Towards Beyond The PDF - a summary of work we've been doing

There's a workshop happening in the USA in February 2011 called Beyond the PDF. This is really exciting, as we have been working for years here at USQ, and with collaborators such as Peter Murray-Rust and Jim Downing from Cambridge on just that: getting beyond the PDF in academia. I'm pleased to see this movement starting to get some traction.

In this post I want to both catalogue some of the work we've done that's relevant to this workshop if only to remind myself, but also to look at some of the issues that will be relevant to the workshop aims.

> **Immediate Goal:** The goal of this workshop is not to produce a white paper! Rather it is to identify a set of requirements, and a group of willing participants to develop open source code to accelerate knowledge sharing. Our starting point, and the only prerequisite to participating, is the belief that we need to move *Beyond the PDF*.
> Specifically, we think that better integration between the research paper and research data is imperative - see our papers for more details on this thinking, and please add your own so we know your thoughts!

The workshop is not all about technology it's about ecosystems and workflows, and changing practice. So while I will go through some of the technologies we've been working on I'll start with some of my ideas about the bigger picture. I will follow up this post with some demonstrations targeted at the *Beyond the PDF* sample files so if you're short of time, you might like to skip all this and wait for the show and tell.

Issues

Some of the big issues as I see them are:

1. **We have to bring tools to the researcher's desktop** that help them to manage their data and make the links between publications and the web of data. There are a few different, overlapping 'camps' there are LaTeX users, there are legions of Word (and OpenOffice.org) users, there may be some disciplines where researchers are writing XML. That's the authoring tools. Then there are the workflow tools, where we have SharePoint, Sakai, Drupal, et al and plain-old file-shares, revision control system and all too often the dreaded email and there's the scientific infrastructure that generates a lot of the data, microscopes and cameras and specialised instruments of every stripe.
2. **Publishers.** We can't ignore them. We either have to work with them or route around them, creating new publishing channel. I explored this in a paper in Serials Review. One of the big issues is that if people are focussed on writing for publication, and the publisher takes their manuscript and then does stuff to it and hoards the result, then that breaks the end-to-end semantically-rich publishing workflow we should be looking at.

> To illustrate the potential divide between the author's version and the publisher's, consider that Elsevier, the publisher of this journal, recently ran a competition, Article 2.0[9] to show the future of a scientific article. The competition winner shows that a journal article may be the Web locus for discussion, annotation and semantic relationships, but this competition was built on XML source documents which are created and held by the publisher, so there is no way that a typical institutional repository could easily provide the same services. This is a case where the publisher is shaping scholarly communications, or at least exploring how to do so, but a lack of tools means that repositories are unlikely to be able to do likewise. This creates a distinct divide between the publisher's more richly marked-up version and the version held by the author in word processing format or the typesetting system LaTeX,[10] neither of which allow high quality HTML unless the

author has used a particular set of templates and/or macros and has access to specific conversion software. So there is no way for most author manuscripts, which are commonly deposited in institutional repositories, to be turned into usable Web content, let alone with links to data and semantic-Web content. The best most authors could hope for with their version would be to convert it to PDF and deposit in a repository, while the publisher can do much more with the article.

3. **Theses.** As Peter Murray-Rust has often pointed out, getting the early career researchers is key – and theses are a form of 'publication' where institutions and disciplines control the entire process so we can innovate, and produce 'beyond PDF' ready researcher at the same time. Having a thesis strand in anything that comes out of the workshop is probably a good idea.
4. **Identifying things**. As has already been discussed on the Beyond PDF list, sorting out author names is imperative, to which I would add we should also get really basic stuff like subject codes and terms such as resource types. The most obvious approach these days is to follow Linked Data principles and define URIs for key terms, people, projects etc. We'll be building a mesh of services with local, discipline, national infrastructure all interoperating.

Many of these themes we covered in the paper I presented at Open Repositories 2010, co-authored with Duncan Dickinson.

What lies beyond PDF? I'd say that it's **pretty obviously The Web**, which is why I coined the phrase "Scholarly HTML". Below I'll revisit some of the issues around what this means.

Technologies and ideas

OK, so the goal of Beyond the PDF is to change science, which I'm taking in the broad sense of the word. In Australia we use the term eResearch rather than eScience so as to include the humanities.

What follows is a disorganised catalogue of work I've been involved in that is relevant to getting beyond PDF.

Getting on the web

One of the main things I have been working on for the last fifteen years is helping people get stuff onto the web. Actually on the web as HTML, not as Word documents or PDF or Flash Paper. You'd think that would be a solved problem – but the fact that we have to have a workshop called 'beyond PDF' is testament to the fact this is is well and truly not solved.

The system we developed at USQ to help teachers get their distance-ed materials online and into print, via their word processor is called ICE: The Integrated Content Environment (Sefton 2006). We have explored how ICE can be used for research publication, most importantly for theses (Sefton & Downing 2010; Sefton et al. 2009; Sefton 2007).

I am firmly of the opinion that given the right authoring tools, HTML can be 'the' format for research publications (we might put them in PDF to print them). I went through some of the reasoning behind this in a recent post about whether it is worthwhile to hook up word processing templates to the DocBook XML schema. That article quotes Mark Pilgrim on how he moved his Dive into Python book from DocBook to HTML.

One idea for the workshop: I am pretty sure we could come up with a format based on HTML 5 using Microformats, RDFa etc – this would mean you could drop articles straight into a blog, and Google Scholar could index them directly – and you could still feed them through print publishing processes as XML.

HTML 5 has section elements, and an article element, and support for extending semantics. Take the recipe microformat. Lots of scientific papers are pretty recipe-like. Google says:

> If you have recipe content on your site, you can get started now by marking up your recipes with microdata, RDFa, or the hRecipe microformat. To learn more, read our documentation on how to mark up recipe information or our general help articles on rich snippets for a more complete overview.

(Pay attention to Google. I think we should all remember what happened with OAI-PMH which was 'our' format. Google tried that briefly then went and built Google Scholar to web-crawl HTML, harvesting metadata that's embedded in web pages. I would bet on the same process happening with full-text content – they're not going to be interested in XML, so important semantics will have the surfaced in HTML, and if we're doing that then why bother with the intermediate format?)

We have some ongoing work on how to add semantics to documents using the tools people really use (ie not XML editors). We did some explorations of how to embed metadata using styles and other techniques (Sefton et al. 2009), but I am really excited about new approaches to this problem which would work by extending the model that Microsoft introduced with their Ontology plugin for Microsoft Word (about which there was a fierce debate last year).

I'll post some demos of the kind of thing I'm talking about including what author tools might look like and ideas for how to embed semantics of all kinds in documents.

- **Embedding metadata** in-line in-context. This would mean not only that we know who is an author, but that the instance of their name in the document is labelled with an identifier and a metadata property that indicates the relationship between that person and this document. This will allow for richer semantics than can be captured using the field-value approach taken by the core document metadata in office documents, and to tie metadata semantics to inline context – making it possible for both users and machines to explore semantic webs with document text (or other content) as jumping-off points. So references to contributors and journals would be linked formally and explicitly with useful URIs.

  We did some work on mechanism for this using styles, tables etc but at the moment I think the best approach will be to use links – will explain and demo this further.

- **Embedding references** to ontologies and taxonomies/controlled vocabularies and geographical relationships with explicit relationships

  (Explicit relationships are important – is this document 'about' this kind of lizard or is the reference to a lizard part of statement of what is **not** in scope?)

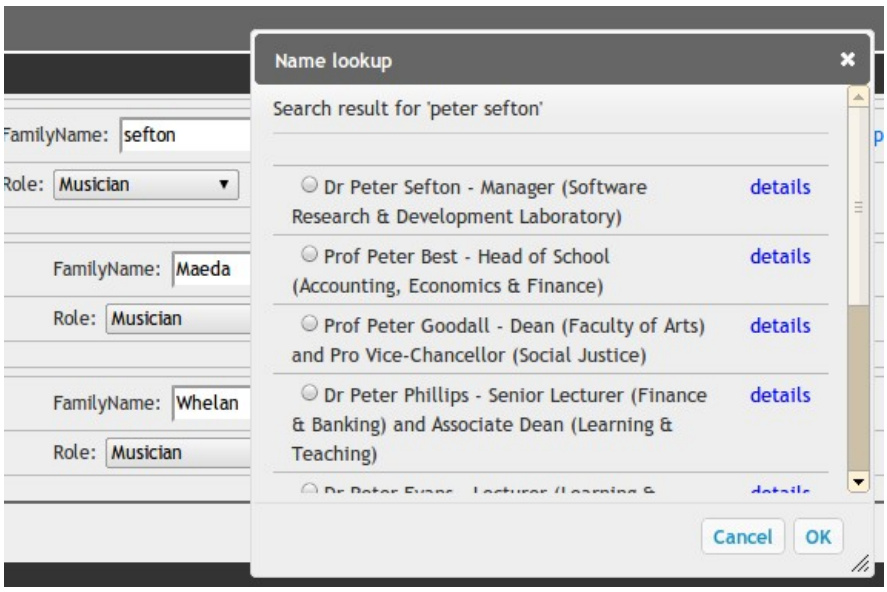
- (Experimentally) **Labelling structural components of documents** to assist in automated conversion to publisher DTDs or directly producing Scholarly HTML. For example a section with a heading “Method” could be labelled with a link to the definition for methodology from the NLM DTD, but the section could also be entitled something like “procedure” or “what we did” but still have the *meaning* of Methodology. This would also allow a section to potentially be associated with more than one document schema. This class of semantics might improve document conversion processes in a similar way to the way heading styles are widely used now, with more semantic rigour and depth.

Improving repository models

A current R&D topic in my team at ADFI at the moment is how to improve on the current IR information model. We're working on a couple of fronts.

1. Exploring, on the ReDBox project which is funded by the Australian National Data Service, how we can describe research data collections, and then post those descriptions to the national aggregation service, via the institutional repository. Part of this work involved establishing an authority control service “The Mint” to handle names as well as subject codes and ontologies, such as geonames. We think you need a source of data that can be made to interact with other tools in a developer-friendly way, We have been working on ways that web developers can look up stuff in The Mint – such as building UIs to help people in resolving the string “John Smith” to a URI for the right Smith, and Toowoomba to the right Toowoomba, not the wrong one.

   Here's a screenshot of the interface in action on the forthcoming USQ Arts repository– this is helping me pick the “Peter Sefton” I mean here.



We need all these bits and pieces of infrastructure to start stitching together the services we'll need to go beyond PDF – pure RDF is not enough this stuff needs to be built into tools.

2. Getting a richer information model to describe the parts of a publication and its related data. The first round of Institutional repositories were not that great at this. You could attach various versions of a paper to a record, but it was not usually clear which was the pre-print, the presentation, the published version, or what might be a data set apart from clues in the file name.

   Duncan Dickinson in my team has worked to expand on outcomes from the Kultur project in the UK, to set up a repository which can use a rich information model to show the relationships between research objects in the arts using the Cataloguing Cultural Objects (CCO) standard. This will help the repository to keep track of relationship subtleties such as the difference between a photograph of a work which is a painting and a photograph which is the work itself, or a photograph which is of three works, etc. The same kind of infrastructure will be needed for the management systems we use for bringing together publications and data – with different information models and vocabularies for different domains.

Packaging

When you look at multiple formats of a document, supporting data, and relations between them packaging becomes important. I'm sure OAI-ORE will be a favourite with the Beyond PDF crowd, but that's a technical framework, not an end-user tool. We've been looking at things that *people* can use to do the packaging, so that machines can handle things like ORE.

We're exploring EPub as a packaging format for exchange and dissemination with easy to use tools.

Another idea we're working on is to make self contained HTML “apps” for scholarly objects – ie package a document, data (or links to it) and visualisations etc into one live thing. Including interaction as well. We have a basic JavaScript toolkit for this called Paquete – the idea is that an document, or aggregation of stuff can be moved around by doing a 'Save as' and an HTML 5 compliant browser will use the manifest to grab all the bits and pieces. The Paquete demo shows how a bunch of document parts can be packaged and served – imagine this model extended to include research data and richer relations between document and data. It is held together with a manifest which could be enriched with more detailed relationship information and transformed to OAI-ORE if necessary. Paquete can allow adding and moving resources as well via drag and drop when embedded in an application.

Annotation

Finally, we're looking at the all-important annotation. We are aware of the work going on in this area, particularly in the Open Annotation Consortium and we plan to interoperate with their protocols, but our approach is a little different than most web-scale annotation systems and proposals.

We see the need to add annotation services to lots of different systems, on lots of different kinds of material, text, image, video, visualisation such as a molecules etc. Our work has been on making an easy to deploy web toolkit that can add such services to existing or new applications. We are focussing on adding annotation to a web service, not building generic clients with the Anotar toolkit.

There are two major design patterns.

1. Build the annotations in to an existing system as an extension. For example the work I did with Ron Ward on adding annotations to WordPress.
2. Allow aggregation across multiple systems. We don't have live exemplars yet but the idea would be to add annotations to multiple systems which would store them back to a central store. Greg Pendlebury has talked about this with VuFind developers – multiple sites, which might not even be running the same software could be set up so collect annotations (comments, tags, taxonomic tags etc) on items to increase the number of participants in the crowd.

This post was written in OpenOffice.org, using templates and tools provided by the Integrated Content Environment project and published to WordPress using The Fascinator.