An integrated approach to preparing, publishing, presenting and preserving theses

Peter Sefton

Distance and e-Learning Centre, University of Southern Queensland, Toowoomba 4350, Queensland, Australia

Abstract

This paper describes progress on a project funded by the Australian government to create Free software; the Integrated Content Environment for research and scholarship (ICE-RS). ICE-RS is a multi-faceted project which will add value to finished theses by making them available in both HTML and PDF, as well as providing a mechanism for packaging multimedia theses. The project will also concentrate on providing services for thesis production, with version control, automated backup and collaboration services.

The paper begins with the established content management system that is the basis for the project, ICE, originally developed to create courseware packages. ICE includes distributed, version controlled collaboration, using word processing software and works on multiple platforms, with standard document formats. We survey other approaches to content authoring and publishing for ETDs.

We showcase exploratory work on integration of the thesis writing process with Institutional Repository software including publishing theses in both PDF and HTML with preservation and descriptive metadata. The presentation will include demonstrations of thesis production at all stages of development from proposal to completion.

In a more speculative vein, we will discuss opportunities for institutions to provide new levels of support for candidates via automated thesis "dashboard" progress reports, supervisor and examiner annotation and comment and support for copyright considerations as early as possible in the process.

Introduction

This paper describes progress made on a project funded by the Australian government to create a free (as in open source) software application and associated documentation. The project is known as the *Integrated Content Environment for research and scholarship* or ICE-RS. The project is tasked with creating and/or documenting software and work practices that allow academics and students writing-up research to create documents, collaborate, manage, publish and deposit their work in repositories. An overview of the project, derived from the successful proposal document is available on the ICE website (Sefton 2006b) .

ICE-RS is supported by the Systemic Infrastructure Initiative as part of the Australian Commonwealth Government's Backing Australia's Ability – An Innovative Action Plan for the Future (http://backingaus.innovation. gov.au).

The project proposal describes the key aim; to provide more flexible content than having all documents delivered in PDF (Adobe 2007) while also making the written dimension of

scholarship more efficient and the result more sustainabl e:

> In the institutional repository world, the Adobe PDF format is currently the expected norm for document delivery.
>
> Even though institutional repositories are web-based systems most content is not available in the native web format, HTML. HTML is more usable and flexible than PDF in many situations, allowing users to skim and sample content more easily that PDF. PDF, on the other hand, is a good solution for printing long documents and can be configured to make reading even book-length content a comfortable experience.
>
> So why is it not the norm for repositories to offer both PDF and HTML?
>
> It is because many of the widely used tools used for creating and storing research do not allow for reliable, automated production of HTML and PDF versions, and repository solutions are not geared to delivering content in flexibl e ways.
>
> (Sefton 2006b)

Introduction to the ICE System

What is ICE?

In a broad sense ICE is a content management system, or *CMS*, but unlike may such systems it is not a simple online web site building tool, having a number of specialized features. It is described in detail in an earlier paper (Sefton 2006a), but will be outlined here.

- **Users work in a word processor** to create content that can be delivered both on the web and in print. Word processors provide a number of features to manage long documents that are not available in online editing applications at present; this is discussed in detail below.
- **ICE maintains detailed version control** over all objects using the Subversion (Collins-Sussmann, Fitzpatrick, and Pilato 2004) version control system, with an easy-to-use interface that removes the complexity inherent in distributed version control.
- Rendering to HTML for **delivery on the web is a completely automated process** driven by the use of word processing styles in ICE documents. The ICE styles are designed to be general purpose and easy to use. The ICE approach is contrasted with other approaches in the section on Related work .

The ICE client is a software application which runs as a web server on the client machine, rather than a central server, accessed through a standard web browser. It checks-out and manages a copy of he user's workspace, consisting of documents, readings, bibliographies and small data sets, using the Subversion revision control system. The resulting working copy resides on the user's machine; where they can create edit and manage content offline and synchronize with the server copy when online.

At time of writing a server-based interface to the workspace under development by the ICE team. This will supplement the existing client-based ICE system, allowing users to navigate their workspace from a web browser and download documents to work on them in a word processor.

This paper will concentrate on the use of ICE-RS for Electronic Theses and Dissertations, ETDs, which is currently just beginning, but will emphasize that ICE is designed to be useful for much more that theses or even just for research outputs. As well as producing course content it has been used as a general-purpose intranet, website management tool and for blogging. The generalist approach is a fundamental foundation principle for ICE, allowing users to use the same tool for as many different authoring processes as possible.

ICE Status

ICE is now used to maintain over 100 full-length university courses with a rapidly growing user base which now exceeds 80 users across the University of Southern Queensland's three campuses. While these numbers do not represent much impact beyond the university they represent a major change in course authoring for the university.

Related work

A previous paper on ICE covers related systems (Sefton 2006a), mainly systems which use word processor input to an XML publishing system. In summary, ICE is more automated than most such systems which rely on word processor input, while it is much simpler and has less structure than typical XML document publishing systems, having been designed around the limitations of word processors rather than an idealized structure.

Of particular interest in this discussion is the work that has been done at Humbolt University on an XML publishing system (Müller and Klatt 2005; Dobratz 2005) and the DiVA system at Uppsala University (Müller, et al. 2003; Müller, et al. 2003). Both applications use an XML schema and a XML tool-chain to render content, with input either via an XML editor, or via word processing templates.

Advantages of a dedicated schema include:

- Detailed control over rendering.
- Automated validation using one of the XML schema languages (Bonifati and Lee 2001) and an XML editor.

In a system where authors are using a word processor rather than an XML editor and an XML schema can introduce a new problem: it is not generally possible to take word processing documents and map them to a schema unless either the schema is very general purpose, or the word processor stylesheet is very complex, and thus prone to usability and mis-use problems.

Take the example of an ordering constraint over elements in an XML document. In a hypothetical ETD or paper schema there might be elements for an Abstract, followed by the body of the document. Using some formalism, the schema would say: “The abstract element

is followed by a body element", with further rules about what can be in the body.

In an XML editor, the application would guide the author, only allowing an introduction element following the abstract, but in a word processor the structure needs to be **implied**. The most effective method for implying structure is to use styles. A style is a named bundle of formatting that can be applied to multiple paragraphs or spans of text to format them in a consistent manner.

The screenshot, Figure 1 from Microsoft Word shows how this might be achieved. The style name of each paragraph is shown at the left of the screen.

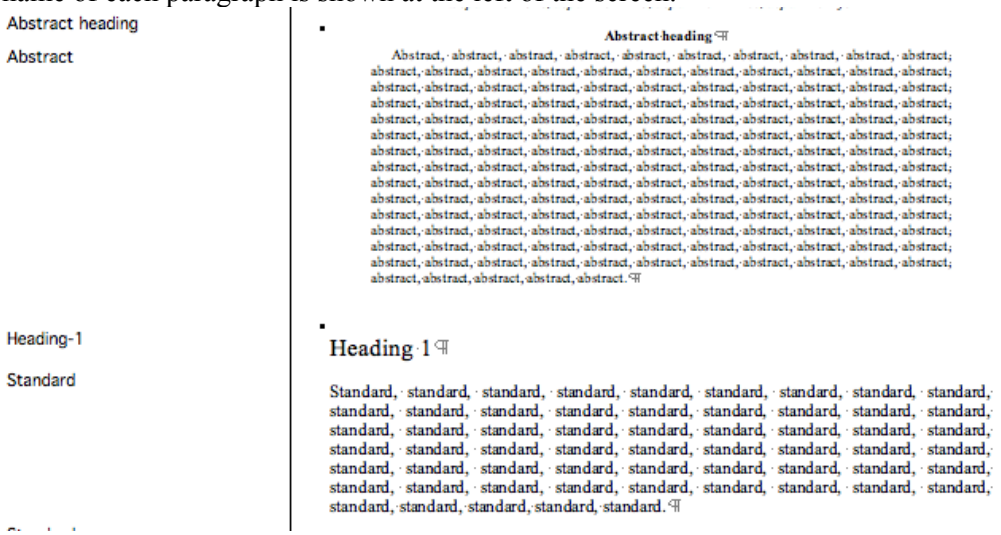

Figure 1:Screenshot showing styles used for editing a document in Microsoft Word. Style names show at the left of the screen

Using styles the structure is implied; a conversion system can 'wrap' everything between the `Abstract heading` style up to just before the `Heading-1` style in an `abstract` element.

```
<abstract>Abstract, abstract,

...

</abstract>

<body>

<section1>

<head>Heading 1</head>

<para>Standard, standard, standard, standard, standard,

...

</para>

...

</section1>

</body>
```

Typically of word processors neither Microsoft Word, nor the free OpenOffice.org Writer application have the ability to enforce ordering constraints in a reliable way and still maintain the flexibility for authors to navigate around their documents. There are some features in these applications designed for forms editing, but they are typically too rigid to use for long documents that need to be edited in a fluid way. Likewise, experience has shown that trying to control document structure via the use of macros is both expensive to develop and error prone.

For a case like the one illustrated, it is not likely that authors would change the template, as the abstract occurs on the first page, but for constraints on other elements, such as a requirement that lists must have two or more elements, or special sections, then it becomes very difficult to enforce rules, and thus manual validation and conversion steps are typically required.

Word does have an schema-validated XML mode, but this is not suitable for structuring long documents. When questioned, the Microsoft staff member responsible for the XML support in Word stated that the feature is intended for embedding small amounts of structure data in a generic word processing documents. (This comment is from a conversation between the author and Brian Jones of Microsoft in answer to questions about whether there existed an XHTML editing mode for Word. There does not.):

> Word is about editing rich documents, so Word is more loose, and the XML is good for

> adding some additional structure for richer semantics.
>
> (Jones 2005)

The key word here is 'loose'; word processors are permissive about what can be placed where, which is why the mapping to XHTML performed by ICE can work reliably while more ambitious XML authoring using word processors typically requires more intervention.

For *tighter* editing, using an XML schema and an XML editor will lead the author through the document structure. But it may not be possible to persuade general users to work with an XML editor. At the University of Southern Queensland, experience with an XML based course editing system has shown that only a handful of academic staff ever embraced the system and considerable costs were incurred migrating documents from word processing or desktop publishing packages, as there were constraints in the schema design for the system that were not met by existing documents or new documents created in word processors (Sefton 2006a).

The Humbolt team report (Dobratz 2005) that there was a manual step for converting word processing documents to XML to overcome the inherent looseness of word processing documents.

The DiVA system has a much more detailed metadata system, than ICE, which at present has only rudimentary metadata handling. See the section on Planned collaboration services: Workflow 2.0, below for some details of planned support for metadata in ICE.

Another team, working at McGill University (Park, Zou, and McKnight 2007), report some statistics on conversion for theses prepared in various word processors and using LaTeX. Conversion to the TEI (Sperberg-McQueen and Burnard 2002) schema was automated using a commercial product:

> To check text loss during conversion, 15 converted theses were randomly selected and reviewed. All showed some degree of textual information loss:
>
> - Less than 1 percent loss – five theses.
> - Less than 5 percent loss – four theses.
> - Less than 10 percent loss – two theses.
>
> Of the remaining four theses, these lost 12 percent, 19 percent, 23 percent and 40 percent of the text. We assume that anything less than 100 percent capture is not acceptable in the final ETD systems. To ensure perfectly formatted theses, the following problems need to be resolved: full linking to external files; successful conversion of formulae; successful conversion of endnotes; and dealing with missing table structure information. Another concern is who should carry out the conversion. We felt that the conversion to XML should be performed by the library in the initial stage of the production, not by students, because the principle that no additional burden should be placed on students is primary to us.

The TEI application is a generic schema, but the problems reported in this study point to the complexity of designing a word processing stylesheet and configuring a conversion system that can not only not lose text, but correctly map it to a useful structure.

A more general approach, as taken by the ICE system offers different advantages. The ICE system is rigorously tested to make sure that there is zero loss when documents are converted, including objects such as equations and images. The DiVA system, by contrast was not as easily able to deal with equations because of its use a custom XML schema (Müller, et al. 2003).

The ICE word processing template is designed to offer styles that correspond to XHTML structures, with no further constraints. For example, the heading style h1 will map to the HTML element h1. A previous paper on ICE (Sefton 2006a) describes the styles.

There are no ETD or other specific elements; all kinds of content are mapped to HTML, but the class attribute in HTML may be used for domain specific semantics. Any ICE style may be sub-classed by adding a hyphenated suffix. Thus a style `p-example` would result in a XHTML element `<p class="example">...</p>`.

ICE does require that users undergo some training, usually about three hours for a coursware author. It is commonly reported that without training styles do not get used widely or consistently. For example this work (Sørgaard and Sandahl) on the way styles are used in a corpus of documents:

> We have developed a classification of problems with paragraph styles. The classification appears to cover the problems we have identified but one category, “Overlooking styles”, dominates quantitatively.
>
> In a sample of documents paragraph styles appear to be little used. Even in documents where use of paragraph styles would make sense (for example, because of planned publication in WWW), there is little use. These practices will make it difficult to benefit from the opportunities of digital documents, standard exchange formats, etc. One cannot realistically assume that current practices in use of word processing provide a good basis for electronic publishing.

ICE transcends general practice in word processing, and avoids the problem of “Overlooking styles” by not only providing users with a style sheet, but by **providing a rapid-feedback mechanism** via fully automated, well tested conversion to HTML based on styles. For documents destined for the web a preview of the document formatted in HTML is always a click away. The screenshot in Figure 2 shows this paper, while in preparation. The upper window is NeoOffice Writer, a version of OpenOffice.org Writer for the Apple OS X platform, while the lower is a the Firefox web browser, rendering the document.

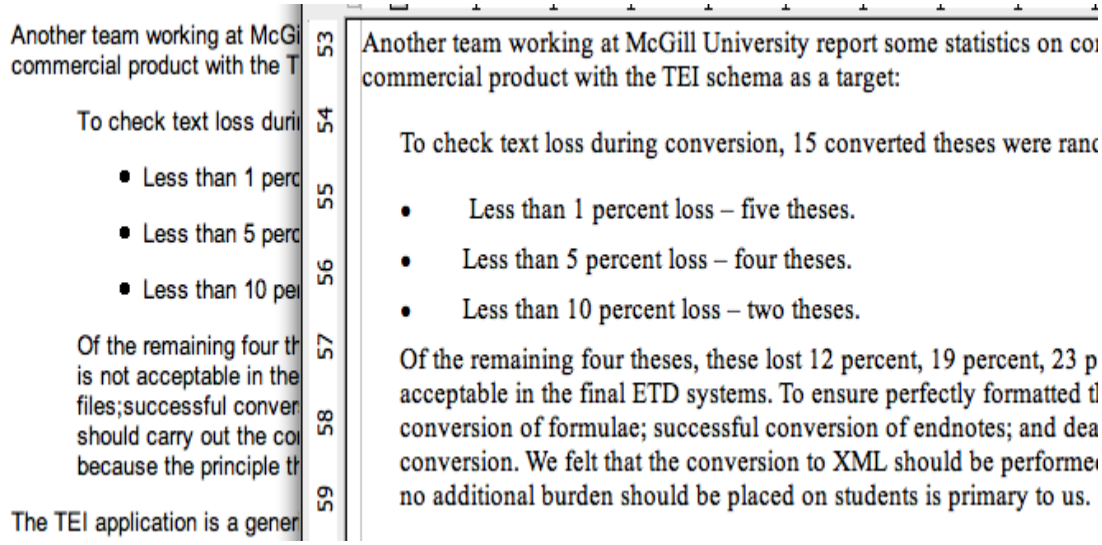

Figure 2: A screenshot of an ICE document being edited, with an automatically generated HTML view showing in the window behind.

The HTML code produced by ICE is valid XHTML, including correctly nested blockquote and list elements for the quotation, whereas a simple 'Save as HTML ...' from either Writer or Word produces code which may or may not look correct, but which does not contain the important semantics that this is a quotation.

So, while ICE may not be distinguishing between the major parts of an ETD – there is still important semantic information being captured. As an example of the use to which this might be put, the section on Planned services: dashboard reporting, below looks at potential reporting of things such as the average length and density of quotations in an ETD. A selective word-count such as this is possible if quotations are marked as such using styles.

Another system, with a direct relationshi p to ICE, is the Digital Scholar's Workbench, DSWB (Barnes 2006; Barnes 2007). The DSWB is a web application which uses the same set of styles as ICE. Initially the application was built to convert word processing documents into the DocBook format. Advantages of using DocBook included:

- Access to an existing free formatting tool chain that can render documents on the server, in contrast to the approach taken in ICE which is to use the What You See Is What You Get (WYSIWYG) capabilities of the OpenOffice.org writer.
- Preservation-quality XML using the DocBook schema.

Disadvantages include dealing with image formats: both Word and OpenOffice.org contain drawing and charting tools that produce formats that may be difficult to render without using the application that produced them, even if the formats are standards, and lack of fine-grained control over pagination when documents are rendered automatically.

In this section the ICE approach to preparing documents has been contrasted with some other heavier-weight approaches. ICE is a well tested, established platform for producing both HTML and PDF versions of document content, which can also package multimedia content, it seeks complete automation and ease of use over detailed structure.

The next part goes on to look at using ICE in a research context with the emphasis on its use for ETDs.

ICE for preservation

While the ICE system enables users to work either Microsoft Word or OpenOffice.org Writer, the conversion engine behind ICE uses OpenOffice.org as part of its transformation engine. Writer's native format the OpenDocument Format, ODF is an OASIS standard (OASIS 2005). Thus all ICE documents at one time or another go through the ODF format.

When Word documents are exported to a repository, it would be possible to add an ODF version to the repository for preservation format. Another possibility is that as the Office Open XML format (ECMA 2006), as used by Microsoft Word 2007 becomes more prevalent it will be possible to use it in preference to the current '.doc' format used by ICE. Plugin converters are available for versions of Word on the Windows operating system back to Word 2000.

Thus ICE users will have a choice of two open standard preservation formats for word processing documents. There is an ongoing debate between proponents of the two different formats as to their relative merits. The ICE approach, though is to use a subset of both formats which is compatible and interoperable, as shown in figure 3. ICE styles are designed to map to XHTML and to work in both OpenOffice.org Writer which uses ODF, and Microsoft Word, which can now use OOXML. The result is that using ICE provides interchangeable documents.

It has been argued (Barnes 2007) that both ODF and OOXML are unsuitable as preservation formats because they are compressed and thus subject to corruption. One remedy for this would be to store the uncompressed files that make up an ODF or OOXML package in a repository in an uncompressed state.

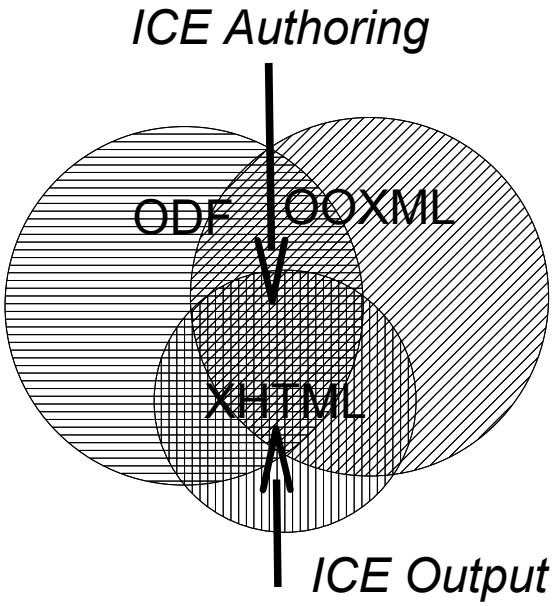

Figure 3: ICE authoring, via a stylesheet and some guidelines for authors, uses an interoperable sub-set of features from the Office Open XML (OOXML) and Open Document Formats (ODF), while targeting a large subset of XHTML.

Another issue is that XHTML (Pemberton 2000) may in fact be considered a viable preservation format. It is an open standard and is generic enough to describe ETDs and many other documents with some semantic encoding where appropriate. For this reason XHTML may be considered along with formats such as DocBook (Walsh and Muellner 1999). One argument in favour of XHTML as a preservation format for ICE is that if the input document is an ICE document, with styles that are similar in scope to HTML, then there is no important information lost by converting to XHTML compared to DocBook . Images and equations and other embedded objects remain a challenge for preservation in any case.

Using the ICE system for research and scholarship

The ICE-RS project runs from late 2006 to the end of 2007, so this paper is reporting mid-project; there are some concrete achievements to report.

A thesis in ICE

To demonstrate the HTML output from a thesis in ICE, a screen shot (Figure 4) is included. The screenshot shows the HTML version of the author's honours thesis (Sefton 1990). At the left is a table of contents for the work as a whole, while the main part of the screen is

occupied with a table of contents for the first chapter. Each part of the thesis is available in PDF format, as is a complete book containing all the chapters. The current demonstration does not have tables of figures, tables or examples but these are built in features of word processors the ICE system documentation is being extended to cover these at time of writing. Additional work involves programming the ICE application to be aware of existing word processing features present in both Word and Writer.

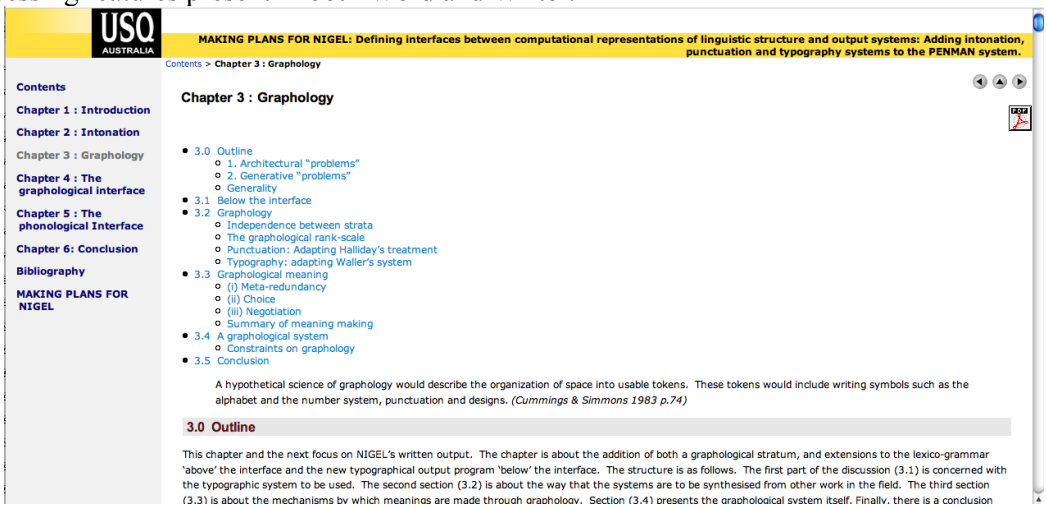

Figure 4: A screenshot showing the HTML view of a thesis in ICE.

The following screenshot, figure 5 is of the ICE style menu that is used to apply styles to a document. This shows the blockquote style `bq1` being applied:

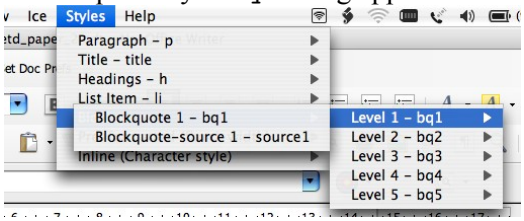

Figure 5: The ICE styles menu in action, applying the blockquote style

Ice provides a 'file manager' view of the documents that make up the thesis. In figure 6, an author is adding a new file to the thesis: in this case an example document for this paper:

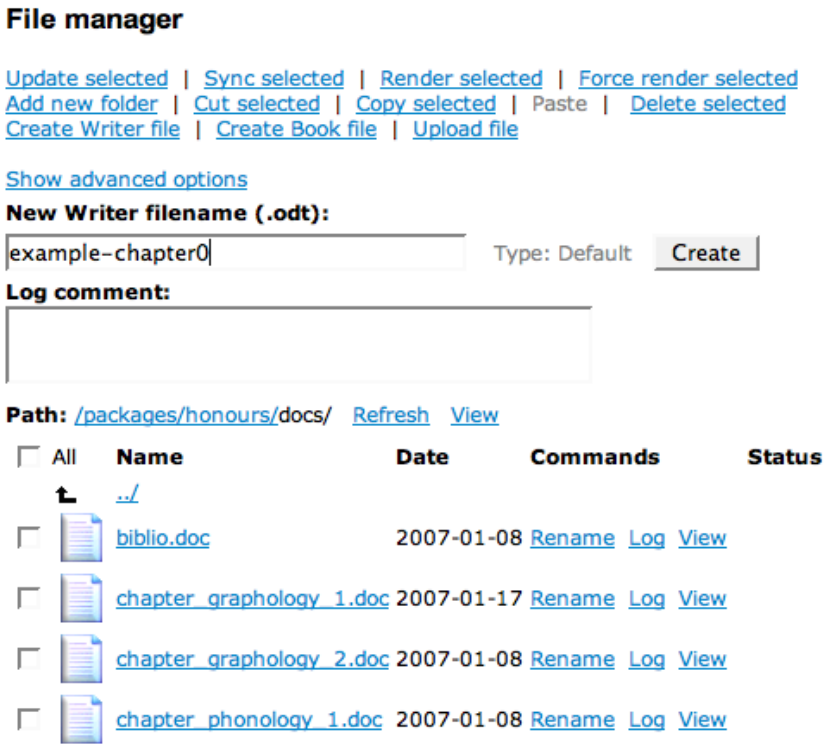

Figure 6: The ICE file manager, adding a new document.

Once the document has been added, the ICE system shows its status in the Subversion repository which is that it has been Added, but not committed to the repository as seen in Figure 7:

Figure 7: Detail from the ICE file manager showing a file that has been 'Added' to the local working copy but not yet committed to the repository

ETD lifecycle using ICE

In the context of ETDs ICE-RS is looking at the entire lifecycle of a thesis from conception to archiving in a repository.

Version-controlled workspace

When a new candidate enrolls, or a new user is introduced to the ICE-RS project they get an ICE account, with a workspace with a pre-populated folder structure for their thesis and associated materials. The ICE application can, at the click of a web-page button, synchronize

all of the user's work to a version-controlled central repository. In time, this will allow researchers to study the way theses are created in a similar way to the way code may be explored when programming students use the Subversion system (Glassy 2006).

In addition to the ETD, the workspace could be used for the research materials and papers written in the course of research, small data sets and references; thus the tools used for an ETD are identical to those used in other pursuits.

The Subversion-based workspace has been proven to work well in team environments. For example the RUBRIC project website is produced using ICE, with several authors collaborating on content which is written using OpenOffice.org Writer (University of Southern Queensland 2007b). A forthcoming major report from the project is also being produced using the ICE software, allowing multiple authors to work on the same content with no chance of losing changes. The report will take the form of both a book, and a collection of electronic resources, similar to a multimedia ETD in scope.

Because it has worked in a team environment we are confident that ICE will work in an environment where there is only one author, with one or more other supervisors and reviewers.

A number of pilot users have been identified at the University of Southern Queensland, and workspace have been created for them. Currently five existing theses are being converted into the ICE format to demonstrate that it will enable authors to create HTML and PDF versions of their work, but also provided at least as effective an editing environment as the *ad hoc* approach currently prevalent at the university. One candidate has begun writing a PhD using ICE and more are being recruited.

One of the known limitations with the current architecture is that not all students have access to a dedicated machine or machines on which to work, so a server-based version of ICE is under development.

Advantages of editing using a word processor

It was a key goal of the ICE project that editing be done in a word processor, and that users would be able to take advantage of the WYSIWYG approach to editing, while still having access to an automated toolchain to generate high quality XHTML.

Word processors contain many useful features for general writing tasks. Some are gimmicks but many are there because they make life much easier. A brief description of some of the more important features follows.

Outlining and rapid document navigation

Both Microsoft Word and OpenOffice.org writer have outline navigation tools that let the author navigate around looking for images, tables, bookmarks and so on. For documents of more than a few pages outliners are an invaluable tool for improving the quality of writing, but some guidance is needed for effective use (Hammel and Kozma 1993) .

Built-in drawing editors

In both Word and Writer, there are a number of tools for drawing and charting. These

tools can be used to create preservation-ready graphics, if some simple guidelines are followed. The main issue is that both packages allow graphics to be drawn on the page itself, over the text a practice that prevents effective export to HTML and even simple re-pagination. The ICE guidelines spell out how authors should proceed (University of Southern Queensland 2007).

As noted in the section on ICE for preservation, nedan use of drawings in MS Word may cause some preservation concerns, but as Office Open XML (ECMA 2006) format becomes more prevalent all drawings created in word processors will become better candidates for long term preservation.

Control over layout via WYSIWYG

Experience by the developers at the University of Southern Queensland showed that automated server-based rendering has two main problems:

a. rendering code is time consuming and expensive to write and maintain, particularly as new requirements come in and,

b. server based rendering causes problems with pagination that can be very time consuming to solve (Sefton 2006a).

ICE therefore allows for WYSIWYG editing using a word processor, but with the use of styles to enable preservation-quality XHTML to be generated automatically.

Discussions are under way with the author of the Digital Scholar's Workbench software to see an automated rendering toolchain based on XHTML rather than PDF as a practical option, this would be a less resource intensive, more consistent approach to formatting an ETD; but major issues remain with rendering images that may have been created using drawing or charting packages embedded in word processor; that the practical solution would probably involve.

Packaging multimedia theses

ICE was originally developed for course materials delivered in print and online and so is able to deal with multimedia such as sound and video as well as presentations such as PowerPoint or Flash. ICE packages courseware using the IMS content packaging specification (IMS 2005), resulting in self-contained web-versions of courses with both HTML and PDF documents and one or more course books that combine the sub-parts. IMS packages can be used with learning management systems, but may also be used as stand-alone web content.

While multimedia features are used widely for courseware at USQ, there are no openly available examples of ICE content. What follows is an illustration of how ICE can be used to include Multimedia content in a document.

Returning to the document that was created earlier, seen in Figure 7, the author can click on the file name to have the document open in OpenOffice.org Writer. The author can link to a sound file they wish to include using standard word processing features. Figure 8 shows this in Writer:

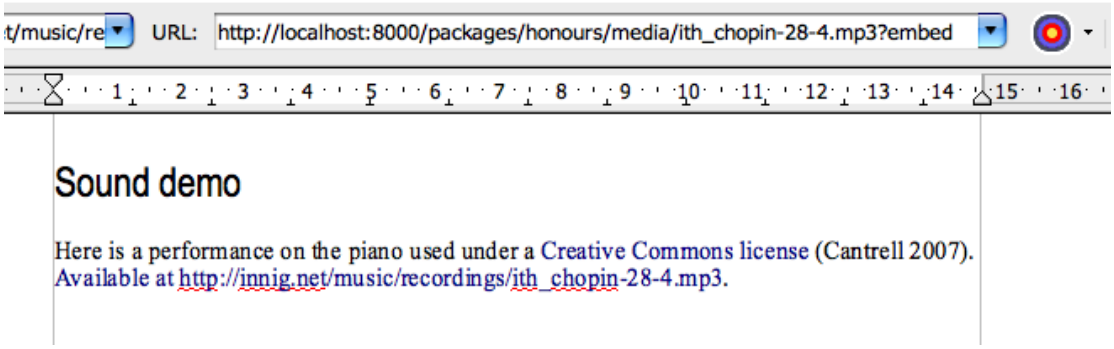

Figure 8: Screenshot of the Writer application with a link to a local copy of a sound file. The text of the link offers a human readable URL for fetching the same resource over the web.

There is a workflow issue here that is of great importance. In the finished version of a thesis there needs to be a readable reference to the online version of the multimedia. In fact, the location of multimedia files may change over the lifecycle of the thesis. In the writing stages they will reside in the ICE repository. When the thesis is sent to examiners the files may be provided on a disc, which is trivial to do using ICE, or made available on a web site.

In this case the link which is visible in the URL window is pointing at a local copy of the music file, while the text shows a link that will be appropriate for readers of the print version to use to access the music.

The HTML version of the example document, which can be distributed via the web or on disc looks as shown in Figure 9. Note that a sound player is embedded in the browser and in this case there is no need for a visible URL as there is in the print version:

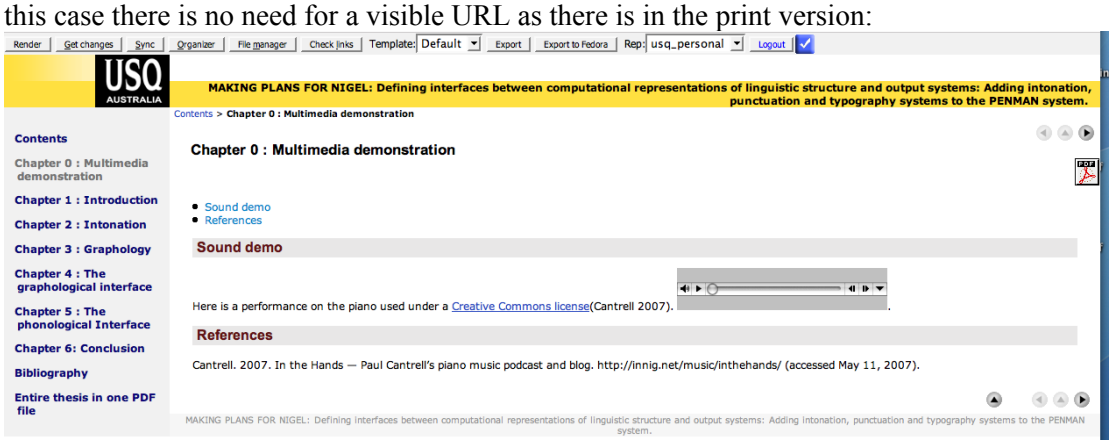

Figure 9: A screenshot of the web version of the example chapter showing an inline sound player.

But how can we link to multimedia without knowing the final URL for the thesis when it is added to a repository? The answer lies with using handles as persistent identifiers.

Persistent identifiers: handles

In order to be able to deal with the linking issue identified in the section on multimedia, above, some kind of persistent identifier and associated resolut ion technology will be needed. The chosen technology for ICE is the *Handle system* (Prasad and Guha 2005, 7).

Work on is planned to add handle technology to ICE. What this means is that each file in an ICE repository will be assigned a handle automaticall y by the software.

1. Initially the handle will resolve to the server-side ICE repository, which because it is in the Subversion system is web-addressable, although usually authentication will be required.

   The author need not worry about handles at all: they can use links in the usual way to manage their content and the system will manage the creation and management of handles when content is exported from the system.

   An internal link would look like an ICE URL:

   ```
   http://localhost:8000/some-path

   When exported it would use a handle resolver:
   http://myresolver.edu.au/hdl/4435387897435
   ```

   Which would resolve to the ICE repository:

   ```
   http://myrepository.myuni.edu.au/some-path
   ```

2. When the ETD goes from authoring to the examination phase the software will export it from the ICE site to a web site, media such as a CD-ROM or access controlled repository: the handles will then resolve to any online version.
3. Upon deposit into a repository, ingest software will update the handle records for every part of the ETD including all the documents, embedded images. multimedia and so on, so that any exiting copies of the work, including both electronic and paper versions will still contain working links.

Bibliography and reference management

Until the recent advent of Microsoft Word 2007 there has not been bibliographic support in Word itself. Word 2007 does have some support, however it is expected that at least at USQ it will be several years before the package is widely enough distributed for it to be a viable option for collaborative work.

OpenOffice.org Writer does have bibliographic support but it is very limited and will not interoperate with Word.

The ICE project has selected Zotero (Zotero - Quick Start Guide) as a bibliographic management tool, because like ICE it is open source and works cross platform, running on Linux Mac OS X and Windows. Zotero is a research tool that works from within the Firefox web browser and integrates a bibliographic tool with a way to manage web page snapshots and PDF files.

While it is usable, and promises to provide a collaborative bibliographic tool worthy of the ICE system, two main issues remain with the Zotero plugin as of May 2007:

- Zotero does not use unique, persistent identifies for resources, so it is impossible for more than one user to collaborate on a document or to reliably share references.
- While Zotero has a general purpose citation specification system only a small number of these are available in the word processor plugin.

The ICE-RS project has contributed to Zotero and will continue to work on and sponsor Zotero development.

Collaboration

A key goal of the ICE project has been to produce collaborative editing software. At the time of writing the basic infrastructure is complete, with the Subversion back-end taking care of distributed editing and version control. Teams of authors and reviewers can exchange documents and work on large projects together.

Current collaboration services

Currently collaboration is tied to word processing documents. Users can take advantage of change-tracking and commenting features that work across both Word and Writer. The subversion version control aids in moving documents between collaborators.

The ICE platform will be used as the basis for a wide range of collaboration services, to be completed in 2007 under the ICE-RS project. Further developments in collaboration services are discussed below.

In the short term, an annotation / commenting system is under development. This will allow reviewers, such as thesis supervisors to add annotations to author's work without touching the original document, and will eventually allow workflows such as voting on a document or rating its content in various dimensions.

Repository systems

One of the key aims of the ICE-RS project is to provide integration between the ICE content management system, which provides a repository for work in progress and the ultimate destination of an ETD in an institutional repository. The ICE-RS team will work with collaborators from the APSR (APSR Project 2007) and ARROW (ARROW Project 2007; Harboe-Ree, Treloar, and Sabto 2003) projects in Australia to show how content can be ingested into Fedora and Dspace.

Already achieved is a demonstration of ICE content, shown in Figure 10, including theses automatically ingested into Fedora. The following is a screenshot of the first, simple interface.

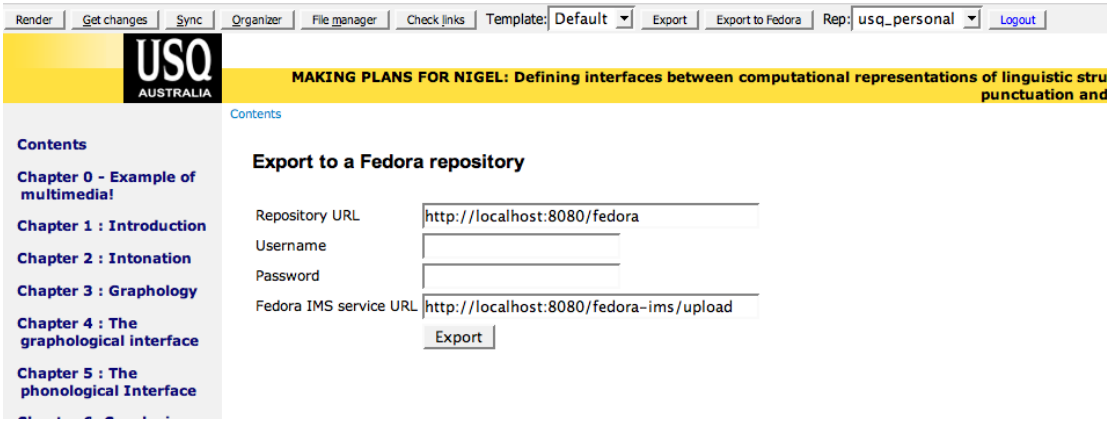

Figure 10 A screenshot of a rudimentary interface for ingesting ICE generated theses into a Fedora repository.

Once the content has been ingested, there is very little to see: the thesis appears to a reader as though it is a small self-contained web site, although all the content is being served by the Fedora repository.

As described in the next section, this rudimentary interface will disappear as the process of publishing an ETD to a repository becomes more integrated into the workflow services provided by ICE.

Planned collaboration services: Workflow 2.0

This section is adapted from a blog posting by the author, which developed ideas about the role of the ICE system in research workflows (Sefton 2006).

The idea of Web 2.0 is one of a collaborative web (Wikipedia 2007) – highlighting has been added to show the relevant concepts for ETDs.

> In the opening talk of the first Web 2.0 conference, Tim O'Reilly and John Battelle summarized key principles they believed characterized Web 2.0 applications:
>
> - the Web as a platform
> - **data as the driving force**
> - **network effects** created by an architecture of participation
> - innovation in assembly of systems and sites composed by **pulling together features** from distributed, independent developers (a kind of "open source" development)
> - lightweight business models enabled by **content and service syndication**
> - the end of the software adoption cycle ("the perpetual beta")

- software above the level of a single device, leveraging the power of The Long Tail.

With ICE there is no intention of adding a workflow model as reported for the SCOPE system (Müller and Klatt 2005), rather ICE is engineered to be a good web 2.0 citizen, allowing workflow to emerge from the interaction of multiple systems: this means that ICE would work well with a workflow system that could route workflows.

Emergent workflow

Currently it is *de-rigeur* for institutional repositories (IRs) to have a web-ingest mechanism. That is, there is a form you can fill out on the web to upload a document.

But in a Workflow 2.0 system the IR might not need to have a web ingest. Instead there could be an **ingest rule** that would fire when a certain context was detected, such as a document attaining a certain state of development, which would be reflected in it's metadata. This is realization of the notion of '**data as the driving force**' quoted above. In this case metadata is doing the driving.

If for example, if there were a system managing the process of sending theses to examiners and collating the results then that system could use an ATOM feed or similar to send updates to related systems. The IR would subscribe to the feed and ingest the new thesis content automatically. That represents **content and service syndication**.

Presented here is a future scenario for this workflow:

1. A new candidate enrolls. The ICE system picks up that fact from talking to the student administration system and automatically creates a workspace for them in ICE, with a directory containing a blank thesis, and documents for the other stages, such as a proposal.
2. The candidate receives training in the use of ICE for general purpose writing, and begins to gain experience writing papers.
3. The candidate decides to write a paper for a conference. Let' s say it's the ETD conference (Uppsala Universitet 2007).
4. She goes to the drafts folder in ICE and adds a new document, choosing 'conference paper' as a template – if a specific ETD paper template is not yet available.
5. ICE brings up a form with a few key details,
    c. name of the conference,
    d. date the full paper is due,
    e. dates the conference will be held,
    f. working title for the paper, and
    g. what referencing style does the conference require?

    (If ICE does not already know some of the above entered by another user)
6. ICE creates a document and the candidate starts working on an abstract.
7. A month before the deadline, her calendar reminds her that the paper is due soon.

When the draft was added ICE added the dates to its iCal calendar, including reminders at strategic points. The candidate's calendar application is subscribed to ICE, so the reminders will pop up at the right time. This illustrates the concept of workflow 2.0; there is no overarching 'flowchart' in ICE that's mapping out steps, rather an arms-length integration with my other tools: **pulling together features.**

8. The candidate finishes the paper and uses ICE to format it to AusWeb's style guide.

   This process could take a number of forms: if there is a pre-built ICE stylesheet for the conference then the formating can be done via the ICE system. If not someone has to:

   Adapt the look of the ICE document by changing its presentation to match that required, without changing the style names used. This is approach is adequate for ETD 2007, as discussed by the author with the organizers. For a colophon on this paper, see the author's website (Sefton 2007).

9. The candidate submits the formatted paper by email.
10. The paper is accepted. The candidate notes this by adding a check box in ICE using the forthcoming flexible annotation system.
11. The USQ e-Prints site notices that the paper has been accepted, via an ATOM feed that watches ICE, and grabs a copy for the repository, where is sits in a buffer awaiting review.
12. A librarian **grooms the metadata** attached to the document and when it meets the requirements of the repository, uses the IR's internal workflow to ingest the document. Optionally, the librarian could do this in ICE some time during the authoring process, negating the need for an approval process at the end.
13. ICE reminds the calendar application that the candidate has three weeks to sort out her travel.

To make this happen the following will be added to the ICE application by the ICE-RS project:

1. Support for document metadata in many formats, including bibliographic and time/date metadata.
2. The ability to publish ATOM feeds and calendars based on queries against that metadata.
3. Indexing of text and metadata to support this functional ity.

Planned services: dashboard reporting

One of the key aims for the ICE system in general is to work well with other systems, as outlined in Planned collaboration services: Workflow 2.0, above. One class of system is a 'dashboard' that would summarize the state of a document or collection.

The ICE-RS proposal promised:

> Each researcher will have access to their own documents.,The centralized reporting system, dashboard, will be able to look into the document store and report on such things as:

- The number and type of documents under development.
- Target completion dates.
- Targeted publishers, journals and conferences.
- Research by subject.
- Completeness of research theses via metrics such as word-count and via ratings applied by supervisors.

These provide management indicators for IT support staff seeking to target their training resources, marketing departments looking for upcoming research that can be tied to a campaign, courseware authors seeking pre-release research, public relations seeking interesting opportunities, and the research office to aid in discovering nascent research, particularly where intellectual property might be at stake.

(Sefton 2006)

The kind of dashboard that may of use to a candidate and her supervisor might look something like the mockup shown here in Figure 11. The dashboard is showing a mixture of automatically generated data, citations and word count, and author-generated content in the form of 'todo' items; placeholders for where there is work to be done.

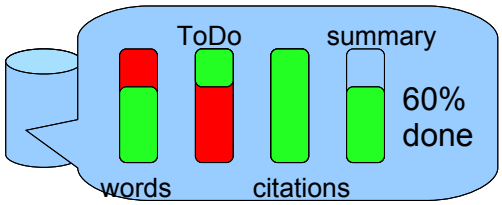

Figure 11: A mockup of a dashboard view of a thesis

As with the enhancements mention in Planned collaboration services: Workflow 2.0, above the main enabler for this functionality will be planned indexing service that will make it easy to perform tasks such as citation counting or listing 'todo' items.

Conclusion

This paper has outlined a project which is taking an established free software application, The Integrated Content Environment – ICE and using and adapting it to scholarly purposes.

The most important contribution of the ICE system as a content management application is that it allows multi-output publishing with underlying XML in the hands of typical word processing users. It will allow publication in both HTML and PDF, with the ability to include multimedia in online versions while still being able to produce print versions of content.

This has the potential to provide much richer institutional and ETD repositories, serving more than PDF files. There is also demonstrable potential for packaging multimedia, and a solution to managing the lifecycle of that multimedi a via handles as persistent identifiers.

Bibliography

Adobe. 2007. PDF reference. http://www.adobe.com/devnet/pdf/pdf_reference.html (accessed May 13, 2007).

APSR Project. 2007. Australian Partnership for Sustainable Repositories. http://www.apsr.edu.au/ (accessed May 10, 2007).

ARROW Project. 2007. Australian Research Repositories Online to the World. http://www.arrow.edu.au/ (accessed May 10, 2007).

Barnes, I. 2006. Integrating the Repository with Academic Workflow. *OpenReposiotries. Sydney, APSR http://www. apsr. edu. au/Open_Repositories_2006/ian_barnes. pdf*.

Barnes, Ian. 2007. The Digital Scholar's Workbench. http://elpub.scix.net/cgi-bin/works/Show?_id=159_elpub2007 (accessed May 14, 2007).

Collins-Sussmann, B., B. W. Fitzpatrick, and C. M. Pilato. 2004. *Version Control with Subversion*. O'Reilly.

Dobratz, S. 2005. Thinking the long term: the XML-based publishing Workflow for handling electronic theses and dissertations at Humboldt-University Berlin. *ETD2005: evolution through discovery 8th International Symposium on Electronic Theses & Dissertations*.

ECMA. 2006. *Standard ECMA-376: Office Open XML File Formats*. Ecma International . http://www.ecma-international.org/publications/standards/Ecma-376.htm (accessed May 9, 2007).

Glassy, Louis. 2006. *Using version control to observe student software development processes* . Consortium for Computing Sciences in Colleges.

Hammel, M. L., and R. B. Kozma. 1993. Using an outliner with a word processor. *Computers in Human Behavior* 9, no. 1: 65-81.

Harboe-Ree, C., A. Treloar, and M. Sabto. 2003. ARROW: Australian Research Repositories Online to the World. http://eprint.monash.edu.au/archive/00000046/ .

IMS. 2005. IMS Content Packaging Overview Version 1.2 Public Draft. http://www.imsglobal.org/content/packaging/cpv1p2pd/imscp_oviewv1p2pd.html .

Jones, Brian. 2005. Word XSLT: Data Only Transform. *Brian Jones: Open XML Formats*. http://blogs.msdn.com/brian_jones/archive/2005/07/08/436973.aspx#452483 (accessed May 9, 2007).

Müller, E., U. Klosa, S. Andersson, and P. Hansson. 2003. The DiVA Project-Development of an Electronic Publishing System. *DĐLib Magazine* 9, no. 11.

Müller, E., U. Klosa, P. Hansson, S. Andersson, and E. Siira. 2003. Using XML for Long-term Preservation. http://edoc.hu-berlin. de/etd2003/hansson-peter/PDF/index. pdf.

Müller, U., and M. Klatt. 2005. SCOPE - An XML Based Publishing Platform. *ETD*. http://adt.caul.edu.au/etd2005/papers/041Mul ler.pdf .

OASIS. 2005. OpenDocument v1.0 specification. http://www.oasis-open.org/committees/download.php/12572/ OpenDocument-v1.0-os.pdf .

Park, E. G., Q. Zou, and D. McKnight. 2007. Electronic thesis initiative: pilot project of McGill University, Montreal The Authors. *Program: electronic library and information systems* 41, no. 1: 81-91.

Pemberton, S. 2000. XHTML 1.0: The Extensible HyperText Markup Language. *World Wide Web Consortium Recommendation xhtml1, January*.

Prasad, A. R. D., and N. Guha. 2005. Persistent Identifiers for Digital Resources.

Sefton, P. 1990. Making plans for Nigel (or defining interfaces between computational representations of linguistic structure and output systems: Adding intonation, punctuation and typography systems to the PENMAN system). BA Honours Thesis. *BA Honours Thesis, Department of Linguistics, University of Sydney* .

———. 2006a. The integrated content environment. http://eprints.usq.edu.au/archive/00000697/ 01/Sefton_ICE-ausweb06-paper-revi sed-3.pdf.

———. 2006b. The Integrated Content Environment for Research and Scholarship. http://ice.usq.edu.au/introduction/ice_rs. htm (accessed April 30, 2007).

———. 2006c. Workflow 2.0. http://ptsefton.com/blog/2006/10/24/ workflow_2_0 (accessed May 10, 2007).

———. 2007. Submitting a paper to a conference using ICE. *PT's outing*. http://ptsefton.com/blog/2007/05/15/ etd_paper (accessed May 15, 2007).

Sørgaard, P., and T. I. Sandahl. Problems with Styles in Word Processing: A Weak Foundation for Electronic Publishing with SGML. *Proceedings of the 30th HICSS*.

Sperberg-McQueen, C M, and L Burnard. 2002. *The XML Version of the TEI Guidelines* . The TEI Consortium. http://www.tei-c.org/P4X/index.html (accessed May 13, 2007).

University of Southern Queensland. 2007a. ICE User guide. http://ice.usq.edu.au/instructions/user_gui de.htm (accessed May 9, 2007).

———. 2007b. RUBRIC. http://rubric.edu.au/ (accessed May 9, 2007).

Uppsala Universitet. 2007. ETD 2007: Added values to e-theses. http://epc.ub.uu.se/etd2007/ (accessed May 13, 2007).

Walsh, N., and L. Muellner. 1999. *DocBookThe Definitive Guide*. O'Reilly.

Wikipedia. 2007. Web 2.0. http://en.wikipedia.org/wiki/Web_2.0#Introduction (accessed May 10, 2007).

Zotero - Quick Start Guide. http://www.zotero.org/documentation/quick_start_guide (accessed October 5, 2006).