Potential projects: #1 A general purpose document annotation system

In December I was in Cambridge visiting Peter Murray Rust's group. I have been working with Jim Downing, Nick Day and Joe Townsend on the TheOREM-ICE project.

Thanks to the Cambridge team for hosting me, the several visits to the pub and to Peter for inviting me to dinner at his college. Luckily I don't think there are any photos of me wearing the tie.

While I was there we also looked at the latest JISC grant funding call, with visitors Ben O'Steen and David Flanders. There was some *very* interesting discussion but I'd better not say any more than that at this stage. Last time I looked, the University of Southern Queensland was not in the UK so we're not eligible for JISC funds, but we can work with UK institutions as we are with the ICE work on the TheOREM project.

What I thought I'd do is put up a couple of posts about small development projects where I think our group here at the Australian Digital Futures Institute could make an efficient contribution, and see if these ideas align with work that groups in the UK might be doing or contemplating.

The first such post is on a general purpose document annotation system, building on the work we have done in the ICE content management system and our repository work. This project is to build collaboration service initially for document annotation and commenting, but with scope for several enhancements which are discussed below. I've talked this one over with Jim Downing, and it has a lot in common with one of his pet projects.

Why?

One motivation for this proposal is that we have an annotation system in the Integrated Content Environment (ICE), which we have been using for document-centered collaboration throughout this year and it has been a very valuable service. The system has been used to seek comment from internal and external parties on documents ranging from blog posts to grant applications to journal and conference articles. The annotation system is a pure-web system where reviewers can quickly and easily contribute comments on a document and carry on threaded conversations inline.

While the system is useful, it has some issues – the main ones being that to invite comment we currently have to add users to our ICE website and the system is also tied to ICE. ICE is a great way to make good quality HTML but there is no particular requirement for this kind of service that the HTML be good quality.

It is very common to want to share a draft compound document with reviewers and contributors. Often this ends up happening by emailing around a word processing document, with contributions via the track changes feature. While track changes works in small groups with homogeneous software and serial workflows it can be unmanageable when multiple people make changes in parallel.

Wikis are one way to collaborate on documents. We use wikis in our software development projects for multi-authored documents, particularly for things that change rapidly.

But for some documents such as grant applications and papers and theses, it makes sense to write the document in a word processor, which makes editing much easier than typing into a web page, with features such as reference management (eg via Zotero) outlining, drag and drop table editing, embedded data visualizations and drawing tools.

Google Docs is one option for collaboration. It has excellent support for simultaneous editing by multiple parties, but it is a proprietary system which does not support many of the features outlined in this proposal, such as OpenId support, stand-off annotations. While an RSS feed is available for comments this only applies to publicly available documents.

There are other free-yet-proprietary systems such as Adobe Buzzword which suffer from similar limitations.

The main issue with proprietary hosted systems is that they are subject to change by the owners, potentially without warning, more so than offline proprietary systems. This proposal is to develop open source software which can be deployed when and as needed and which is not subject to the whims of a software company.

Another option would be to use a blog for seeking comment, but then the comments usually go to the bottom of the page, not inline.

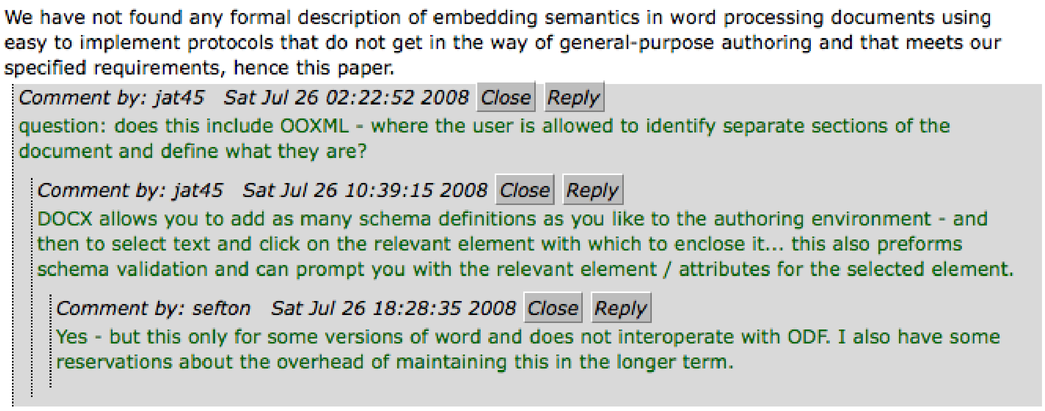
Some simple use-cases for this proposed service:

- Circulate a draft paper to a group for review with minimal configuration: simply upload the document to the server then mail the URL.
- Request confirmation from a group of authors that a paper is ready to submit, via a simple form.
- Subject a document to formal review where the reviewers are unknown to each other, with the ability to both add paragraph level annotation to make specific comments or point out typos and to fill out a form evaluating the document. This could be used for thesis examination for example.

What?

This service will allow ad hoc document sharing and commenting. It will accept any web-renderable compound document via a variety of services and expose it to for paragraph level commenting, using the annotation system which has already been developed for ICE. The system will build on prior work on schemas and infrastructure for representing stand-off annotations, notably work done by Jane Hunter's group at the University of Queensland.

A user will be able to input word processing documents, spreadsheets, presentations etc, have them automatically converted to web pages and allow others to add threaded comments to them at a paragraph level. Below is a real example of this in action in a paper:



We have not found any formal description of embedding semantics in word processing documents using easy to implement protocols that do not get in the way of general-purpose authoring and that meets our specified requirements, hence this paper.

Comment by: jat45 Sat Jul 26 02:22:52 2008 Close Reply

question: does this include OOXML - where the user is allowed to identify separate sections of the document and define what they are?

Comment by: jat45 Sat Jul 26 10:39:15 2008 Close Reply

DOCX allows you to add as many schema definitions as you like to the authoring environment - and then to select text and click on the relevant element with which to enclose it... this also preforms schema validation and can prompt you with the relevant element / attributes for the selected element.

Comment by: sefton Sat Jul 26 18:28:35 2008 Close Reply

Yes - but this only for some versions of word and does not interoperate with ODF. I also have some reservations about the overhead of maintaining this in the longer term.

Document input

Document input will be via a variety of means:

- Simple file upload – if the document uploaded is an ICE document then it will be rendered into HTML and PDF, if not the service will make a 'best effort' attempt to render to HTML using something like the save as HTML in Word or potentially via a service like Google Docs.

- Via a URL, for web content; the system will be able to import a web page and its associated images.

- AtomPub, via any AtomPub client including the ICE template and the ICE template along with Microsoft Word 2007 and others.

- SWORD – a smart SWORD service will accept single documents, zipped web content such as IMS packages or ORE.

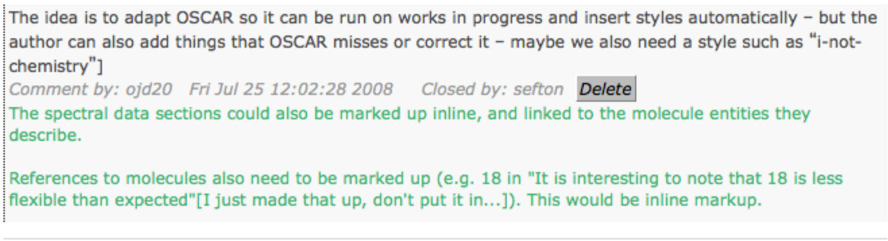
- Atom feed – documents can be auto ingested from any service with an Atom feed.

- Email – a user will be allocated an email address to which they can send documents. These would go into an inbox, and the service will mail back a web link to the admin screen for document sharing.

Commenting features

The key feature of this service will be inline paragraph level annotations as used in ICE.

Comments are tied to a particular version of a paragraph. If the paragraph changes the annotations are kept, along with a snapshot of the paragraph and displayed at the end of the document:



The idea is to adapt OSCAR so it can be run on works in progress and insert styles automatically – but the author can also add things that OSCAR misses or correct it – maybe we also need a style such as “i-not-chemistry”]

Comment by: ojd20   Fri Jul 25 12:02:28 2008   Closed by: sefton

The spectral data sections could also be marked up inline, and linked to the molecule entities they describe.

References to molecules also need to be marked up (e.g. 18 in "It is interesting to note that 18 is less flexible than expected"[I just made that up, don't put it in...]). This would be inline markup.

Annotations will be stored in a 'stand-off' fashion, that is they are not inserted into the document but stored alongside it.

In addition to the inline annotations the system will allow for the user to upload a form to be displayed alongside the document. This could be as simple as two options like Accept/Reject or could encompass a detailed assessment rubric. The form data will be serialized into JSON and stored as an annotation.

Authentication and authorization

Authentication for this system will be via OpenId.

Access to documents will be via obscure URIs with an option to make documents discoverable via browse search or not. Annotation will be subject to the following access controls:

1. **Open commenting** where the document is available to all and anyone can comment, logging in with an OpenId or,

optionally picking a nickname without logging in. (TODO: Is there an OpenId service we could use which can make it painless for people to mint a new ID for casual commenting?)

2. **Whitelist access** to certain openids.

Comments need to be treated in different ways for different use-cases:

- Annotations visible to all reviewers.
- Annotations visible only to the reviewer who made them, and to the document owner.

Integration points

This service is not intended for long term document hosting, but for short-term document review (although it might make the basis for a good blogging system or an add-on to a repository in future). Annotations and form data collected about an item will be made available to other systems via authenticated ATOM feeds and ORE resource maps.

The idea is that another service, such as a graduate studies department thesis management system, which we are exploring in the TheOREM-ICE project could use this annotation service to collect reports from thesis examiners (who would log in with OpenIds and not be able to see each other's annotations and/or form data).

Technical implementation

Implementation for this project would draw on open source software – most of what is required is a simple integration job. The following are promising starting points, but other libraries and code might be used at implementation time:

Function	**OSS**
In-page annotation	JQuery implementation from ICE with some refinements to generate paragraph Ids in-page (currently documents are pre-processed in ICE)
Container application	Apache Tomcat + Velocity templates
Document indexing / access control	The Fascinator (built on Tomcat)
Annotation storage	Fedora (allowing The Fascinator to index annotations and serve them appropriately)
Forms serialization	Forms serizlier / deserializer from the Java port of VALET that we're working on at USQ.
Form authoring	Unknown.

ORE / SWORD	TheOREM-ICE components
Document conversion	ICE conversion services (with additional code to convert presentations and do best-effort conversion on non-ICE documents, potentially using Lemon8-XML)
Annotation metadata schema	Use work from Jane Hunter's group at UQ and any code that has been released by that group for assembling metadata records

Future work

This system is designed to be a piece of research and learning infrastructure which can interoperate with other services.

Future directions might include:

- Adding a plugin architecture so other kinds of content than document may be annotated (eg crystallography, tabular data, chemical reaction chains).
- Reworking the service as a plugin for other systems such as WordPress, Drupal, Sakai, or Plone.
- Extending the annotation system to in-paragraph changes and subsequently to a full online word processor (that is building the open source equivalent of Google Docs).
- Tighter integration with other systems so that the annotation server appears seamlessly inline eg in an assignment marking system or in a repository. For example, the system could be embedded into the Moodle learning system, with annotations being automatically fed-back into the Moodle discussion system.
- Extending the system into a blogging platform with the emphasis on integration into academic workflows.