Community-led Open Publication Infrastructures for Monographs (COPIM)

Open Sourcing Reuse

Janneke Adema Simon Bowie Rebekka Kiesewetter

Community-led Open Publication Infrastructures for Monographs (COPIM)

Published on: Apr 03, 2023

URL: https://copim.pubpub.org/pub/combinatorial-books-documentation-open-source-tools-platforms-post5

License: Creative Commons Attribution 4.0 International License (CC-BY 4.0)
This is the fifth blogpost in a series documenting the COPIM/OHP Pilot Project *Combinatorial Books: Gathering Flowers*. You can find the previous blogposts [here](#), [here](#), [here](#), and [here](#).

In the context of the *Combinatorial Books: Gathering Flowers* Pilot Project one of our aims has been to use, wherever possible, modular and open source writing, editing, annotating, and publishing software, tools, and platforms. We wanted to show how these can be used in the context of reusing and rewriting existing open access books or collections of books. Instead of creating our own custom solutions we have tried to create (technical) workflows that consist of existing open source applications, to enable other authors and publishers to apply or adapt these to their own writing and publishing workflows.

At this stage of the Pilot Project we want to share some preliminary insights and reflections in combination with a closer description of the tools and platforms we have used. We will do so in textual form and as part of an audio interview with Simon Bowie, who is working as an open source software developer on the COPIM project. Specifically, we want to share our rationale for using open source applications, and reflect upon how these tools can either be integrated into or require adaptations to classical editorial and publishing workflows, timelines, tasks, and relationalities between those involved in publishing a book (for example those between tool and platform providers, publishers, developers, and editors).

### Why Open Source?

Open source tools, platforms, and technologies can, in principle, foster the more collaborative, open, fluid, and equal processes and forms of knowledge creation, evaluation, and distribution ([Fitzpatrick, 2011](#); [Kalir & Garcia, 2019](#); [Montgomery et al., 2018](#); [Risam, 2019](#)) that we want to promote in our experimental book Pilot Projects.

Experimenting with such technologies is even more important in an academic publishing landscape in which many larger presses run or have acquired proprietary editorial and workflow management tools and multi-channel publishing software. One example of this is the workflow management tool AriesSystems. Bought by Elsevier in 2018 it is used by publishers forming part of the RELX Group, and by Taylor & Francis, DeGruyter, and Springer. Aries operates amongst others the Editorial Manager® submission and peer review tracking system, which, similar to other widely used systems such as Clarivate’s ScholarOne, is highly
complex and standardised. However, this doesn’t line up easily with scholarly communication inputs, which, as Cameron Neylon has stated, are “messy and heterogeneous” (Neylon, 2015). As he points out, these kind of large, complex, standardised editorial management systems are considered too complex and are hated in equal measure by both authors and publishers, where the few examples of systems that users actually like (e.g. Open Journal Systems), tend to be purpose built by small publishers. As such, as Neylon states, “these [submission] systems are the biggest blocker to innovation in the industry” (Neylon, 2015) and this is also why we feel proprietary manuscript submission and management systems are generally not very flexible or adaptable to the working contexts of communities partaking in experimental publishing as they do not easily accommodate experimenting with publishing formats and relationalities (Adema & Kiesewetter, 2022). Commercial publishers also often use their own customary “open” licenses on these platforms, which can place restrictions on certain forms of (data) sharing, mining, and reuse, which makes their adaption within, export to, and implementation in different working environments – for example other documents, websites, or publications – difficult (COPIM, 2022).

Seemingly open, collaborative, and modular, yet very much proprietary platforms such as Google Docs are not really a solution either. Google Docs can be easily integrated with other proprietary systems such as the project management tool Slack, the organising software Trello, or Microsoft 365 Excel. In principle users can – without having to write code – assemble sets of interoperable proprietary solutions to create a publishing workflow. Yet there is a lot of secrecy around Google’s operations. As Strickland explains, “only people within Google know how the system supporting Google Docs works” (Strickland, n.d.). This prevents users from shaping and adapting it. In other words, Google Docs “as a piece of software …. is proprietary, cloud based, not installable/deployable, and hardly modular or interoperable” (COPIM, 2022).

However, a differentiation between open source applications has to be made (COPIM, 2022; Maxwell et al., 2019; Chang et al., 2007), as they are supported by a variety of business models. For example, many large corporations have implemented open source solutions to build proprietary software on top of them. Examples include how IBM contributes to the Linux open source ecosystem, while selling software that runs on top of the open source core. Therefore a distinction should be made between “software packages and hosted solutions, and between the commercial, not-for-profit, and other underlying business models (e.g., institutional support) that support these services or platforms” (COPIM, 2022). Within COPIM, we have thus tried to work as much as possible with “those tools that have been made available as self-hostable packages under the premise of open, permissible licensing (e.g., GPL, Apache 2.0). We also highlight the underlying value system and modus operandi chosen by each of the tools so as to make visible the features that may prove conducive for inclusion in a curated selection of such tools, as we seek to do in the COPIM project.” Additionally, within COPIM, we have concentrated on interoperable applications that can be used modularly as “many interesting experiments happen (both in digital scholarship and publishing) when using and combining different tools together in new ways” (COPIM, 2022).
From all of this it becomes clear how researcher and publishing technologist Simon Worthington (2015), who has been working on our Computational Publishing Pilot Project with COPIM colleagues, describes his work with community based open source publishing projects as much as a political issue (as an activity antagonistic to the corporate enclosure of both content and infrastructure) as a technical one (collaboratively building and sharing software and infrastructures that could advance such a politics).

Simon Bowie, what are some of the potential difficulties and costs that publishers and authors have to be aware of when using open source software and adopting these into their established workflows? And what role do you see developers play in experimental book publishing projects?

**Community-forming and -engagement Properties**

Experimental publishing requires a new and different skill-set from authors, editors, and publishers, which may include learning new software, familiarising oneself with different forms of licensing, or even just being open to the idea of a different way of working (Adema & Stone, 2017). Tara McPherson argues for the need to “provide comfort” for authors who may be alienated by technical requirements (in general but also connected to open source tools and software) or overwhelmed by the possibilities that experimentation offers. This is also related to the fact that the use of open source tools occurs across the life cycle of a project and diverse users with different levels of technical skills and digital literacy might have to engage with them. Providing comfort towards the use of open source tools and platform could be established by providing a close briefing on the use of open source solutions (for example, via the user guidelines often provided on these platforms themselves or via one to one explanations) to authors, reviewers, and copyeditors; but it can also be supported by simply trying out different applications to find out which tool or platform fits a specific book and the involved communities best.
The authors of *Ecological Re-writing* annotated the PDF version of *The Chernobyl Herbarium* available on the [Open Humanities Press website](https://openhumanitiespress.org/). As part of the research and writing phase of the *Ecological Re-writing as Disappropriation* project the editor and authors used the hypothes.is annotation tool to annotate the online PDF of the *Chernobyl Herbarium* available on Open Humanities Press’ website. Hypothes.is is open source annotation software that adds a “conversation layer over the entire web that works everywhere, without needing implementation by any underlying site.” As a tool it is seeing wide-spread adoption across the Higher Education sector and it is featured in a variety of open publishing projects to foster the uptake of social annotation practices (see Kalir & Garcia, 2019, and Part 1 of our Interaction & Reuse report). Increased collaboration is supported on a technical level through the provision of a set of tools to help integrate hypothes.is functionality in a variety of other platforms also used for open access book publishing such as WordPress, Omeka, Open Monograph System etcetera. This platform-agnostic nature of hypothes.is makes the tool a versatile candidate for implementation in third-party environments (COPIM, 2020).

The authors of *Ecological Re-writing* tagged their annotations in hypothes.is and from there, once done with annotating and discussing the source text, moved their annotations to a HedgeDoc pad, a collaborative writing environment, to work them into draft chapters. HedgeDoc (formerly CodiMD) is an open source, web-based, self-hosted, collaborative markdown editor, which lets you create real-time collaborative Markdown notes. It can be installed on a self-hosted server or offers a cloud demo version that works in the browser. It is inspired by Hackpad, Etherpad and similar collaborative editors.
As editors of Open Humanities Press’ Combinatorial Books books series, we became increasingly more involved when discussing and choosing, together with the book’s editor, what platform the final book would be published on and how we would further enhance the publication. Engaging with open source tools and platforms and implementing corresponding workflows has required various efforts of (non-linguistic) translation and negotiation from our side towards the various communities (authors, publishers, reviewers, copy-editors, platform and software providers etc.) involved in the creation and publication Ecological Re-writing as Disappropriation (Kiesewetter, 2022). This included discussions about the benefits and drawbacks of (and possible adaptions to be made to) the tools and platforms used; or about how writing, editing, reviewing, and publishing practices and expectations might need to be adapted to these tools and platforms. It also included engaging in ongoing conversations with the developers and designers behind these platforms or drawing on the resources available for their users. These conversations evolved either via emails between Simon Bowie, part of our project team and working as an open source developer on the COPIM project, and the editors and the platform providers, or via the PubPub GitHub page, where Simon Bowie has suggested possible adaptations and further developments of their editing and publishing environment based on issues and requirements that emerged in the process of publishing Ecological Re-writing.

One of the benefits of open source applications is that many of the software communities, platforms, and digital tool developers are open to collaboration and to forming networks. As Worthington argues with respect to the now inactive Hybrid Publishing Group, for example: “It is important to emphasize is that the HPC is not a fixed and finalised group and we are only at the beginning of forming the network. We want to invite more
people to join. The plan is for long term collaboration with a network of stakeholders to support Open Source infrastructures for transmedia, multi-format, scholarly publishing” (2015).

As editors we tested several writing, annotation, and publication platforms, in collaboration with *Ecological Re-writing*’s editor Gabriela Méndez Cota. This included looking at the open source community-led platforms **Manifold** and **PubPub** as places to publish the book on. PubPub is an open source, community-led, end-to-end publishing platform for knowledge communities. It supports annotation on the technical level of the tool and its affordances, but also on the level of fostering social interaction and community-building on and with PubPub. PubPub has good import functions (Word, XML, LaTeX, and Markdown) and export functions (PDF, Word, Markdown, LaTeX, JATS XML, and more) and allows embedding of rich multimedia. PubPub is already used by the scholarly community to run journals on (e.g. Reviews in DH, San Antonio Review), and to support book publications (e.g. media.studies press, MIT Press, Goldsmiths Press). Manifold is a publishing platform that is more directly focused on the publication of books, converting existing content into responsive online texts that can be annotated and augmented with rich media. It predominantly focused on supporting the research process, or forms of processual publishing, enabling raw data, drafts, notes etc. to be incorporated into or connected to the book publication. Presses such as the University of Minnesota Press and Athabasca University Press are already using Manifold to publish their books on.

Together with the book’s editor we wanted to make sure that the publication platform that we would chose would capture the nuances of the relational, contextual, and open-ended nature of the authors’ work, and that it could accommodate the skill-sets of the different communities involved. We finally settled on PubPub as it accommodates versioning, community writing and collaboration, and has its own build-in annotation system, which we could use to make links back to the various previous versions of the book and its chapters as published on other platforms (in pads, annotation software etc.). All the various English and Spanish chapters that together make up *Ecological Re-writing* were subsequently transferred from the writing environment (the HegeDoc pads) via Word (which was used by the book’s editor to proofread and review the first draft of the book) to separate “Pubs” on PubPub (a unit of content on PubPub, e.g. an article or a chapter).
Although PubPub allows annotation (and hence social practices such as open peer review which is a practice we experimented with during Ecological Re-writing), it does not offer the option for the creation of private peer review groups that only can be accessed and seen by communities invited to these groups (e.g. authors, editors, and reviewers). For this reason, we chose to use the hypothes.is annotation tool again for our (semi-)open peer review process (we will reflect more on our experiences with open peer review in a next blog post), as it comes with more flexible, anonymous and non-anonymous, private and public group settings than PubPub’s built-in annotation function. We are using PubPub’s annotation function too, to make backlinks to previous versions of the book in development. For example, we wanted to make sure there are various connections back to the Chernobyl Herbarium as source text, as well as to texts under revision and in development in different annotation and writing environments, for those wanting to retrace these processes and the context of the responses (which we are enabling by putting in bi-directional links using the annotation function on PubPub between the source-text PDF and the chapter-length responses to the source-text published on PubPub).

Such a process of experimenting with and testing different solutions also involves an awareness of the drawbacks and benefits of the platforms used and how editing and reviewing expectations and practices might need to be adapted according to the platform used. One of the drawbacks to using PubPub, from a writing and (copy-editing) perspective, is that it is very much set up as a publication platform in first instance and not as a (copy-)editing platform. PubPub does not offer editors and authors the possibility to make colour-coded track changes or does not allow one to easily revert any edits made (it is missing an ‘undo’ button). Hence why in the end we decide to use Word to do our copy-editing, exporting the draft chapters into Word, and importing them...
again once the copy-editing is done. PubPub also offers less structural flexibility than we originally hoped for, for example regarding the implementation of non-linear formats. We had to dismiss our initial idea to have a randomised, shape-shifting chapter structure and display of the “Pubs” for Ecological Re-writing, which could have been used to underline the open-endedness and fluidity of Ecological Re-writing.

These drawbacks notwithstanding, we believe that ongoing experimentation with these kinds of open source platforms and tools and the mutual sharing of experiences and insights of using them, might further help publishers, editors, and authors integrate these technologies into more established, or conventional, publishing systems and workflows or to further adapt these. Additionally, working with and where needed adapting existing open source solutions (in collaboration with the providers of the tools and platforms used), as we have done in this Pilot Project, can help to decentre the costly, tailor-made writing, editing, and publication environments, based on closed source code, that dominate academic publishing.

Through sharing our experiences here we hope to help establishing the kinds of relationships that will further enable and bring about the more equitable and community-led not-for-profit ecosystem to support academic book publishing that COPIM wants. Additionally, our Books Contain Multitudes Report (2022) provides an overview of further open source tools and technologies currently available to support publishing experiments – as well as a way to potentially connect to the open source communities that maintain these tools and infrastructures, which might lead to further collaborations. This spring will also see the beta-release of our Experimental Publishing Compendium, which will conjoin diverse open source tools, examples of experimental books, and experimental publishing practices.

**References**


