nothing to stop the process happening client-side, within the user's browser (both Mozilla Navigator and Mozilla Firefox support the required XSLT standards). Client-side translation may be of use when server resources are at a premium.
- No “undoing” work is necessary; only the transformations appropriate to the user are carried out.
- No tolerance of non-conformance is required as all DocBook XML documents supplied to the system should be valid – they are checked for validity before all other processing takes place.
- Though there is no absolute need for annotation, we have already established a business case for it (increased productivity and usability for all), so it need not be seen as an extra chore.

The proposed approach works because an organisation would have *control* over the standards and transformations in use at all stages – this is rarely the case when trying to improve the accessibility of third-party websites.

7. CONCLUSIONS AND FUTURE WORK

Use of essentiality tracks and the above techniques for transformation are proposed as an efficient and effective method for organisations such as companies and educational institutions to disseminate their materials¹⁰. There is still a lot of potential for future work, however:

- Other output formats than traditional HTML and PDF could have the principles of essentiality and proficiency applied to them. For example: translation of DocBook into Braille is not impossible and – given the essentiality filter – could be an ideal format for accessible hard copies.
- The semantics of essentiality tracks could also be improved. The examples given in this paper indicate that they could be of great benefit to users, however the task of marking up content for the authors could be made easier. Section 6.2 has already suggested some work that could be carried out to further improve the proposed system.

¹⁰Though in the case of lecture notes and similar material, the tool should be used to improve general accessibility, not to give students direct answers (thus reducing their ability to think critically) – research into the balance between these effects should be carried out.

- The essentiality editor [11] should be extended to support tracks.
- More tests should be carried out with respect to the most natural scale for essentiality. The one used here is taken from previous work (levels 1-10, with 10 representing the most essential information) but it may be that a different scale is more useful.

The ideas proposed so far are being formally tested. The community of AGRIP [14] is participating in these tests as (a) a large amount of DocBook documentation is already used within this project and (b) as most users are blind, the effects on accessibility as well as general productivity could be assessed. So far, informal user feedback has yielded positive results.

Additionally, our future research will be touching on some of these areas.

8. ACKNOWLEDGEMENTS

This work was made possible by the sponsorship of the Grundy Educational Trust and Loughborough University.

9. REFERENCES

- [1] Jatinder Dhiensa, Colin Machin, Francesca Smith, and Roger Stone. Optimizing the user environment: Leading towards an accessible and usable experience. In *Accessible Design in the Digital World Conference 2005*, 2005.
- [2] Vicki L. Hanson and John T. Richards. A web accessibility service: update and findings. In *Assets '04: Proceedings of the 6th international ACM SIGACCESS conference on Computers and accessibility*, pages 169–176, New York, NY, USA, 2004. ACM Press.
- [3] NC State University. What is universal design? http://www.design.ncsu.edu:8120/cud/univ_design/princ_overview.htm, April 1997.
- [4] Trace Centre. General concepts, universal design principles and guidelines. http://trace.wisc.edu/world/gen_ud.html, 1971.
- [5] Jatinder Dhiensa, Colin Machin, and Roger Stone. Assistive technology: Going beyond the disability. In *Proceedings of Include 2005*, April 2005.
- [6] Yehya Mohamad, Dirk Stegemann, Johannes Koch, and Carlos A. Velasco. *imergo: Supporting Accessibility and Web Standards to Meet the Needs of the Industry via Process-Oriented Software Tools, Lecture Notes in Computer Science*, volume 3118, pages 310–316. Springer-Verlag GmbH, January 2004.
- [7] Starr R. Hiltz and Murray Turoff. Structuring computer-mediated communication systems to avoid information overload. *Commun. ACM*, 28(7):680–689, 1985.
- [8] Hal Berghel. Cyberspace 2000: dealing with information overload. *Commun. ACM*, 40(2):19–24, 1997.
- [9] Disability Rights Commission. Formal investigation report: web accessibility. <http://www.drc-gb.org/publicationsandreports/report.asp>, April 2004.
- [10] Web Accessibility Initiative. Web Content Accessibility Guidelines (WCAG) 1.0. <http://www.w3.org/TR/WCAG10/>, May 1999.
- [11] Yangfan Cheng. Essentiality Mark-Up Editor to Address the Issue of Inaccessibility Using XUL. Master's thesis, Department of Computer Science, Loughborough University, September 2005.
- [12] Matthew T. Atkinson. AGRIP Documentation Project. <http://docs.agrip.org.uk/>, 2004.
- [13] Apache Software Foundation. Apache Cocoon. <http://cocoon.apache.org/>, 2003.
- [14] Matthew T. Atkinson and Sabahattin Gucukoglu. Accessible Gaming Rendering Independence Possible. <http://www.agrip.org.uk/>, May 2003.