The Ontogenesis Knowledgeblog: Lightweight publishing about semantics, with lightweight semantic publishing

Phillip Lord
Newcastle University
Newcastle-upon-Tyne, UK

Simon Cockell
Newcastle University
Newcastle-upon-Tyne, UK

Daniel C. Swan
Newcastle University
Newcastle-upon-Tyne, UK

Robert Stevens
University of Manchester
Manchester, UK

ABSTRACT

The web has moved from a minority interest tool to one of the most heavily used platforms for publication. Despite originally being designed by and for academics, it has left academic publishing largely untouched; most papers are available on-line, but in PDF and are most easily read once printed. Here, we describe our experiments with using commodity web technology to replace the existing publishing process; the resource describing ontologies that we have developed with this platform; and, finally, the implications that this may have for publishing in a semantic web framework.

1. INTRODUCTION

The Web was invented around 1990 as a light-weight mechanism for publication of documents, enabling scientists to share their knowledge, in the form of hypertext documents. Although scientists and later most academics, like the rest of society, have made heavy use of the web, it has not had a significant impact on the academic publication process. While most journals now have websites, the publication process is still based around paper documents or electronic representations of paper documents in the form of a PDF. Most conferences still handle submissions in the same way[1]. Books on the web, for example, are often limited to a table of contents.

For the authors (certainly from our personal experience), the process is dissatisfying; book writing is time-consuming, tiring and takes a number of years to come to fruition. If the book has one or a few authors, it tends to reflect only a narrow slice of opinion. Multi-author collected works tend to be even harder work for the editor than writing a book solo. Books do not change frequently; they are therefore out-of-date as soon as they are available. Authors feel a greater pressure for correctness, as they will have to live with the consequences of mistakes for the many years it takes to produce a second edition; most scientists welcome feedback, but being asked to justify something you wish you had not said becomes tiresome, especially if you are waiting to update it.

For the consumer of the material (either a human reader, or a computer), the experience is likewise limited. Books on paper are not searchable, not easy to carry around, are often not cheap to buy and more commonly very expensive to buy. For the computer, the material is hard to understand, or to parse. Even distinguishing basic structure (where do chapters start, who is the author, where is the legend for a given figure) is challenging.

All of this points to a need to exploit the Web for scientists to publish in a different way than simply replicating the old publishing process. Here, we describe our experiment with a new (to academia!) form of publishing: we have used widely-available and heavily used commodity software (Wordpress [6]), running on low-end hardware, to develop a multi-author resource describing the use of ontologies in the life sciences (our main field of expertise). From this experience, we have built on and enhanced the basic platform to improve the author experience of publishing in this manner. We are now extending the platform further to enable the addition of light-weight semantics by authors to their own papers, without requiring authors to directly use semantic web technologies, and within their own tool environment. In short, we believe that this platform provides a "cheap and cheerful" framework for semantic publishing.

2. THE REQUIREMENTS

The initial motivation for this work came from our experience within the bio-ontology community. Biomedicine is one of the largest domains for use of ontology technology, producing large and complex ontologies such as the Gene Ontology [27] or SNOMED [26].

As an ontologist, one of the most common questions that one has is: 'where is there a book or a tutorial that I can read which describes how to build an ontology?'. Currently, there is some tutorial information on the web, there are some books; but there is not a clear answer to the question.

[1] Even conferences about the use of Web technologies!

Many of the books are collections of research-level papers, or are technologically biased. Currently many ontologists have learned their craft through years reading mailing lists, gathering information from the web and by word of mouth. We wished to develop a resource with short and succinct articles, published in a timely manner and freely available.

We wished, also, however to retain the core of academic publishing. This was for reasons both pragmatic, principled and political. Consider, for example, Wikipedia, that could otherwise serve as a model. Our own experience suggests that referencing Wikipedia can be dangerous: it can and does change over time meaning critical or supportive comments in other articles can be "orphaned". Wikipedia maintains a "neutral point-of-view" which, many are of the opinion, makes it less suitable for areas where knowledge is uncertain and disagreement frequent. Finally, Wikipedia is relatively anonymous in terms of authorship: whether this affects the quality of articles has been a topic of debate [16], but was not our primary concern; pragmatically, the promotion and career structure[2] for most academics requires a form of professional narcissism; they cannot afford to contribute to a resource for which they cannot claim credit. Of course, our experiences may not be reflective of the body academic overall; there has, for example, been substantial discussion of the issues of expertise on Wikipedia itself [7]. Although the reasons may not be clear, it is clear that academics largely do not contribute to Wikipedia, and that Wikimedia sees this as an issue [15].

We also had an explicit set of non-functional requirements. We needed the resource to be easy to administer and low-cost, as this mirrored our resource availability; authors should be offered an easy-to-use publishing environment with minimal "setup" costs, or they would be unlikely to contribute; readers should see a simple, but reasonably attractive and navigable website, or they would be unlikely to read.

3. THE ONTOGENESIS EXPERIENCE

Our previous experience with the use of blog software within academia was limited to "traditional" blogging: short pieces about either: the process of science (reports about conferences, or papers for example); journalistic articles about other peoples research; or, personal blogging, that is articles by people who just happen to be academics. Although we wished to develop different, more formal content, this experience suggests that many academics find blogging software convenient, straight-forward enough and useful.

To test this, we decided to hold a small workshop of 17 domain experts over a two day period, and task them with generating content, conduct peer-review of this content and publish it as articles on a blog.

3.1 Terminology and the Process

Like many communities, the blogosphere has developed its own and sometimes confusing terminology. To describe the process we adopted we first describe some of this terminology. A *blog* is a collection of web pages, usually with a common theme. These web pages can be divided into: *posts* that are published (or *posted*) on an explicit date and then unchanged; and *pages* that are not dated and can change. Posts and pages have *permalinks*: although they may be accessible via several URLs, they have one permalink that is stable and never changes. Posts and pages can be *categorised* – grouped under a predefined hierarchy – or *tagged* – grouped using *ad hoc* words or phrases defined at the point of use. A blog is usually hosted with a *blog engine*, such as *Wordpress* that stores content in a database, combines it with style instructions in *themes* to generate the pages and posts. Most blog engines support extensions to their core functionality with *plugins*. Most blogs also support *comments* or short pieces of content added to a post or page by people other than the original authors. Most blog engines also support *trackbacks* which are bidirectional links: normally, a snippet from a linking post will appear as a comment in the linked to post. Trackbacks work both within a single blog and between different distributed blogs. Many blogs support *remote posting*: as well as using a web form for adding new content, users can also post from third party applications, through a programmatic interface using a protocol such as XML-RPC or even by email. Posts and pages are ultimately written in headless HTML (that part of HTML which appears inside the `body` element), although the different editing environments can hide this fact from the user.

Our initial process was designed to replicate the normal peer-review process, with a single adjustment, that peer-review was open and not blind: papers would be world-visible once submitted; the identities of reviewers would be known to authors; all reviews would be public. We adopted this approach for pragmatic reasons. Wordpress has little support for authenticated viewing and none for anonymisation. The full process was as follows:

- Authors write their content and publish using which ever tooling they find appropriate.
- The author posts their content, categorising it as *under review.*
- An editor assigns two reviewers.
- Reviewers publish reviews as posts or comments. Reviews link to articles, resulting in a trackback from article to review.
- The author modifies the post to address reviews.
- Once done to the editors satisfaction, the post is re-categorised as *reviewed.*

Our expectation was that following this process, articles would not be changed or updated; this is in stark contrast to common usage for wiki-based websites. New articles could, however, be written updating, extending or refuting old ones.

3.2 Reflections on the Ontogenesis K-Blog

Our initial meeting functioned to 'bootstrap' the Ontogenesis K-Blog. This was useful to acquire a critical mass of content, but also, on this first outing, to explore the K-Blog process and technology. The setup for the day was the vannilla Wordpress installation. The day started with

[2] We use the term "career structure" in the loosest sense

a short presentation on the K-Blog manifesto [21] and an overview of the process, including authoring and reviewing. The guidelines to authors were to write short articles on an ontology subject (a list of suggestions was offered and authors also made their own choices) and to produce the article in whatever manner they felt appropriate. There was a certain level of uncertainty among authors as to the K-Blog process (partly because one of the objectives of the meeting was to 'force out' the process) and this, naturally, pointed to the need to document the K-Blog process so that authors could have the typical 'instructions to authors'.

This first meeting produced a set of 20 completed and partially completed articles. Some even had reviews. Even on the day itself there was some external interest seen from Twitter. The first external blog post (outside of those produced by attendees) happened during the meeting [18] with a second shortly after [17].

We also held a second content provision meeting and together these generated a collection of articles that felt like an academic book in terms of content, but generated with considerably less effort. This experience was also sufficient to gather requirements on how to improve the K-Blog idea. A useful K-Blog on the K-Blog process itself was produced by Sean Bechhofer [12]. There is also a K-Blog looking back on the first year of the Ontogenesis K-Blog [22].

Several requirements emerged with respect to **authorship**. The principle of the short, more or less self-contained article was attractive (though the audience were somewhat self-selecting). Authoring directly in the editor provided by Wordpress was felt to be poor by those that tried it. Authoring in a favourite editing tool and then publishing via Wordpress worked reasonably well for most authors. There were, however, a variety of issues with the mechanism of this style of publishing; referring to articles that will be, but have not yet, been written. To some extent this was an artefact of the day (many articles being written simultaneously), but authors needed to refer to glossaries and articles in progress.

One stylistic issue was the habit of putting full affiliations at the top of an article. The ontogenesis theme presents the first few lines when displaying many articles, but in many cases this was simply showing the title and author affiliation; where it would be more useful to have the first sentence or so of the article itself.

For the whole K-Blog, a table of contents was felt to be important. This would give an overview of contents and a simple place for navigation about the K-Blog. This raised the issue of **attribution**; the table of contents needed to expose the authors, including multiple, ordered authors. This is not an unsurprising need, as the authors' scientific reputation is involved. In this vein, making K-Blog articles citable by issuing of Digital Object Identifiers (DOI) was requested.

For scientific credibility, the ability to handle **citations** easily was an obvious requirement. Natively, Wordpress has little or no support for styling citations and references. The ability to cite via DOI and, in this field, PubMed identifiers to automatically make links and produce a reference list was felt to be important. Also, having the Ontogenesis K-Blog articles in PubMed would also be attractive to authors.

The last **authorship** issue was the **mutability** of articles. One aim of K-Blog is to enable articles to change in the light of experience and scientific development, as well as a procedural requirement for updates following review. There was felt to be a conflicting need for articles not to change, so that comments and links from other documents work in the longer term.

The last significant issue was the **reviewing** of articles. The aim was to have this managed by authors choosing reviewers (with editorial oversight). On the Ontogenesis K-Blog day this could work with authors calling across the room for a review. This is, however, not a sustainable approach. Wordpress, however, lacks tracking facilities to manage the reviewing process, whether this is done by an author or an editor. The realisation that such management support is needed is not the greatest insight ever gained, but the requirement is there even in a light weight publishing mechanism.

4. IMPROVEMENTS TO THE TECHNOLOGY

Our initial experiment with the ontogenesis K-Blog suggested a significant number of issues with the use of Wordpress for scientific publication. In this section, we describe the extensions that we have made or used to the publication process, documentation or to Wordpress itself. Following our initial experience with Ontogenesis, we have started to trial these improvements, including through another workshop which resulted in a new K-Blog [11], describing the scientific workflow engine Taverna [23]; work is also in progress on the use of a K-Blog for bioinformatics [**?**], and another for public healthcare [2].

Currently, we have 11 plugins extending the basic Wordpress environment. For completeness, all of these are shown in Table 1. Our theme is also extended in some places to support the plugins. In general, the plugins are orthogonal and will work independently of each other. One advantage of using Wordpress is that many of these plugins are freely available, written and maintained by other authors; while other academic publication environments, such as the Open Journal System [4] exist and are relatively widely-used, but Wordpress is used to host perhaps 10% of the web, making the plugin ecosystem extremely fertile.

Reviewing: The initial process was self-managed and required two reviews per article; this was found to be cumbersome. We have addressed this in two ways; first, we have defined a number of different peer-review levels (public review, author review, editorial review [14]), including a lightweight process now being used for Ontogenesis; authors now select their own reviewers, and decide for themselves when articles are complete. Second, we have added software support. Initially, we attempted to use RequestTracker – an open source ticket system, but found the user interface too complex for this purpose. We are now using the EditFlow plugin to Wordpress that was designed for managing a review process—albeit a hierarchical rather than peer-review process.

Plugin	Use	URL
Co-Authors Plus	Allows K-Blog posts to have more than one author	http://wordpress.org/extend/plugins/co-authors-plus/
COinS Metadata Exposer †	Provides COinS metadata on K-Blog posts (used by Zotero, Mendeley etc)	http://code.google.com/p/knowledgeblog/
Edit Flow	Gives editorial process management infrastructure	http://editflow.org/
ePub Export	Exports K-Blog posts as ePub documents	http://wordpress.org/extend/plugins/epub-export/
KCite *	Automatic processing of DOIs and PMIDs into in-text citations and bibliographies	http://knowledgeblog.org/kcite-plugin
Knowledgeblog Post Metadata Plugin *	Exposes generic metadata in post headers	http://code.google.com/p/knowledgeblog/
Knowledgeblog Table of Contents *	Produces a table of contents based on a category of articles. Posts are listed with all authors	http://knowledgeblog.org/knowledgeblog-table-of-contents-plugin
Mathjax LaTeX*	Enables use of TeXor MathML in posts, rendered in scalable web fonts	http://knowledgeblog.org/mathjax-latex-wordpress-plugin
Post Revision Display	Publicly exposes all revisions of an article after publication	http://wordpress.org/extend/plugins/post-revision-display/
SyntaxHighlighter Evolved	Syntax Highlights source code embedded in posts	http://wordpress.org/extend/plugins/syntaxhighlighter/
WP Post to PDF	Allows visitors to download posts in PDF format	http://wordpress.org/extend/plugins/wp-post-to-pdf/

Table 1: Wordpress plugins employed by K-Blog. Plugins marked with * are written by the authors. Plugins marked with † are modified by the authors.

Authoring Environment: The standard Wordpress editor was found impractical by most authors, even for short articles. Wordpress does provide "paste from word" functionality, but this removes all formatting which defeats the point. While the lack of a good editing environment could have been a significant problem, our subsequent experimentation has shown that it is possible to post directly from a wide variety of tools, including "office" tools such as Word, Google Docs, LiveWriter and OpenOffice. This is in addition to a variety of blog-specific tools and text formats (such as asciidoc), which are suitable for some users. We have added documentation to a kblog (`http://process.knowledgeblog.org`) to address these. In practice, only LaTeX proved problematic having no specific support. To address this, we have produced a tool called **latextowordpress**; this is an adaptation of the plasTeX tool, a python based TeX processor, to produce simplified HTML appropriate for Wordpress publishing. Our experience with using the tools is that while none are perfect, sometimes requiring "tweaking" of HTML in Wordpress, most reduce publishing time to seconds.

Citations: We have addressed the lack of support for citations within Wordpress with a plugin called **kcite**. This allows authors to add citations into documents as `shortcodes` with either a DOI or Pubmed ID (other identifiers can and are being added to kcite). Shortcodes are a commonly used form of markup of the form:`[tag att="att"]text[/tag]`; they are often found where a simplified HTML-like markup is desired. A bibliography is then generated automatically on the web server. Requiring authors to add markup to otherwise WYSIWYG tools is damaging to the user experience. We believe that this is soluable, however, by extending bibliographic tools, by developing a "kcite" style-file or template; we have a prototype of this (using CSL [9]) for Zotero and Mendeley, and another for asciidoc with bibtex. It is also possible to just use native tool support in Word or LaTeX, and convert bibliographies to HTML; the disadvantage with this approach is discussed later.

Archiving and Searching: Archiving is primarly a social, rather than technological, problem. A blog engine is fully capable of storing content in the long-term, but authors and readers have to believe that it will do so. As a novel form of academic publishing, K-Blog is not automatically archived by as a scientific journal. However, we have taken advantage of its web publication; the main K-Blog site is now explicitly archived by the UK Web Archive, as well as implicitly by other web archives. We have enhanced the website with an "easy crawl" plugin–that is a single web page pointing to add articles classified as reviewed. We now support the (technical) requirements for LOCKSS and Pubmed. Simultaneously, this also enhances the searchability of K-Blog, fulfilling the requirements for Google scholar.

Non-repudiability: The K-Blog process does not allow authors to make semantically meaningful changes after an article has been reviewed. Unfortunately, it is hard to define "semantically meaningful" computationally, so we have made no attempt to address this by locking articles; rather, all versions of articles are now accessible to the reader (Wordpress provides this facility to the authors by default). This enables community enforcement of a no-change policy.

Multiple Authors: We believe that authoring is best done outside Wordpress. This also means that we do not support multiple-authorship; we have made no attempt to add collaborative features to Wordpress. However, we did need articles to carry a byline attributing the articles to multiple authors; although not critical to the functioning of a K-Blog, it is socially critical to appease the professional narcissism (see Section 2) of scientists. Fortunately, this is a common requirement, and a suitable Wordpress plugin existed.

Identifiers: Wordpress already supports permalinks; although we believe that URLs are entirely fit for purpose technologically while DOIs do little other than introduce complexity [10], K-Blog required DOIs for professional narcissism. We considered becoming an DOI authority, but this proved impractical. Instead, we have used DataCite [1]. This has required a small extension to Wordpress to extract appropriate metadata and to store the DOIs once minted.

Metadata: K-Blog now uncovers various parts of its metadata in a number of ways; unfortunately, there appear to be a large number of (non-)standards in use, each with its own application. K-Blog currently provides: COiNS, enabling integration with Zotero and Mendeley; meta tags for Google Scholar; and Dublin Core tags for no specific reason than completeness. We are in the process of providing bibtex export (for bibtex!), and a JSON representation to support citeproc-js [13] in the second generation of kcite.

Mathematics and Presentation: We have also provided several pieces of technology that did not stem from concrete requirements arising from the initial Ontogenesis meeting. We have improved parts of the presentation system by adding, for example, syntax highlighting to code blocks. Additionally, we have created the **mathjax-latex** plugin enabling the use of TEX(or MathML) markup in posts that are then rendered in the browser using scalable fonts. Wordpress has native math-mode TEX support, but using image fonts which do not scale and have an ugly pixelated display.

5. DISCUSSION

We have been motivated by a lack of enthusiasm for traditional book publishing to devise another mechanism by which we can achieve the same ends. We wished to avoid the downsides of an "all or nothing" approach to creating a "static" paper document that is read by relatively few people due to price. The K-Blog approach allows authors to publish in a piecemeal fashion; writing only that which they are motivated to write using a mechanism that avoids a third party making arbitrary decisions on formatting with peculiar time-scales.

To avoid all this, the K-Blog is a light-weight publishing process based on commodity blogging software. We have taken an approach of writing short articles around a theme of 'ontology in biology'; the Ontogenesis K-Blog. At the time of writing we have 26 articles and page viewing numbers that are pleasing (see Figure 1). These statistics are generated by Wordpress directly, and represent (an approximation of) "real" page reads, with robot and self-viewing removed. This is confirmed by the ten most read articles (Table 2) that reflect our expectations – "What is an ontology" being first. In this sense, we consider the K-Blog process to be a success, especially when considered against the circulation of an equivalent book.

The social processes with K-Blog are largely similar to traditional publishing, with one exception – reviewing is public. While we may have been interested in experimenting with this for principled reasons, in practice we adopted it because we did not know how to support blind anonymous review with Wordpress. Open review is not a new idea: Request For Comments are common in standards processes; both

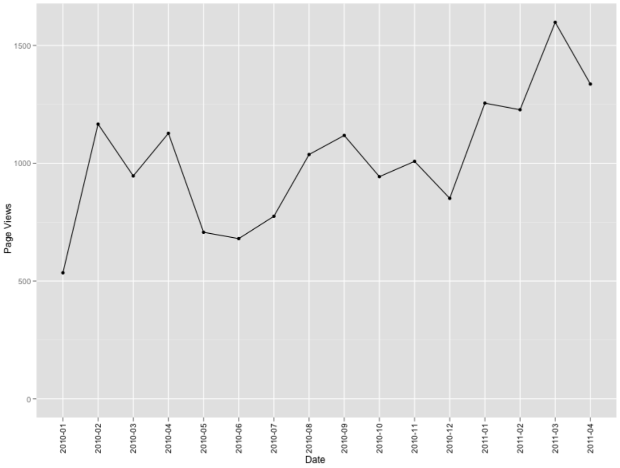

Figure 1: Month page view statistics for the Ontogenesis K-Blog.

What is an ontology?	1,737
OWL Syntaxes	1,246
Ontology Learning	882
Table of Contents	740
What is an upper level ontology?	684
Reference and Application Ontologies	630
Protegé& Protegé-OWL	522
Semantic Integration in the Life Sciences	517
Automatic maintenance of multiple inheritance ontologies	469
Ontologies for Sharing, Ontologies for Use	330

Table 2: Most Viewed articles for the Ontogenesis K-Blog (Totals).

Nupedia [3] (the fore-runner of Wikipedia) and H2G2 [5] (which predates Nupedia) use public peer-review. It is still, however, unusual in academia. In our experience from Ontogenesis, it raised no worries from among our contributors, except that reviewers often wanted to be more involved in the proofing, a role normally played by authors low down the author list; open review processes blurs these lines somewhat.

One open area for the discussion is the extent to which authors can, should be and wish to change articles after publication. While the ability to update is inherent in the web, the desire for non-repudiability was considered to be important; the contradiction here appears fundamental, and we do not feel we have reached a good compromise yet. In one sense, our use of the post-revision display plugin solves this problem; even if the article changes, it is still possible to refer to a specific version. However, like all automated versioning tools, many versions get recorded often with very fine-grained changes, which makes selection of the "right" version hard to impossible. We could replace this with an explicit versioning tool, similar to a source code versioning system; but these systems are hard-to-use for those unused to them, as well as being difficult to implement well. An environment like K-Blog, however, does allow rapid publication of and bi-directional linking with articles; combined with typed linking with CiTO, the ability to publish erratum, addendum and second editions may be a better solution.

Our experiences with K-Blog, we think, are useful in understanding how semantic web technology can and will impact on the publication and library process. Both from our initial work with Ontogenesis, and subsequent work with `http://taverna.knowledgeblog.org`, it has become obvious that good tool support is critical. 'Good' in this sense can be straight-forwardly interpreted as "familiar" that in general can be interpreted as MS Word. Our choice of a blogging engine here was (unexpectedly) well-advised, as this form of publication is already supported by many tools. It is also clear that there are many other tools that could be added; while Ontogenesis has the content, for example, that might be found in an academic book, it does not currently have the presentation of the book. Articles are already available as ePUB, and more recent work has used our Table of Contents plugin to provide a single site-wide ePUB of all articles [24]. Pre-existing tools such as Anthologize [8] may also be useful for adding organised collections of articles gathered from the whole.

This has a direct implication on the addition of further semantics to content. On the positive side, the use of Wordpress makes semantic additions plausible in a way that many conventional publishing processes do not. For example, the publication of our (PWL, RS) recent paper [19] required conversion from the LaTeX source to PDF (by latex), to another PDF, to a MS Word file (by hand), to XML before arriving at the final HTML form. This process took many weeks, required multiple interactions between the authors and publisher. It still failed to preserve the semantic use (to humans) of Courier font highlighting in-text ontology terms and requiring post-publication correction. The equivalent blog post [20] gave us nearly instantaneous feedback on the final form, allowing us to check that the semantics was present and correct.

The requirements for semantics have, however, to be light. We have concentrated throughout K-Blog on the ease of delivery of content; even with this focus, it is hard. In most cases, asking for more work, for more semantics than authors are used to giving in papers is problematic. For example, I (PWL) attempted to add microformat-based markup to Ontogenesis, again, identifying ontology terms. So far, all article authors have ignored this markup (including, embarrasingly, myself).

One solution to this issue is to ensure that authors themselves benefit directly from extra semantics. For example, the Mathjax-Latex plugin allows Wordpress to present mathematics in TeX or MathML markup in the final document, which is more semantically meaningful than the default Wordpress behaviour of rendering an image. From the authors perspective, it also enables the use of TeX markup in Word, and the end product scales and looks less ugly on the web page.

With Kcite, we allow the user to embed DOIs or Pubmed IDs; this can be achieved at no cost to the user, if they already use a bibliography tool, as it can transparently produce citations for them using Kcite shortcodes. Development versions of Kcite already allow easy switching of bibliographic style that we hope will become at the option of the author (rather than the website or publisher as is currently the case), and/or the reader. With this additional information, we can also embed more semantics into the end document at no additional cost to the author, using for example the least specific CiTO `cites` term. However, further use of CiTO that will require the author to decide which term to use, with relatively little gain to themselves, and may require extension to bibliographic tools if we are to maintain transparency of Kcite shortcodes; even if the tools are present, it is unclear whether authors will use them. We note that semantics useful to domain authors is likely to be domain-specific; mathematicians are more likely to care about maths presentation, but less likely to care about Pubmed IDs. We need to be able to extend the publishing model and environment for different journals to cope.

From a technological perspective, we have found the use of shortcodes to be a good mechanism for readers to add semantics. They are simple and relatively easy to understand. In some cases they can be hidden from the user entirely; forcing users to add markup to otherwise WYSIWYG environments such as MS Word is best avoided. Although the direct use of a more standard XML markup would seem more sensible, in practice it requires tool support, as XML markup will be escaped by helpful remote posting tools. Extension of remote posting tools is hard (for tools like MS Word) or impossible (for cloud tools such as Google Docs or LiveWriter). A blogging engine such as Wordpress makes it trivial to replace shortcodes both with a *presentation* format and machine interpretable *microformat*; for example, the development version of Kcite transforms DOI short codes (`[cite]10.232/43243[/cite]`) into in-text citations (Smith et al, (2002)) embedded in a span tag (`<span kcite-id="10.232/43243">Smith et al,` (2002)</span>) that are subsequently transformed into final presentation form within the browser using Javascript. The presentation form can also support additional semantic markup such as CiTO [25].

Although we believe that additional semantics are a good thing, we will not enforce a requirement for additional semantics on authors. If authors choose not to use kcite, then this is their choice. We need to show that they are useful. Our experience with many (non)standards such as CoINS, DOIs, OAI-ORE, LOCKSS is that they are not simple, speaking primarily to publishers or librarians. For a semantic web approach to work, it must focus on authors and readers, as they produce and consume the content. Extracting even light-weight semantics even from authors who are ontology experts is hard. For other domains, the situation may be worse.

Current publishing practices make use of semantic web technology impractical; semantics added by authors are unlikely to be represented correctly if the end product is a PDF typeset by hand. More over, we can see little point adding semantics to individual articles if this is done in a bespoke way. With K-Blog, we have focused on providing both content, and a full process, with review, using existing tools and workflows, adding semantics secondarily or incidentally where we can. As a result, the level of semantics that we have achieved is light-weight. However, we believe that K-Blog and Wordpress combined with associated tooling provides all the basic requirements for a publishing process, and that it provides an attractive framework on which to build a semantic web.

Acknowledgements

We would like to acknowledge the contribution of the authors of articles for both the Ontogenesis and Taverna K-Blog, whose feedback was essential for this process. K-Blog is currently funded by JISC.

6. REFERENCES

[1] Bioinformatics. http://bioinformatics.knowledgeblog.org.

[2] Datacite. http://datacite.org/.

[3] Health and Public Health. http://health.knowledgeblog.org.

[4] Nupedia. http://en.wikipedia.org/wiki/Nupedia.

[5] Open Journal System. http://pkp.sfu.ca/?q=ojs.

[6] The Guide to Life, the Universe and Everything. http://www.bbc.co.uk/h2g2/.

[7] Wordpress. http://www.wordpress.org.

[8] Wikipedia:expert retention, 2008. http://en.wikipedia.org/wiki/Wikipedia:Expert_retention.

[9] Anthologize, 2010. http://anthologize.org/.

[10] Citation style language, 2010. http://www.citations-styles.org.

[11] The problem with DOIs, 2011. http://www.russet.org.uk/blog/2011/02/the-problem-with-dois/.

[12] The Taverna Knowledgeblog, 2011. http://taverna.knowledgeblog.org.

[13] Sean Bechhofer. Reflections on blogging a book. Ontogenesis, 2011. http://ontogenesis.knowledgeblog.org/647.

[14] Frank Bennett. Citeproc-js. https://bitbucket.org/fbennett/citeproc-js/wiki/Home.

[15] Simon Cockell, Dan Swan, and Phillip Lord. Knowledgeblog types and peer-review levels. Process, 2010. http://process.knowledgeblog.org/archives/19.

[16] Zoe Corbyn. Wikipedia wants more contributions from academics, 2011. http://www.guardian.co.uk/education/2011/mar/29/wikipedia-survey-academic-contributions.

[17] Casper Grathwohl. Wikipedia comes of age. The Chronile of Higher Education, 2011. http://chronicle.com/article/article-content/125899/.

[18] D. Kell. Metabolomics, food security and blogging a book, 2010. http://blogs.bbsrc.ac.uk/index.php/2010/01/metabolomics-food-security-blogging-book/.

[19] Jim Logan. What is an ontology? | ontogenesis, 2010. http://ontogoo.blogspot.com/2010/01/what-is-ontology-ontogenesis.html.

[20] Phillip Lord and Robert Stevens. Adding a little reality to building ontologies for biology. *PLoS One*, 2010.

[21] Phillip Lord and Robert Stevens. Adding a little reality to building ontologies for biology, 2010. http://www.russet.org.uk/blog/2010/07/realism-and-science/.

[22] Phillip Lord and Robert Stevens. The Ontogenesis Manifesto, 2010. http://ontogenesis.knowledgeblog.org/manifesto.

[23] Phillip Lord and Robert Stevens. Ontogenesis: One year one. Ontogenesis, 2011. http://ontogenesis.knowledgeblog.org/1063.

[24] Tom Oinn, Mark Greenwood, Matthew Addis, M. Nedim Alpdemir, Justin Ferris, Kevin Glover, Carole Goble, Antoon Goderis, Duncan Hull, Darren Marvin, Peter Li, Phillip Lord, Matthew R. Pocock, Martin Senger, Robert Stevens, Anil Wipat, and Chris Wroe. Taverna: lessons in creating a workflow environment for the life sciences: Research articles. *Concurr. Comput. : Pract. Exper.*, 18:1067–1100, August 2006.

[25] Peter Sefton. Making epub from wordpress (and other) web collections, 2011. http://jiscpub.blogs.edina.ac.uk/2011/05/25/making-epub-from-wordpress-and-other-web-collections/.

[26] David Shotton. CiTO, the Citation Typing Ontology. *Journal of Biomedical Semantics*, 1(Suppl 1):S6, 2010.

[27] M.Q. Stearns, C. Price, K.A. Spackman, and A.Y. Wang. SNOMED clinical terms: overview of the development process and project status. In *AMIA Fall Symposium (AMIA-2001)*, pages 662–666. Henley & Belfus, 2001.

[28] The Gene Ontology Consortium. Gene Ontology: Tool for the Unification of Biology. *Nature Genetics*, 25:25–29, 2000.