Hacking towards Scholarly HTML

Following from my attendance at the Beyond the PDF workshop in January, I've been invited to Cambridge by Peter Murray-Rust. That's where I am now. (We have two just under weeks to revolutionise scholarly communications :) Peter has organized a hackfest for the weekend of March 12th and 13th and the theme is "Scholarly HTML" - something I have been writing about, but doing much about for a while now.

What's Scholarly HTML?

I think it's a way of representing 'research objects' along with associated data and metadata, **for the web**, so they can be efficiently created, then reviewed, discussed, machine processed, copied and do on. Scholarly HTML is a way to encapsulate research objects in a portable, preservable and sustainable way using simple technologies so that research can be not just *on* the web but *of* the web.

What I hope to do at the hackfest is map out some guidelines for what Scholarly HTML might look like and then start working out how to author it, and how to present it. And how to do scholarship on top of it. Open review, machine processing, nanopublications – these things all need a platform. Yes you can do these things in PDF, if you write new PDF viewers and so on, but it is so much easier to use the web. That's what the web was originally for, after all. Yes, I will be talking about word processors, and our work on desktop repositories. And yes Martin Fenner will be looking at WordPress and PMR's team are looking at machine generated journals – but I think it's really important to think about what we want in an underlying *format*. Something that can be preserved and exchanged and made to work in different systems.

(I think that sometimes the format itself is hard to see when you are looking at tools like WordPress which are both authoring tool and publishing environment but it is doubly important in that case to make sure that what we're creating is not locked in to a particular platform. How would you archive a document which depends on eight CMS plugins to work? How sustainable is a journal that requires multiple CMS plugins? I'm not saying the folks working on tools now are not thinking about these issues, just emphasising that we will be thinking about hard about separating tools from formats and considering interoperability at the workshop. That includes 'interop with the future' AKA digital preservation.)

In scope for the *format* Scholarly HTML:

- **Articles, theses, reports, lab notebooks, blog posts**, which reference associated stuff such as data and provenance and anything else that makes research reproducible and/or able to be validated. Peter Murray-Rust contributed this:

  > It is critical that the tool [I think he should have said *format* rather than *tool* here] supports easy aggregation and publication of data as part of the article. This includes the following - in text
  >
  > - tables in some accepted format (CSV, *.tex are most common)
  > - bit-map images (micrographs, gels, galaxies...)
  > - vector graphics (SVG, ?WMF, etc.). Plots, apparatus, etc.
  > - maths - equations, variables, code
  > - chemistry - CML and other Open markup languages (PubchemXML?0
  > - maps (GML, coordinates)
  > - supplemental files. All the above and more

- **Structure and semantics for document contents**, like headings, being able to label parts – like "This bit is a *Method"* where *Method* is well defined using a URI and extensible methods for describing relations between and within objects.
- Ways to **represent citations in an unambiguous machine readable way** so that readers can have them presented in ways of their own choosing.
- **Techniques for embedding domain semantics**.

- **Packaging** so that research objects can be moved around, saved, posted, etc. This is going beyond HTML, I know, but it's one of the great unsolved problems of the web. I've talked about this before – and Martin Fenner has picked up the Epub ball and taken it for a good run. More soon on this but just as a teaser, I'm now thinking of proposing that a minimal scholarly HTML package might consist of a zip file with at least one file at the root, index.html, which can link other pages if it wants. This is inspired by Eprints and by the simple standards such as Bagit developed at the California Digital Libraries. This is a minimum – the package could also be an ePub and have a formal ORE manifest etc. Chris Rusbridge mentioned the importance of packaging over on Peter Murray-Rust's blog.

  > … My concern re Scholarly HTML relates to the difficulty in saving an article in HTML and then accessing it later (one of PDF's big advantages is that it does this simply and reliably). Safari does this well with .webarchive but it's non-standard and no other browsers support this. Most use a really clumsy convention with a .html file and an associated directory; not very easy to manage.
  >
  > So my suggestion to help Scholarly HTML would be a plugin (eg for Firefox) or filter that converts Scholarly HTML articles to .epub (eg see mfenner's plugin for epub from WordPress).

Related to the format, but not actually part of Scholarly HTML:

- **Annotations on the above**, where annotation is taken in the broadest possible sense. Scholarly HTML needs to worry about anchors for annotation. Annotation can be used for peer and open review, inline discussion at all stages of a research object life cycle, adding formal semantics and lots of other scholarly processes.
- **Tools, tools tools.** Word, WordPress and anything else that starts with W, maybe even some things that don't. I want to be able to work on papers in OpenOffice and post them to collaboration spaces – including WordPress, so I think a lot of the hackfest should be about interop.
- **Authoring tricks and techniques.** Stuff like KCite where you can add citations in any-old editor using text [[cite source='pubmed']17237047[/cite], and my approach of using links to embed document semantics, like this assertion that I am the author of this post: Peter Sefton.
- **Browser-side plugins to make declarative Scholarly HTML come alive**. There are some problems with browser-side code, particularly when various scripts get in each other's way, but one of the big pluses is that it can work across more than one system really easily, so you don't just get a WordPress plugin, you get something that can be added to any system or turned into an app for offline use.

Important properties of scholarly HTML:

- **It's declarative**. That is if we are linking a document to, say, a map then the document will contain:
  - At least, a link to the map, which must be in a standard format. The link will be labelled in some way to say "I am a link to a map", maybe with further attached semantics.
  - Optionally a static placeholder image for the map so that any old web browser can be used to at least show something.

  Systems for rendering Scholarly HTML will know how to interpret the semantics that there is a map, or a molecule on the end of the link and do something useful. A web page containing Javascript tied directly to Google Maps doesn't qualify – it's not portable. (But as Phillip Lord pointed out in email, if the basic declarative stuff is there then it should be OK to have the script as well.)

  Another example of where a declarative spec is important is citations. Scholarly HTML will have a way to represent citations – there's a really useful discussion going on of the issues around this on the wordpress-for-scientists@googlegroups.com list. Martin Fenner is committed to working on a tool that can find citations in text and format them, this will be a web service. I think we can use the workshop to progress work on microformats for citation and at least consider some of the questions about what are reliable end points for citations and how much potentially redundant bibliographic data to store in the Scholarly HTML.
- It has a **simple structural backbone**, using HTML 5 elements to give the basic hierarchy with optionally much more detail. Using the HTML article element, it should be possible to identify the bounds of a work, and separate it from navigation and branding.

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.