

John W. Maxwell
Simon Fraser University

Abstract

The British Columbia Association of Magazine Publishers and Simon Fraser University's Canadian Centre for Studies in Publishing are working on an adaptation of Open Journal Systems (OJS) for publishers of small cultural magazines, a community of users with needs that are very different in a wide variety of ways from the standard groups OJS has traditionally served in scholarly publishing.

John Maxwell is Assistant Professor in the Master of Publishing Program at Simon Fraser University, 515 West Hastings Street, Vancouver, BC V6B 5K3. Email: jmax@sfu.ca.

Keywords

OJS; XML; Content management; Magazine publishing; Editorial workflow

CCSP Press

Scholarly and Research Communication

Volume 1, Issue 1, Article ID 010103, 9 pages

Journal URL: www.src-online.ca

Received August 22, 2009, Accepted September 16, 2009, Published December 21, 2009

Maxwell, John W. (2010). The OMMM Project: Toward a Collaborative Editorial Workflow. *Scholarly and Research Communication*, 1(1): 010103, 9 pp.

© 2010 John W Maxwell. This Open Access article is distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc-nd/2.5/ca>), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Introduction

Open Journal Systems (OJS) has been a great success in supporting journal publishing both online and off. Its set of functions – from submissions management and support of the peer-review process to general editorial process modelling and central organization of publication tasks – has supported thousands of publishers in effectively producing scholarly publications. At Simon Fraser University's Canadian Centre for Studies in Publishing, we have been interested in how OJS can be introduced to a different or wider audience: small cultural magazines.¹ The Online Management Model for Magazines (OMMM) project began by asking how this successful free/open source software could help manage the process of moving articles from submission through to production not just for scholarly journals, but for small magazines, too, with which they share a number of characteristics.

We began looking at opportunities to adapt OJS to magazines as early as 2006 (Maxwell, 2007). In spring of 2008, Ana Torres, then executive director at the British Columbia Association of Magazine Publishers (BCAMP), indicated that BCAMP was interested in funding the development of software to handle electronic submissions and editorial workflow and ultimately in exploring online publication for its membership. At that time, we reckoned that the most straightforward route to this goal would be the adaptation of OJS for a magazine publishing context: removing its academic trappings and bringing it more in line with the expectations of small magazine publishers.

After consulting a variety of BC-based magazines, we learned that electronic submissions-handling was indeed a critical need for many small publications. Rather than carrying around mail sacks full of paper manuscripts and trying to keep track of any number of disks, e-mail attachments, and so forth, small magazine publishers dearly wanted a centralized, online submissions process. None of the small magazines we heard from had the capability to handle electronic submission; the only publications with this capability are larger, well-funded ones with well-developed online versions and content management systems in place already.

Online submission seems simple enough, and the OJS software would certainly deliver this functionality, but this apparent simplicity gives way to deeper problems, such as publishers' traditional reliance on Microsoft Word as a writing and content delivery format. OJS typically assumes attached Word .doc files as its submission format, and this neatly reflects users' comfort with Word as a kind of document *lingua franca*, but Word is hardly a panacea. It is a proprietary file format, prone to being changed periodically as Microsoft updates the software. Its extensive feature set gives authors licence to format submissions in dozens of different ways, making the jobs of layout editors and production folk – especially those charged with converting submissions to other formats, such as HTML – rather difficult. Nor is support for Microsoft Word by any means the norm among online content management systems; indeed, there may be considerable advantages to an entirely Web-based submission process. But, that said, most existing magazine (and journal) workflows still assume that writers will be submitting Word .doc files, and as such an online submission system must be able to handle them.²

Regardless of the exact toolset chosen, an editorial management system for BCAMP's members needed to be based on a mature, existing tool, preferably based on free/open source software for the sake of adaptability. While it was technically possible to create a tool from scratch to meet BCAMP's needs exactly, this was not particularly attractive given the up-front capitalization required to fund development and the ongoing problem of maintaining such a system over time, in the face of evolving needs. Adopting a standard platform supported by a mature, existing user/developer community seemed a much safer and more robust approach. We surveyed existing options for such a platform, and OJS seemed the clear leader in publication workflow management.

Adaptation and development of OJS

In the summer of 2008, we began work on OJS 2.2 to adapt it to the needs of small BC magazine publishers. The scope and funding of the project kept our focus on the user interface and the facets of OJS that are customizable in a straightforward manner, as opposed to getting into deeper coding and development – a position reinforced by the need to maintain compatibility with the main OJS development work over time. We did not want to get into a position of forking OJS or otherwise having to take responsibility for code maintenance as new OJS versions emerged; rather, we sought to create a customization “layer” on top of whatever version of OJS was current. As such, our adaptation had three parts:

1. an overall streamlining and simplification of OJS' user interface, removing components that are specific to scholarly/academic journals. This results in two general benefits identified early in the project's planning stages: first, the interface is rendered relatively simpler, shorter, and easier to follow; second, a layer of academic jargon is removed (writers submitting work are not asked to supply their “institutional affiliation,” for instance). This part of the customization was conducted mostly within OJS' set of “smarty” templates.
2. an overall “translation” of the language used throughout the site's user interface so that the terminology and jargon conforms to the norms of the Canadian magazine publishing community rather than the norms of scholarly communication. In most cases, this represented a shift from the formal language of peer-review processes to a more open-ended and inclusive language that was designed to be adaptable to a variety of magazine publishers' editorial processes. We performed a top-to-bottom edit of both the administrative interface and OJS' extensive set of automated e-mail notification templates, resulting in the effective equivalent of a language translation of the software. As OJS supports “internationalization” of the software, facilitating, say, wholesale translation from English to French, our translation effectively produced a version in “magazine-ese” that remains relatively easy to edit further – or revert to the standard English version.
3. specific customizations of the software to suit the needs of specific publications. This layer of customization was done to suit the publications *Room* and *sub-TERRAIN*, and it encompassed setting up editorial user roles for key staff; customizing submissions policy details; arranging proper e-mail notification processes; and customizing unique style/colour/typographic themes for each. This is not very different from the customization of OJS that any journal undertakes when it adopts the software. But in testing our work, we discovered more complexity here than we had anticipated.

The devil in the details

In preparing our adaptation of OJS for *Room* and *sub-TERRAIN* magazines, we drew feedback from magazine editors as well as BCAMP generally. Joy Gugeler from the *Room* editorial collective provided a great deal of initial consultation going into the translation and interface streamlining, and the resulting editorial changes were reviewed by Fiona Lehn from *Room*.

In the fall, with the initial streamlining and language translation complete, we invited *Room's* Joy Gugeler to review the prototype with respect to its fitness to the submissions-handling process that *Room's* editorial collective employs. This resulted in a detailed mapping of *Room's* editorial workflow process to the workflow model embodied in OJS. In order to better accommodate this mapping by better reflecting the names and roles used by the *Room* collective in describing their editorial workflow, we conducted a second translation of the user-interface language. For instance, OJS' administrative hierarchy of Editor, Section Editor, and Peer Review was changed to Acquisitions Editor, Issue Editor, and First Reader, with the sequence of notifications between these three modified (to the extent that this is possible in OJS) to suit the generally non-hierarchical editorial collective model at *Room*.

In January, we looked into adapting the prototype for *sub-TERRAIN* magazine and brought it to editor Brian Kaufman for feedback. At this point, we found that *sub-TERRAIN's* editorial process reflected neither the original OJS workflow nor the one we had adapted for *Room*, but was in fact a third alternative that required further modification and translation. Especially thorny here was the system of notifications that the software would need to produce as a particular submission went through the editorial process, since *sub-TERRAIN's* communications needs differed from those of *Room* and the general journal-publishing model OJS embodies. This suggests that such modifications might be required for every new publication to adopt the software, considerably taxing the available resources supporting this project.

We had assumed that a core editorial workflow existed, to which all magazines could be expected to roughly conform. We learned that, rather, small, cultural magazines could exhibit a vast range of different editorial processes/models, as the operating models of such publications arise organically from individuals' desire to create and publish – which stands in marked contrast to the generalized peer-review model that precedes and governs a vast swath of scholarly communications and makes OJS such a nice fit.

Examining editorial workflows

In the course of gathering feedback and testing the OMMM prototype in the context of particular publications, the real challenge proved to be handling the variations in editorial process from one magazine to another. We had encountered this issue early in the OMMM project, in consultations with *Geist* magazine, but my hope had been that “literary review”-style publications such as *Room* and *sub-TERRAIN* would present a more consistent editorial workflow, not to mention one with more overlap with the scholarly journal workflow embodied in OJS. Our experience with *Room* and *sub-TERRAIN*, however, proved otherwise, especially with respect to differences in collaborative models and how editorial authority is manifest in the details of the

process. How a manuscript is handled and passed from one person to the next varies considerably, reflecting these publications' operational adaptation over the years as well as their fundamental governance models – *sub-TERRAIN*, for example, is relatively more centralized around Kaufman as publisher, whereas *Room* is distributed across a non-hierarchical collective with a rotating issue editor, but both publications embrace collaboration at various stages.

OJS, on the other hand, embodies a workflow model designed for scholarly journals that is hierarchical and non-collaborative, with relatively stable editor and section editor roles, a detailed peer-review process, and a delegation-based process to include peer reviewers, copyeditors, and proofreaders. A typical academic journal relies on the authority of stable editorial and review participants, coupled with discrete delegation of copyediting and production tasks to a periodically changing stable of junior staff. As a result, the enrolment of particular users to particular workflow tasks is fairly rigid and forms a core structural feature of the software.

While we were able to change the *apparent* character of the OJS workflow by streamlining and re-naming various roles and components, what remains is an awkward fit between publications' existing processes and the model embodied in the software. Furthermore, OJS was developed in such a way that its peer review-based editorial workflow is hard-coded in the software; our only options for changing it seemed to be (1) re-naming the roles and components; (2) making limited changes to how the automated e-mail notifications work; or (3) bypassing the review process altogether, effectively making much of the software's functionality irrelevant. Through a combination of strategies involving these three options, we succeeded in creating an adapted version that is functional, but so far it has proved to be far from compelling, despite the apparent enthusiasm of people such as Kaufman and Gugeler to the general idea of this project. Though everyone consulted agrees that this project is a good idea, real success will only come with making a system that people actually want to use.

A long-term solution: Development of a flexible workflow architecture in OJS

In spring 2009, with the structural limitations of OJS' editorial workflow model in mind, I tasked a group of graduate students in SFU's publishing program with designing and documenting a more flexible, inclusive model for how editorial workflow could be handled in the software, with the aim of presenting a specification to the OJS development team so that a future version of the software could escape the limitations described here. This model, developed by the "Flip Systems" project group (Chen, Gaumont, Johnson, Newman, & Simms, 2009), was presented to the OJS development team in April 2009. The model they outlined consists of four core editorial stages:

1. Evaluation: Submission and initial review
2. Negotiation: Conditions of acceptance (revisions, et cetera)
3. Editing: Development of material for publication
4. Output: Pre-production and production

Each of these stages would contain a set of optional substages that a particular publication could choose to add in order to reflect the particular details of their process. For example, the "Evaluation" stage might include a blind peer-review process

(in the manner of the existing OJS), or it might be configured to merely notify an editorial group that a new submission requires attention. Similarly, the “Editing” stage might include a set of discrete sign-off steps to flag substantive editing and copyediting substages as complete, or it might alternatively eschew explicit substeps, letting an editor personally manage an article for publication.

In an important conceptual move, the “Flip” model emphasizes *tasks over roles*. The effect is that particular staffing/delegation decisions are not prescriptive. So, for example, rather than there being an “issue editor” role with a specific sign-off responsibility, the Flip model works the other way around: a specific editorial responsibility may be handled by anyone with sufficient administrative permissions within the system. This approach reflects the workings of magazines like *Room* and *sub-TERRAIN*, where groups of editors often collaborate on decision-making and developmental responsibilities. Notably, this moves away from the discrete delegation approach that OJS takes. The trade-off is the loss of tight control of editorial process that OJS provides; giving this up may not be appropriate for most scholarly journals, but it may be a better fit for many small magazines operated on a collective basis.

That said, the “core plus options” architecture proposed by the Flip Systems group might be sufficient to build a delegation-based peer-review workflow such as that of OJS, and as such, we are hopeful that the OJS developers will seriously consider moving to a more modular, user-configurable workflow model. We are encouraged by the interest they have expressed so far. Realistically, though, this represents a fairly long-term solution; it would not be available to magazine publishers in the near future.

A near-term solution: Adoption of a minimal workflow system

To address the existing needs of BC magazine publishers, particularly with respect to enabling online submission, an interim solution may be found by exploring the potential of general-purpose Web-based content management tools. As a guiding principle, more immediate value may be found in a more minimal system, rather than a full-featured one such as OJS. As a full-featured system, OJS has considerable appeal, given all the features it offers to journal publishers. But the trade-off for the mature set of features and functionality seems to be flexibility. This is apparent in the centrality of its built-in workflow model, as well as in the feature complexity that comes from years of development for its specific audience – in short, OJS may have too much invested in satisfying its core scholarly publishing community to be easily adaptable outside this community. Therefore, as a near-term solution to the problem of how to move small magazine publishers toward an electronic, Web-based editorial flow, we have been investigating other options, with an eye to identifying a tool that satisfies the following four minimal criteria:

1. provides a means of electronic submission
2. offers basic workflow (notification/sign-off) functionality
3. is based on a free/open platform, allowing unencumbered customization and adaptation
4. features a simple and uncluttered interface.

Our review identified three likely candidates for a minimal platform. All three are mature free/open source software projects with healthy development and support communities. The three candidates represent very different approaches to content management.

The first option, OpenDocMan (<http://www.opendocman.com>), is a very minimal file- and attachment-based document management tool. It offers few features beyond simple document management, presenting itself as a kind of Web-based file server with minimal workflow and tracking functionality added. It is not an online publishing tool, but it offers a straightforward editorial review process and centralized, Web-accessible storage and management.

A second option is WordPress (<http://www.wordpress.org>), the popular, full-featured blogging platform. WordPress as a general-purpose content management tool has been successfully adapted to a variety of different tasks beyond blogging. WordPress has considerable popularity owing to its straightforward, well-designed interface and enormous support community. Its basic functionality includes a simple two-stage workflow system, and it is highly customizable via the thousands of third-party plug-ins that are available. The key to its success in our context would be the extent to which it could serve an off-line, print-oriented workflow, as opposed to just a Web publishing orientation.

A third option is a Web-based content management system, Plone (<http://www.plone.org>). Plone is a very robust general-purpose content management platform. It offers many features in common with OJS, but where OJS has been developed with a very particular context in mind, Plone has been developed as an open-ended content management toolkit. Plone can be easily compared with the popular content management tool Drupal, which powers millions of websites around the world. Plone's approach and feature set are very much like Drupal's; the critical difference seems to be that Plone has document-oriented workflow at the very core of its functionality, while Drupal seems more oriented to online communities.

In recent months, we have experimented with a Plone-based prototype of a submissions management system. In this system, we have defined two classes of users: authors and editors. Authors, once registered in the system, are presented with a simple form for uploading a Word document (or equivalent) and adding basic metadata.³

In the prototype, a submission initially lands in a "submission bucket," and the system generates notification e-mails confirming the submission to the author and notifying the editorial group that a new submission has been received. When they log in, editors see a "dashboard" interface that provides information about all submissions logged by the system. Each submission can be flagged by an editor as "draft," "pending review," or "published," with a simple colour-coding system (red/yellow/green – an idea contributed by *sub-TERRAIN*'s Brian Kaufman). When the status of a submission is changed, editors are once again notified by e-mail. Last, a submission can be moved by an editor at any time from the "submission bucket" to a folder marked for a particular publication issue. All changes to a submission over its lifetime are tracked within the system, and Plone allows changes to be rolled back if necessary.

Such is the minimal functionality of our prototype. It provides an organized set of containers for central storage and offers simple communication for all parties. The prototype focuses on announcing the discrete stages of a submission's development: arrival, review, and approval for publication. It does not manage the specific assignment of tasks to individual editors, nor does it require a specific sequence of sign-offs to move a submission through to publication. Rather, the power is left in the hands of the editors – as such, the system assumes the existence of a functional, mutually trusting editorial group. We believe that this is appropriate to how small magazines generally operate. But this “distributed” model of authority may not be right for all publications; scholarly publications may require more discrete task management.

Conclusion

OJS delivers an enormous amount of value to publishers. This is clearly evident in the set of features and functionality that it provides to publishers of academic journals, and its growing ranks of users (over 3000 journals worldwide, at this point) are a testament to its utility. But OJS provides a different kind of value as well: in offering a concrete embodiment of a model of periodical publishing, it serves as a point of discussion and departure for issues in publication management. This notion was recognized early on in OJS' development, when the OJS team noticed that the software served as a way of showing would-be publishers how to actually run a journal – as a kind of curriculum in journal publishing operations, scaffolded by OJS' careful and detailed modelling of the many tasks and responsibilities involved in producing a publication.

For our purposes – serving small magazine publishers – OJS serves a similar role, though not so straightforwardly. Rather than teaching magazine publishers how to do their jobs, OJS provides an analytical reference point for articulating what actually goes on (or what needs to go on) in an editorial process. It provides a way of making what is in many cases implicitly understood about workflow into something explicitly articulated. This is not a matter of mapping poorly understood practices to OJS' well-defined structures, but rather an analytical process of comparison between actual practice and modelled practice. In our case, this has served to help us better understand how processes should be articulated, and it leads directly to the development of new software tools and new ways of thinking about the publication process.

It is worth acknowledging explicitly that the steps we have taken to define editorial workflow more fully and to prototype a new system for magazine publishers could not have been reached without careful engagement and study of the model OJS offers; through successive attempts to map OJS' structures with what we were finding – bit by bit, in some cases – in actual practice among small magazines, we have been able to elaborate an alternative model. In some respects this model draws heavily on OJS; in other respects, the new model is a reaction to OJS' way of doing things. But it is clear that OJS has become something of a paradigm in periodical publishing; whether one follows it directly or deviates from it, the fact remains that it has provided the foundation for the discourse that ensues.

We are closer to our goal of providing BC magazine publishers with online submission and editorial management support, in terms of both understanding the intricacies of the problem and actually providing workable software. That said, the challenge that

remains is to find a comfortable mapping between some actual practices and software models – one that encourages people to use the software on a daily basis.

NOTES

1. The terms “small” and “cultural” have specific – somewhat problematic – histories in Canadian magazine publishing (see Pfeiffer, 2009). By “small, cultural magazines,” we mean independent magazines, usually with an arts or culture focus, and usually with small circulation (<10,000). This is not an official category of publication, though.
2. An issue identified at the very beginning of the OMMM project was the possibility of using electronic submission and editorial management as a pathway for magazines to ultimately explore online publication in addition to or instead of print publication. A tool such as OJS has features to support online publication, so even if a magazine were to adopt it solely to facilitate electronic submission, the possibility of online publishing would be relatively close at hand. Publishing online further problematizes the use of Microsoft Word as a content format, though (see Maxwell, 2007).
3. A welcome feature in Plone is that it handles both a Microsoft Word .doc-based content model as well as Web-native content, and it offers workflow and versioning for both. This provides a smooth pathway out of a Word-dominated submission model.

REFERENCES

- Chen, Annie, Gaumont, Adam, Johnson, Leanne, Newman, Jenna, & Simms, Michelle. (2009, April). *Tech projects 2009: Editorial workflow*. URL: <http://thinkubator.cvsp.sfu.ca/TechProjects2009EditorialWorkflow> [December 16, 2009].
- Maxwell, John W. (2007). Extending OJS into small magazines: The OMMM project. *First Monday*, 12(10). URL: <http://www.uic.edu/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/1962> [December 16, 2009].
- Pfeiffer, Claire. (2009). *The Small Magazines Office of Magazines Canada: 2003–2007*. Unpublished project report, Master of Publishing Program, Simon Fraser University.