

Sustaining OER Efforts: Lessons From My First 17,000 Pages

Jonathan A. Poritz

jonathan@poritz.net
poritz.net/jonathan



28-30 October 2025, online
Open Ed 2025



This presentation, except where otherwise indicated, is by [Jonathan Poritz](#) and is distributed under the [Creative Commons Attribution-ShareAlike 4.0 International](#) license. This version: 14 Dec 2025 00:17CET
The slides, also in editable form, are available at poritz.net/jonathan/share/sustOER17k/ .

Thank you for coming to my talk, in a manner of speaking!

I apologize for speaking quickly! Note that these slides are available at the URL shown in the lower-left corner of each slide. Each slide also shows the icons for the CC license under which I am releasing the slide deck, in case anyone does a screen grab and wants to know what their reuse rights are for this work.

Please note that this version of the slides has been updated since the version used for the OER 25 video, only to update the cover images and links for some of the OER shown on page 4.

Positionality acknowledgement



¹"Silence = Genocide with Palestinian flag by Daynarama is released under [CC0 1.0](#).

The place where I live now does not have a tradition of land acknowledgements. I could say that I worked on some of the things I will talk about when I lived in the traditional, unceded lands of the Ute and Arapaho peoples, and that others who lived on that land included the Cheyenne, Comanche, and Apache. For more information, I recommend consulting the [Colorado Tribal Acknowledgement List](#).

But actually I think a different acknowledgement is probably better, coming from me in particular, at this moment in history: You see, while I am an entirely non-religious person, I am of entirely Jewish ancestry.

So I want to add my voice to others who are identified as Jewish by society around us, no matter our own religious feelings or lack thereof, in saying that what Israel is doing in Palestine is genocide and must stop!

The F-bomb



Sometimes I think the only thing we should be talking about is how to prevent the further rise of **fascism**.

But, arguably, getting down into the weeds of how we can do *education* better, teach more *critical thinking*, use more approaches from *open education* is exactly how some of the people at this conference can most powerfully use their knowledge and skills to contribute to ending fascism.

Keep in mind, though, that we may have to reevaluate the basic priorities of the work we do, because: fascism.

Let's talk about sustainability of OER publication efforts. First:

What are we trying to sustain?

²This image, by Jonathan Poritz, incorporates the image [Fasces](#) of Višeslav with license [CC BY-SA 4.0](#)) from Wikimedia Commons, as well as an image with license [CC BY 4.0](#) created by Poritz adapting the [Global OER Logo](#) of [Jonathas Mello](#), which also has license [CC BY 4.0](#).

I believe that one of the very important roles of scholars is to speak the truth, which is why I've been using the "*F-bomb*" a lot lately, meaning I use the word **fascism** a lot.

There is an argument that we in open education should at this moment be concentrating on what we can do under the menace of fascist control.

There is also an argument that what we can do best each in our own fight against rising fascism is to be better educators, teach our students true history to know what humans have really done, true science to know what the world really is like, and better critical thinking skills so they will be harder targets for fascist indoctrination. Arguably, many of us at this conference are best at using well the more effective and liberatory practices of open education, and at supporting the creation of excellent and affordable (free!) resources which will enable even today's struggling students to maximize their chances to get those benefits of education I just mentioned.

So I'm going to narrow my focus, just for the next nine minutes, to those practical tasks of open education.

In particular, before it all went up in flames, as a community we used to worry a lot about how to make open education more sustainable. Which makes me ask this question ... "What are we trying to sustain?"

Some very successful OER publication projects



CC BY 4.0 by the Washington State Board of Community and Technical Colleges



CC BY 4.0 by Open Oregon Educational Resources
(covers, at least...?)



PB PRESSBOOKS



CC BY-NC-SA 4.0 by Lisette Helena Assia Espinoza and Deirdre Budzyna/Doris Buckley



Rebus COMMUNITY



One answer would be “things like this.”

These are just some of the (book-shaped) OER on which I have worked.

They are wonderful books which will enormous benefit to the students in WA, OR, and MA ... and, through the miracle of openness, potentially far afield from the west coast of the US!

Most of them are many hundreds of pages, which, including the many revisions books tend to go through, is where I got the 17,000 pages of my title.

I chose to display these ones in particular because *they look like textbooks*, in fact rather beautiful textbooks, thanks to the Pressbooks platform.

It was my great pleasure to work with the Open Washington and Open Oregon groups in the last few years, helping to move the books to Pressbooks for their students.

(For the other two books, I had the pleasure of working with Rebus as a facilitator of their “Textbook Success Program,” as it was called at that time; but I was not deeply involved in getting them onto PB.)

“Move books to Pressbooks” – where were they before? and why move?

A fairly common workflow in our community seems to be:

Create/adapt OER in Google Docs (hereafter **GD**)

and

share them with students in Pressbooks (hereafter **PB**)

We might call this ***The GD~>PB Flow***.

Pros of GD	Pros of PB
● familiarity: e.g., MSWord-like ● UI: WYSIWYG ● supports collaboration well ● supports a11y, after user effort ● Google well-integrated in our lives	● familiarity: <i>looks</i> like an e-book ● reading is clean, simple, responsive ● allows embedded videos, H5P, <i>etc.</i> ● supports a11y, after user effort ● discoverability: the PB <i>Directory</i>

On this slide “UI”=“User Interface” and
“WYSIWYG”=“What You See Is What You Get.”

Before, they were in GD. I will call the transfer GD- $\rightarrow$ PB the “GD- $\rightarrow$ PB flow.” Why is this process so common in open ed?

Well, this slide shows some of the features of those two platforms that open ed folks appreciated.

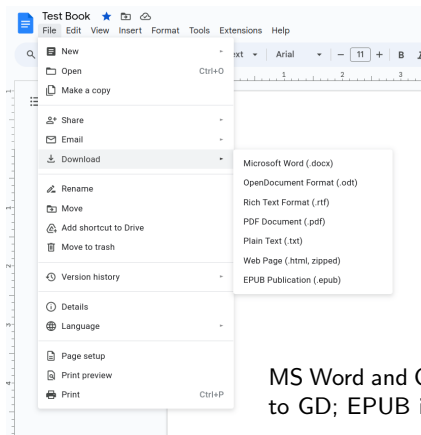
But moving from one to the other is not easy.

While I could do an entire presentation about *The GD $\rightarrow$ PB Flow* – I have! – my aim instead is to try to learn some big lessons from sustainability of OER projects from this experience.

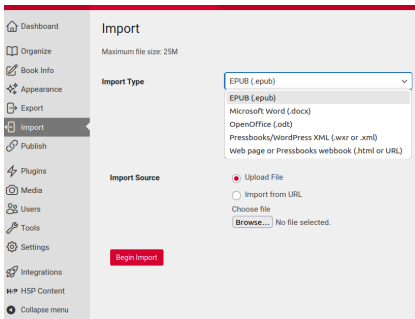
Let me first give you a two minute tour of the specific issues that made this work challenging.

Come on, surely this must be nearly automatic

GD export (“download”):



PB import:



MS Word and OpenOffice are quite similar, also both similar to GD; EPUB is a bit different, but I rejected it quickly.

First idea: export to Word from GD and just import that Word document to PB.

If two such famous platforms provide this pathway, surely it “just works”!

Unfortunately GD→Word→PB fails in several ways

Experimentation with this process found that the alt texts of images were being lost along the way – that’s a deal-breaker!!

Additionally, it was very unclear how to get elicit from the Word version of a GD book the features, like embedded videos, H5P, *etc.*, to appear in PB.

A way to allow that might be to add text decorations to the GD which could be translated to features in PB, such as wrapping a video URL in indications like “[embed] ... [/embed]” when it is desired that the video at that URL be embedded in the PB, something like “[GL] ... [/GL]” to mark text that will be associated with a glossary term in the PB, *etc.*

Adding enough of these decorations makes the GD start to look like it was written in a **markup language**, such as HTML or T_EX, which is something many OER support teams are reluctant to impose on their authors: “non-programming humans don’t like markup.”

An approach based on a kind of markup language **would** allow for markup which was *semantic*, which is more useful and powerful than markup which is purely *stylistic*, or WYSIWYG which confusingly mixes the two.

But it doesn't "just work," in particular the alt texts for images in the GD get lost. I think the loss is happening in the GD export. Just part of Google's enshittification, I guess (to use Cory Doctorow's recently coined word).

Additionally, there are features on both platforms that we want to match, like embedded videos and PB glossary terms ... the programmer's way to do this would be to add some decorations to the GD files which make a kind of lightweight "markup language."

Apparently, markup languages are very scary to non-programming humans, Open Oregon's Pheobe Daurio and I used to say, but writing and mathematics and street signs and so much around us uses markup languages, so I guess I feel like we could try to get over that fear.

A better approach: Get the OER out of GD as HTML

Or we could go all-in on the idea of markup language, and export from GD using the option “Web Page (.html, zipped)”: HTML is manifestly a ML; the zip file contains all the GD's images. Google makes quite ugly HTML, though. *E.g.:*

```
<html><head><meta content="text/html; charset=UTF-8" http-equiv="content-type"><style type="text/css">@import url(https://themes.googleusercontent.com/fonts/css?kit=0PeqXG-QxW3ZD8BtmPikfA);.lst-kix_qpcrjg12wdx5-3>li:before{content:"" counter(lst-ctn-kix_qpcrjg12wdx5-3,decimal) ". "}.lst-kix_5iabqo6dki7m-5>li:before{content:"\0025a0 "}.lst-kix_qv847ouxndzf-2{list-style-type:none}ul.lst-kix_qv847ouxndzf-1{list-style-type:none}ul.lst-kix_qv847ouxndzf-0{list-style-type:none}.lst-kix_6orila2j8o67-2>li:before{content:"" counter(lst-ctn-kix_6orila2j8o67-2,lower-roman) ". "}.lst-kix_7qiebeeinotp-8>li:before{content:"\0025a0 "}.lst-kix_zh0djvhcuw11-5>li:before{content:"\0025a0 "}.lst-kix_qpcrjg12wdx5-7>li:before{content:"" counter(lst-ctn-kix_qpcrjg12wdx5-7,lower-latin) ". "}.lst-kix_ymhtletofw0-0>li{counter-increment:lst-ctn-kix_ymhtletofw0-0}.lst-kix_h0qa59x4hax3-1>li:before{content:"\0025cb "}.lst-kix_e9xxj2arum8r-7>li{counter-increment:lst-ctn-kix_e9xxj2arum8r-7}.lst-kix_88guulohcsd4-1>li:before{content:"\0025cb "}.ol.lst-kix_2g5fjno92eyf-0.start{counter-reset:lst-ctn-kix_2g5fjno92eyf-0}.lst-kix_jfrs6xlc6t05-2>li:before{content:"\0025a0 "}.lst-kix_6orila2j8o67-6>li:before{content:"" counter(lst-ctn-kix_6orila2j8o67-6,decimal) ". "}.lst-kix_6omuljmxba6-1>li:before{content:"\0025cb "}.lst-kix_78u85j5w8j7h-0>li:before{content:"\0025cf "}.lst-kix_78u85j5w8j7h-8>li:before{content:"\0025cf "}.lst-kix_h0qa59x4hax3-5>li:before{content:"\0025a0 "}.lst-kix_wtp5ial5q6ui-0>li{counter-increment:lst-ctn-kix_wtp5ial5q6ui-0}.lst-kix_zh0djvhcuw11-1>li:before{content:"\0025cb "}.ul.lst-kix_qv847ouxndzf-6{list-style-type:none}ul.lst-kix_qv847ouxndzf-5{list-style-type:none}ul.lst-kix_qv847ouxndzf-4{list-style-type:none}.lst-kix_5iabqo6dki7m-1>li:before{content:"\0025cb "}.ul.lst-kix_qv847ouxndzf-3{list-style-type:none}.lst-kix_jfrs6xlc6t05-6>li:before{content:"\0025cf "}.lst-kix_bxpnnqgj9tbtn-3>li:before{content:"\0025cf "}.lst-kix_bxpnnqgj9tbtn-7>li:before{content:"\0025cb "}.ul.lst-kix_qv847ouxndzf-8{list-style-type:none}ul.lst-kix_qv847ouxndzf-7{list-style-type:none}.lst-kix_78u85j5w8j7h-4>li:before{content:"\0025cf "}.lst-kix_zg53fmwob2lv-6>li:before{content:"" counter(lst-ctn-kix_zg53fmwob2lv-6,decimal) ". "}.ol.lst-kix_ldxyxwrakbft-1.start{counter-reset:lst-ctn-kix_ldxyxwrakbft-1}.lst-kix_9ulwjb7lfd5-1>li:before{content:"" counter(lst-ctn-kix_9ulwjb7lfd5-1,lower-latin) ". "}.lst-kix_zg53fmwob2lv-2>li:before{content:"" counter(lst-ctn-kix_zg53fmwob2lv-2,lower-
```

That particular book goes on for 1.6MB of HTML ... all on one line! And filled with HTML features like classes and anchors whose names are meaningless alphanumeric strings... argh!

Ugly ... but “it’s all text”

HTML files are at least just text files, so can be processed in a quite straightforward way by code.

This was one of the great insights of the Unix operating system, which was the ancestor of Linux and macOS: everything should be thought of as (text) files, so that one can just sequences of tools to build complex behavior and data processing.

Following the model, the ugly HTML output by GD can be cleaned up, with new, bespoke tools and existing ones like the (FLOSS!) tool [tidy](#) from the [World Wide Web Consortium](#).

The $\rightsquigarrow$ **PB** part of the **Flow** means that we have to translated any markup we may have created for our books as they live in GD into the HTML that PB can handle, Again, use code. And stylistic handling on the PB end can be done quite a bit by custom CSS in PB.

Once these HTML processing matters are handled, we still have to deal with the unfortunate WYSIWYG interface to PB. Fortunately, there are tools which allow one to automate access to web interfaces, such as [Selenium](#), so that we can do the pointing and clicking PB is expecting to upload the HTML we have created by textually processing the GD output.

But we can process all of that ugly HTML, plus our new markup language, using code, since it's all just text. That's actually easier than processing a GD file, which can only be reached by a WYSIWYG interface!

When we're done processing, we can upload the nice, PB-adapted HTML to PB's graphical user interface using a tool such as the one I like, called "Selenium", often used to stress test web sites and other related tasks.

Now we can deal with images ... and many other things

The HTML that is exported from GD at least has all of the GD alt texts in the generated HTML that looks like

```

```

This means that our code-manipulated HTML can easily put those alt texts in the right place to make useful ally enhancement of OERs in PB.

We can also do in code lots of other things, like:

- Use HTML that puts different kinds of textboxes around author material of different (preselected) types.
- Auto-generate attribution statements
- Accept an author-created spreadsheet of glossary terms and their definitions and then turn on the PB glossary feature for those terms in particular contexts.
- Implement a fairly sophisticated syntax for cross-reference links.
- Generate QR codes that appear in printed exports in place of embedded videos, so that students can, when they have internet access, scan the code and view the video.
- Check and, when necessary, correctly renumber Figures and sections.
- *etc.*

All of the bullets on this page are nice things which we can do now that we are processing the text of OER in HTML/text form. A few comments:

Yes, PB will generate attribution statements, but it handles only fairly limited situations and it is fairly easy to do much better.

The QR code approach is something kind of cool, I've not seen a comparable solution for helping folks who mostly consume an OER offline which nevertheless contains online-only features.

Humans are very bad at counting, so figure and section numbers (and other, enumerated artifacts) are very often wrong. Computers are good at counting, so should do this boring work!

And so we can make this approach, and I am grateful to Open Oregon and Open Washington for paying me to do this work. I do think it was a lot of money to be spent for the benefit.

But at least with the benefit of this experience, and comparing to my other OER creation/adaptation experiences, there may be lessons that can be learned.

Pain points in this multiplatform approach to OER

First of all, for these particular platforms:

Cons of GD	Cons of PB
• easy to mess up a1ly • extreme surveillance capitalism • WYSIWYG: no automation possible • terrible (non-existent?) math • interoperates poorly	• easy to mess up a1ly • heavyweight solution to a lightweight problem • “text” editor and custom CSS, but limited flexibility • MathJax, not yet fully implemented • interoperates poorly

More generally:

- Humans aren't good at counting.
- Non-programming humans don't like markup (really?).
- Maybe the obsession with *OER as objects* has lead to a misplaced desire to emulate commercial textbook publication processes.
- Maybe non-programming humans can learn to use tools which do interoperate well and which clearly distinguish between *semantics* and *syntax* [style].

Some pain points of this approach are mentioned in this slide. I'd like to point out in particular that bullet near the bottom that I think an obsession with OER as objects has lead to a misplaced desire to emulate commercial textbook publication processes, since they produce such beautiful results.

But this discussion all fails under rising fascism where we will have no funds or support folks' salary monies, all the while withe much greater student need!

Conclusion, pre-lapsarian version

Platforms for adopting/adapting/creating OER must provide good support for:

- collaboration
- all features
- responsive display
- discoverability
- automation
- math and interactive features such as video, H5P
- **interoperability**

and must avoid:

- surveillance capitalism
- platform lock-in
- heavyweight solutions when the problem is lightweight

which probably involves using

- plain text files, wherever possible
- markup (or -down) to describe semantics and advanced features

This slide represents the scenario where there is some money around OER and a desire to run centralized OER services ... but nevertheless we need to understand how to choose platforms and processes.

I think this era is gone for good, as I said, but if there is any remnant of it, please come back and look at this slide and think about the suggestions.

For example, brave little California still fighting the good fight has funded OER in the community college system. They just made a choice about a major platform where their OER with preferentially live. I sure hope they thought about the issues on this slide!

Conclusion... what can we do after the fall?

In my opinion, centralized OER publishing projects with reasonable-sized teams of support staff doing book design, marketing, grants, formal proofreading and peer review, setting up and enforcing a project-wide “house style” are probably nearly extinct, and not to return.

This may be a good thing.

When the center cannot hold, the periphery has to do the work. The culture of individual instructors has to change to understand and then live in an open educational world.

- Practitioners can learn about open licenses and open practices: citation is drilled into the head of every scholar, why not attribution, and some of the legal technology around it?
- Practitioners can insist, loudly³ or quietly⁴, on FLOSS software, open access scholarship, the exclusive use of OER, *etc.*
- In the end, we can rely upon ***the beauty of ugly OER***, in that an ugly OER which is not of a well-designed house style and which may have typos and may reside on a very informal platform – Google Docs or a plain web page in a domain of one’s own, for example – in the hands of students and created by a human with deep pedagogical insight for their subject ... is actually the most beautiful thing.

³ organize! support your union!

⁴ use the *weapons of the weak*; see James C. Scott

The era of well-funded, large OER publishing projects is probably over. That could be a good thing – I've certainly argued here that an obsession with replicating the structures of commercial publishing, while perhaps a good first step for our movement, is not one that has a future. This is likely true if nothing else because of fascism, but it is also probably a good thing.

I used the phrase ***the beauty of ugly OER*** at the first Open Ed I attended, in 2019, and I think that idea is even more salient today, that what matter is the pedagogy of an OER, not the house style or fancy platform

Examples of this idea include the huge website of [CSU Pueblo World Languages](#), filled with amazing OER in a variety of informal formats; messy collaborative textbooks perhaps built in MediaWiki.

A last comment: I strongly disagree that the way to build pretty OER without our former levels of funding is to hand their creation over to genAI. But my analysis of why that is a terrible solution to our community's problem of sustainability will have to appear in another forum.

Contact Info

Email: jonathan@poritz.net

Fediverse: [@poritzj@mastodon.social](https://mstdn.social/@poritzj)

These slides:

poritz.net/j/share/sustOER17k.pdf

and all files for remixing:

poritz.net/j/share/sustOER17k/

Or else scan 

to go to my homepage, where you can
follow a link to my shared materials page.



Please feel free to contact me in the ways shown on this slide – e.g., I'm happy to talk about how AI is not a solution to the problems that ail us, but rather a deep worsening of those problems!