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 made 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 sustainable:

> 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 flexible 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 on-line 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.

What is ICE?

- **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 off-line and synchronize with the server copy when on-line.

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 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). In summary, ICE is more automated than most XML based publishing 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. (It sound like you are saying that XML based publishing system rely on word processor input more so that ICE!)

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