# A very brief history of Scholarly HTML

Martin Fenner, Gobbledygook

March 19, 2011

The history of HTML begins 1989 at CERN, the European Laboratory for Particle Physics in Geneva. **Tim Berners-Lee**, **Robert Cailliau** and colleagues invented HTML (as well as the transport protocol HTTP and the web browser) to facilitate collaboration between CERN physicists.

HTML by nidhug at Flickr.

HTML was originally invented for scholarly communication, but of course by the mid-1990s was also used by everybody else. But when electronic distribution of scholarly journal articles became possible, most publishers switched to PDF instead of HTML as the electronic document format of choice.

In April 2003 Inera, Mulberry Technologies and the NCBI published the NLM Journal Archiving and Interchange DTD Suite (NLM-DTD, read here about the history of this DTD Suite). The NLM-DTD has become the de facto XML standard for scholarly publishing and archiving. Although some tools for authors can write articles in this format (including Microsoft Word with the Article Authoring Add-In), the NLM-DTD has never caught on as an authoring format for scholars and I'm not aware of any publisher accepting manuscripts written in this format.

In April 2009 Learned Publishing published a paper by **David Shotton** titled Semantic publishing: the coming revolution in scientific journal publishing. David listed six rules for semantic publishers:

1. Start simply and improve functionality incrementally.
2. Expect greater things of your authors.
3. Exploit your existing in-house skills fully.
4. Use established standards wherever possible.
5. Publish raw datasets to the Web.
6. Release article metadata, particularly reference lists, in machine-readable form.

Although the paper focusses on scholarly publishers, these rules also very much apply to what we could call Scholarly HTML today.

At about the same time (March 31, 2009) **Peter Sefton** for the first time used

the term Scholarly HTML in a blog post. He thinks that Scholarly HTML should allow the following:

1. Documents should definitely have headings.
2. Protocols for representing things like examples.
3. Metadata, using a linked data approach.
4. Links from terms mentioned in the text to ontologies that describe them.
5. Linkable paragraphs.
6. Dead simple reference management via links to trusted sources.

In January 2011 **Phil Bourne** organized the Beyond the PDF workshop, and I was lucky to attend. We had a number of interesting sessions about how to improve the current scholarly paper published as PDF. I was involved in the working group thinking about better authoring tools, and one of my personal conclusions was that ePub is a very interesting alternative to PDF if we need a packing format for HTML, e.g. for journal submission or archiving.

**Peter Murray-Rust** was able to capture the ideas of the authoring working group with a drawing, and he picked the term Scholarly HTML as the best description of what we want to achieve. He subsequently invited Peter Sefton, Brian McMahon and me (and a number of other people interested in Scholarly HTML) to a workshop that took place last weekend in Cambridge. Some of the thoughts of Peter Murray-Rust and Peter Sefton are summarized here and here. Two outcomes of the workshop are Scholarly HTML Principles and a FAQ.

My approach to Scholarly HTML is through developing WordPress plugins. The future will hopefully bring of a number of Scholarly HTML authoring tools, and WordPress will be just one of them. But Wordpress is a great platform to test ideas and improve them over time, or to quote David Shotton: start simply and improve functionality incrementally. One starting point has been citations in Scholarly HTML and we have already made good progress.

The idea behind Scholarly HTML is not to build alternatives for Microsoft Word or LaTeX for authoring, but instead to build tools that will allow us to do something new and exciting with scholarly content. The trick here is to improve the scholarly document – and the author should see the immediate benefit - without putting too much extra burden on the author. And this might in fact be the big challenge for Scholarly HTML.