What open source webpublishing software has the scientific community for e-journals?

Roughly one year ago the electronic journal Ikaros decided to upgrade its web-publishing system. We have reviewed several open-source applications and it was a bitter surprise that at the beginning of 2006 there was no suitable solution for us – of course, there were many content management systems for magazines, newsletters, sites or simple collections of web pages, but only a few ones for the publication of the electronic journal.

Situation has changed since then, but at the beginning of 2006 we found no better solution than to develop our own system [4]. It was built on top of a general Content Management System (CMS) and was made available to the open-source community that accepted it. It already has several installations around the world and it could be interesting to describe how the CMS allowed us to build the e-journal system so quickly and efficiently. Yet perhaps that is another story. Purpose of this paper is to compare specialised open-source webpublishing systems for e-journals. I would like to return one year back and see if the situation has changed. I would like to focus on those web publishing systems which are peer reviewed, because especially for the peer reviewed journals there existed only few open source answers.

Peer review workflow.

First, let us briefly summarize what is a peer review. It is a process of quality control in which submitted papers are evaluated by experts in the field and if approved papers are published. If not approved, articles need to be corrected or can be rejected completely. Web publishing systems implement a workflow to facilitate this process. The web interface allows authors and referees to upload their submissions and reviews. Editors and editorial staffs interact with the system assigning reviewers and preparing the issue based on their feedback. Most systems use concepts of 'roles', 'queues' and 'actions' and only appropriate users are allowed to intervene and only in certain situations. For example, a referee will see only the papers with which s/he is associated and is allowed to access, but journal editors will see all those papers and many more just to control and assign other tasks to other people. The workflow of such systems is largely a predetermined path which starts at the submission of the paper and usually ends with the publication of the paper in one of the issues of the journal [adapted from 5, pp. 10-11].

Open source solutions

In 2005, Mark Ware [5] conducted a comprehensive survey among journals and also (not so detailed) analysis of available publishing systems. It was interesting that out of 11 available peer review systems on the market, there was only one open source software present. Approximately one year later, we found two more systems, one was not yet released to public, and the other was unsuitable for the needs of the e-journal Ikaros. The situation today still does not seem different from 2005. Given the strength, activity and history of the open source movement, it is strange that there is, or at least, there was such a gap in the web publishing software.

The oldest surviving open source project was set up in 2001, which is at least 10 years after the commercial systems which inevitably grew more mature and sophisticated. I cannot give a full account of this “gap”, but one can speculate it was too early for open source systems – Budapest Open Access Initiative was in 2001, Berlin declaration in 2003 and exactly at that time the two open source players have their roots [3,7]. It is relatively few years that the scientific community has publicly announced its discontent with practices of publishers and started to interfere into the traditional publishing cycle with archives of preprints such as arxiv.org, digital repositories or OAI-PMH. Many other variables might be affecting this publishing game and its software, but if nothing else, things started to move faster.

If you go searching for open source peer review publishing systems now, you might identify these few: DPubS <http://dpubs.org>, GAPworks <http://gapworks.berlios.de>, Hyperjournal <http://www.hjournal.org>, ePublishing Toolkit <http://dev.livingreviews.org>, SOPS <http://www.scix.net/sops.htm>, TOPAZ <http://www.topazproject.org>, projas <http://sourceforge.net/projects/projas> and possibly one or two others. After a closer look you will realize that many of those systems are either discontinued or poorly maintained though. For instance the Slovenian SOPS which was a pilot operating in 2002-2004 had only one journal installation. Another example is the German GAPworks, updated for the last time in June 2006, with a low version number, without any documentation and even installation instructions. Projas is relatively new as it has been launched in may 2006 but from the sourceforge.net statistics you can notice it did not receive a lot of attention, has low vitality, little documentation and it is really a simple application. We could continue until hjournal which has some documentation, it has been released with version number 0.5 and seems to be extended. This project is a living one, but appears to be hanging on line between development and stall – its last update was released almost 10 months ago and the number of known installations is very low indeed – there is one Italian e-journal, which itself does not seem to be very active. Consequently we may wonder what is left?

In 2006 there were two and in 2007 there are three choices – DPubS, ePubTk, OJS. Other systems do not seem to have a bright future and one cannot advise users to waste energy on them, not only because they do not have big institutions behind them (this is not true for the german system backed by the GAP – German academic publishers http://www.gap-portal.de/) but mainly because they are way behind other open source systems, they do not have a large community of users and are not actively developed.

About the three systems

Let us start with DPubS – it has been announced two years ago as a Cornell library's initiative for open access publishing. It received a lot of attention and press releases – so much, that even if it had not released any code until May 2006, it was ranked among the top ten in search engine results for e-journal and was included in many software listings.

DPubS has its roots in a Project Euclid where Cornell library is acting as a publisher and service provider, maintaining publishing platform for some 50 mathematical journals from different publishers around the globe. Project Euclid has its own publishing system and DPubS should be its open source port. DPubS is written in a scripting language Perl, implements Dienst architecture of distributed services and, owing to this, can be easily integrated with other systems (such as DSpace, Fedora and various digital libraries). They can "talk to each other" through an elaborate system of APIs (Application call interface) and such system can grow easily. It was designed with flexibility in mind, so another services (possibly in different programs in language) can talk&agree with DPubS on actions which should be performed.

As a confirmation DPubS offers you the choice between two (not internal, but integrated) indexing engines – either a perl program called freewais or a feature-ladden and extremely powerful Java search engine called Lucene. These systems are external to DPubS and show good examples of integration and reuse of open source softwares. (As we will see later, the level of integration will be even higher for the next system.)

DPubS has more than a nice architecture, yet let me focus on its dark sides – apart from a few not-so-important weak points, there is one quite important weakness (hope exists it is only temporary). At the time of writing this article, DPubS is unfinished – it might be functional as a repository but does not have implemented the peer review workflow. This is visible from the close inspection of the source code and I am sure there are many features missing in the current version of DPubS (released 30 October 2006). We can get confirmation even from the wiki documentation which says: “The system **currently** has administrative interfaces for **basic** tasks, such as submitting content, loading content, and setting user privileges. Development work is continuing on additional administrative functions and on editorial support tools. This functionality will likely be added beginning in early 2007.” [7, emphasis added by the author].

To return to the less important weak points, we could mention the lack of documentation – even the technical documentation is scarce – there is one e-mail lists with 50 posts, which means you cannot find a big community around DPubS (and no support from them). Last but not least – installation of the DPubS is not easy, you must have the appropriate hardware and the technical expertise. Universities and publishers can use their existing infrastructure but others would have to invest a little more.

Nevertheless, DPubS is a very interesting system and if the Cornell library continues to develop it (possibly faster than in the pace of the last two years), then it could be useful to those who need a flexible, powerful, and configurable web publishing system for peer review journals.

ePubTk

ePubTk was developed and maintained by Max Planck Society (Heinz Nixdorf Center for Information Management) since 2002/2003. It is used to operate a family of publications called Living reviews (4 in total). ePubTk is written in the Python scripting language and its license allows for free use only to non-commercial projects; other institutions and individuals would have to pay licensing fees. ePubTk is a tool for publishing LaTeX authored documents and therefore is specially suited for natural sciences such as mathematics or physics where LaTeX is a publishing standard. Because of LaTeX ePubTk uses various other applications such as ImageMagick or tex4ht to make format conversion, it is in this regard unique as to the quality and complexity of the process. The system also interacts with other external applications, such as Lupy, a search engine, (which is a Python port of Lucene, by now discontinued), mx toolkit and other softwares for text manipulation, subversion management and database interaction.

The installation of such system is not really a trivial task, especially on Windows because ePubTk was developed primarily for Linux. Whatever platform you choose, it requires considerable technical expertise even to make the system work in a default configuration, not mentioning occasional upgrade and bug-fixing of all its components.

The main difference from the previous DPubS is that ePubTk was designed for a special type of publications – so called living reviews, “invited papers” updated even after their publication. These “invitedness” and “liveliness” are the unique features that distinguish the system from other web publishing options. Its workflow is tailored to situations where authors do not submit articles themselves but rather the editorial board of the journal decides which topics should be covered in the future issues (even though ePubTk does not work with concepts of issues in the traditional way). Topics need to be inserted into the system and assigned to possible authors who are then contacted and invited to write the paper. If somebody accepts the offer, the author is recorded and gets assigned reviewers – And from this point on, the workflow might resemble the situation of most journals where papers are usually not solicited but submitted for review by authors.

I would like to emphasize that it is this peculiar management of reviews which make this software unique but also somewhat special. With it, articles are not organized into sections and issues published, but topics are created and then managed one by one. Of course, we could say that the topic is one article and many articles can be organized into one issue, but all the steps in the introductory phase seem unnecessary for most other journals where invited papers are published only once in a while.

ePubTk is technically very sophisticated and actively developed by the team of Living reviews. It is mainly suitable for special publications, mostly not-high-volume ones where one wants (or can tolerate) the workflow designed solely for invited reviews. I dare to say that this publishing system is not the best for more standard and more general workflows. It could be tailored to the different needs but this would require considerable costs. It is a precise example of an open source software which takes advantage of other different packages and tools, implementing them into a functional unit which however looks confusingly at first sight and may need a lot of “fiddling”.

OJS

Open Journal System was developed in 2001 by Simon Fraser university in Vacouver. It is the “oldest” among the three mentioned systems and unique in several aspects – it is written in yet-another-scripting-language. Before we had Perl, than Python, now it is PHP. Secondly, OJS stores its data not on the filesystem (such as DPubS) or in the PostgreSQL database, but in another database, MySQL. Finally, OJS is more monolitic and in my opinion more robust than the other two systems – because it is older, more tested, does not have so much packages and dependencies to solve (but also not so many functionalities regarding format conversions). It is its “own” from A to Z.

This system has the modest requirements of beyond all the mentioned systems. It can be set up even in the shared environment of the webhosting services. In fact, the installation is not difficult and this could partly explain such a high popularity. OJS works out of the box and, unlike the other two, journal administrators can control settings exclusively through the web interface (for more complex settings file edits are needed though). OJS serves approximately more than 900 journals (it is difficult to obtain exact numbers, everybody can download and start to use the software). What is more important is that the system has been through major and complete rewrite in 2006. According to documentation, it was the result of cooperation with 250 editors of 40 journals in the past 4 years. This rewrite took place at the beginning of 2006 and meant a big step towards usability and flexibility of the system. OJS is designed around a model of “standard” workflow where authors submit a paper, section editors assign reviewers and than interact with the system until the point where article is proofread and copyedited. The article is then ready and published in some of the created issues (issues are prepared by editor in chief). The workflow cannot be changed, and for some publishers it can be too strict or too loose for others. The only way to change before the release 2.1 would be to “hack” the source code of the OJS but now it is different.

Version 2.1 introduced a mechanism of plugins which opens OJS to changes. The system allows developers to register their own plugins that are associated and called within specific tasks/places along the standard pathway and this effectively means that you can extend and tailor OJS without need to track the source code. The set of calls (in jargon called “hooks”) are detailed enough to let you influence all important actions; for example you could interfere into the authorisation or submission process and your actions would still be “legal” - they will fit into the whole framework. I think this change is a proof that OJS developers are carefully listening to their users and the system is actively developed.1

Of course, OJS has its pitfalls, one of the most itching would be the absence of a powerful search. The system does implement a sort of native PHP search engine and if you have ever compared the features and results of such implementation with “real” search engines (i.e. Lucene or freewais) you would have realized that the PHP is not a competitor for them. Even the native fulltext search functions of MySQL database give better results than the PHP search engines.

The lack of more powerful search functionality is sorely, nevertheless OJS would still win in many other areas – it has extensive documentation, both technical and user manuals, and by far the biggest community of users who are possibly ready to help you with problems. You can reach them at the forum and the fact that the new hooks were introduced means they have their say into the development. With their help and feedback OJS is quickly moving towards the point of maturity where OJS could be as good as the commercial systems that exist for more than a decade. OJS might compete with them in the next few years, which also depends on the fact whether more institution will invest their time and resources into its installation and refinement.

To conclude, DPubS system is very interesting and is worth following, yet still not fully operational and not suitable for peer reviewed journals. ePubTk is designed for the special type of publications and possibly not interesting for many e-journals. OJS is the only system which could be recommended to anyone who would like to publish peer reviewed (but also non-reviewed) journals and it is predictable that the more users it attracts, the more features and functions it will have. If you need or want, for some reasons, to use open-source webpublishing system for e-journals and you cannot buy or develop your own, this could be the good choice. And even if you can afford to buy or license some other solution, you should include this system in your analysis. Resources you would spend altering this system may be insignificant compared to what you get for it. And in many cases, you would not need to alter anything.

References:

1. Living reviews. [cited; Available from: http://www.livingreviews.org/index.html.
2. Ikaros : elektronický časopis o informační společnosti. [cited 31-05-2007 23:32]; Available from: http://www.ikaros.cz.
3. Public knowledge project : Open Journal Systems. [cited 31-05-2007 23:30]; Available from: http://pkp.sfu.ca/?q=ojs.
4. E-journal. [cited 31-05-2007 23:32]; Available from: http://drupal.org/project/ejournal.
5. Liwing reviews : development and documentation. [cited 31-05-2007 23:41]; Available from: http://dev.livingreviews.org/.
6. Library, C.U. DPubS - Digital Publishing System. [cited 29-05-2007 20:53]; Available from: http://dpubs.org/development.html.
7. Velden, T. Implementing Open Access : activities of the Max Planck Society and an update on Berlin Process. [cited 29-05-2007 20:19]; Available from: http://www.enssib.fr/divers/liber2004/Velden.pdf.
8. Ware, M., Online Submission and Peer Review Systems. 2005: Assoc of Learned & Professional Soc Publisher.