

I live now in a place where there is not a tradition of giving land acknowledgements, but I have that habit from many years before I moved here, so I want to say:

I am grateful the chance I had to live and learn, and build some part of the foundation on which was built the work I will describe today, on the unceded territory of the Ute Peoples. The earliest documented people in that area also include the Apache, Arapaho, Comanche, and Cheyenne. An extended list of tribes with a legacy of occupation there can be found on the [Colorado Tribal Acknowledgement List at Colorado College](#) – you can find the URL by searching for that phrase.

BTW, “FLOSS” in the subtitle stands for “free/libre/open-source software.”

I am very much looking forward to this session because it seems like my life in recent weeks or even months has been a no-holds-barred attack on technology, and I feel like I’m in danger of becoming a crotchety old man shouting at the youngsters to get off my lawn with your damn gadgets – a Luddite. I even was on a panel yesterday at Lucca Comics and Games, the third largest comic-con on the planet, arguing with a member of the European Parliament about how AI doesn’t actually do all the amazing things it is said to do, and instead is begin used today to “grease the slide into dystopia,” [to use a phrase I started shouting at the Creative Commons Summit last month].

But: actually, a new book – *Blood in the Machine*, by Brian Merchant – argues that the Luddites were not against technology and in fact were enthusiastic about some technology, they just didn’t like technologies which took autonomy and agency away from people and were used to concentrate power and wealth among the few ... so maybe I am a Luddite!

Luddite or not, today’s talk is about a bunch of software I wrote to help humans do a particular task in a particular situation ... because I love technology that empowers humans. As a paper bag at Lucca C&G yesterday said: “Proud to be a Nerd.”

A friend of mine at my former institution wrote an OER structured to begin every chapter with a case study: she thought a concrete example was the best way to introduce a new concept, every single time .., in every textbook on every subject.

While I don't know that I agree her approach is *always* the best, I'm going to follow it today, talking about a case study of the GD \rightsquigarrow PB workflow.

This outline is definitely not the most original, but it just seemed to be the way it was coming together, so I decided not to fight it too hard...

Really amazing! I've had the privilege of working with many OER projects and teams – including

- some through the OEN,

- many through the Colorado OER Council,

- a few through the Rebus Foundation,

- a handful through OE4BW,

- a great many (309 students total!) through the CC Cert course,

- and several through other random situations

– and OO is one of the most amazing organizations, with the best people and the most amazing track records, with whom I've had the chance to work!

That was our by then empty nest in Colorado Springs, CO, USA. Not a bad place ... a lot of work to get a house built in 1896 ready to sell, though!

[The daughter of our landlord in Italy asked if houses in the US are “so new” because of all the tornadoes ... I think she had recently seen a documentary about Tornado Alley in the US, or else had watched *The Wizard of Oz*.]

Now we live in a house that could have been designed by M.C. Escher: it has stairways that lead up to places lower than where you started, I am convinced. And in the winter, when the leaves fall off of the chestnut trees and the air is clear, we can see Les Alpes Maritime of the coast of France – we know our selves to be **very lucky**.

And in case any of you are thinking of burning your lives to the ground and moving to Italy, I can tell you that our cost of living is much, much lower here than it was in Colorado ... even including the cost of the medication I take regularly, which is 43¢ a month.

I apologize that I am not perfectly clear on all of the details of the back-and-forth of this process, and who would be responsible for what, when.

For example, I know that Chemeketa Press “the academic publishing arm of Chemeketa Community College in Salem, Oregon” is playing an important role, and I think there was an separate book designer who made a wonderful plan for how the final version of the book would look.

But from my point of view, as Open Education Instructional Technologist, I have tried to narrow my focus to the simplified version on the next slide.

"Quickly" and "painlessly" because this was, as we can see from the previous slide, going to happen many times, for the dozen or so OO OER.

"Accurately," in that we needed to preserve essentially all of the information put into GD versions, and get it into appropriate places and formats for the PB versions as desired by OO.

This is all aspirational – humans are not good at being accurate, methodical, and consistent.

For example, it's hard to get authors to write alt texts, because (it seems) they don't like to slow down their writing to do that – but alt texts must be written by the authors, because they describe the pedagogical purpose of each image in the text, it is essentially impossible for outsiders to create them

It's also easy to use an incorrect heading level with a [sub][sub]section title, particularly because these are chosen by a pull-down menu in GD on which an author might make a mistake.

Likewise, it's easy to forget to renumber figures, or to do so incorrectly, when moving bits of text around in a book, as authors often do.

I have a lot of sympathy for the authors who feel uncomfortably constrained by house style: one of the reasons I wrote the OER I've written – three that anyone would call “textbooks” and two to four that are quite different beasts – is a kind of obstinacy about wanting my damn class to go my damn way, which, once you let it overwhelm you (and you need *some* strong emotion to drive you to do all the work of making your own course materials ... it's certainly not the pay!), is hard to suppress just to go along with the house style someone else wrote!

Side note from a non-US perspective: the disease of textbook cost increases is **not global!** This is hard for Americans to get their heads around, but it is true: textbooks (and education itself, and healthcare, and good food, *etc.*) are not insanely expensive in some (maybe most) countries. Therefore, the “come for the cost, stay for the quality” slogan which I was so used to quoting when spreading the word about open education in the US is a **non-starter** in much of the world.

Instead, in a number of places on the planet, one has to lead with academic freedom and encouragement of the kind of educator individualism that makes a rigid house style particularly unappealing.

I don't know that I feel the need to eschew long scrolling pages as strongly as the OO folks – my OER often have long (even book-length!) scrolls, and, arguably, traditional books have pretty long scrolls... but of course it was OO's decision!

Long image descriptions were turned into linked, page-level footnotes with a return link, following a strategy described by BCcampus.

Glossaries are a surprisingly complex issue, which I'd be happy to talk about if there is interest.

Perhaps I was a coward not to look deeper into EPUBs – they just seem to me to be yet another detailed technology for which there are not great FLOSS tools to manipulate directly.

I think the loss of alt texts was happening in the GD export. This goes against what we tend to think would happen: just as it is hard not to assume that a more expensive textbook is “better” than a free one, it is hard to believe that the products of one of the largest and richest companies on the planet (market cap as of the time of this writing of around **\$1.6trillion!!!**) would lose important metadata that is vital for a11y during an export process.

It is a mistake to think of Google (or any of those plutocratic top four or five companies) as obviously not making mistakes. How could they, when they can afford the top talent on the planet? (Actually, they can afford the top talent *that is for sale* on the planet... maybe not everyone is for sale.) *E.g.*, if your OER work involves using YouTube videos, it might interest you to know that YT's version of Creative Commons licenses is **absolutely terrible**: information pages about CC licenses contains straight up false information; the licensing statement it uses is not very well constructed; and the link from that licensing statement makes it clear that YT's CC license is only [CC BY 3.0](#). But one of the main improvements when CC went from v3.0 to 4.0 is a feature which pretty much eliminates copyright trolling ... so it is vital that we use v4.0 licenses!


Note that the desire expressed in this slide to have computers “do the boring part” is something that is very hard to do through pointing-and-clicking interfaces ... or at least it seems to be difficult. In any case, this gives me an excuse to do one of my favorite rants!

I did say “for certain tasks.”

A police officer directing traffic usually just points to where cars should go, and that is probably more effective than having each one stop, roll down the window, and describe where to go. (Although stopping would probably be done with a gesture, and we could discuss the semiotics of the officer’s uniform....)

Similarly, for years when I shut down my computer, I would go to a “terminal window” – the main place people who want to use their computers for real work communicate with those computers – and type

```
shutdown -h now
```

That works fine, but I do it often enough, and I have enough screen real estate, that I put a button  up near the top of the screen (near other single-purpose buttons) which does that same command automatically, when it is clicked with the mouse.

There are plenty of other tasks that are easier to do with a GUI than with a CLI ... but the project of GUIfying everything is about taking power away from the users of computers. It’s as if reading and writing were invented, and only governments, large corporations, and a few crazy anarchists learned how to do it, while everyone else was told it is too complicated for the masses, and you need to go to a specially trained person to read or write for you.

I mostly program in Python these days, it's a really nice language. All FLOSS, of course, as are packages like Selenium and many others I use.

Having heard my rant on CLIs vs GUIs, you will not be surprised that merely automating the GUI interaction did not appeal to me. I do use Selenium later in the workflow, but in a very limited way.

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!

Now we're down to 969KB, and it's much more readable, e.g., being split over 3,269 lines.

But still there are those

```
id="t.3"
```

and

```
ol start="1"
```

and

```
id="h.aoycyemv93xp"
```

which are a bit unpleasant to look at. The rest is fine HTML.

Phoebe Daurio, the OO person with whom I worked most directly in the last year (but who, unfortunately for OO!, left for another position elsewhere a couple of months ago), and I used to joke about how “non-programming humans” don’t seem to like markup languages, and we shouldn’t force the OO authors to do anything that looked at all ML-like.

Note that there is markup language called **Markdown**, which I think was originally envisioned as a limited subset of HTML, that would be easier and friendlier for non-programming humans to use. But actually, I believe, it has gradually grown to the point that it is pretty much as powerful as HTML, while still remaining easy and friendly. There are publication systems which use markdown, and I think that is a very good way that OER teams might want to use in the future – it’s friendly, but at the same time gives a huge amount of control.

Actually, in the end we did, and you might have noticed the `[box]` and `[/box]` in the tidied html book I showed before: that is essentially a little bit of ML that we’re asking OO authors to use when they want the text those marks enclose to be in a textbook when moved to PB.

“WYSIWYG” here stands for “What You See Is What You Get.”

Among programming humans, there is a distinction between a “word processor,” which WYSIWYGs over the ML, and a “text editor” (or just “editor”), which simply works with the text.

My code makes a .docx file which simply contains all of the book's images, and one has to upload it to PB by hand: PB automatically creates media library entries for all images in .docx files it uploads.

In fact, if that .docx file is larger than 25MB, first it must be broken into (maybe two or three) smaller pieces, and individually uploaded.

I then download the HTML version of the PB page it made from that .docx upload, which allows my code on my laptop to make a logical connection between the images as they appear in `book.html` and the references PB has made to its media library.

It's relatively little work – a dozen or so clicks per GD→PB conversion – so I haven't bother to automate it with Selenium. But that could be done if it were desired.

Google redirects are quite clever: you can't see them when you hover over a link in GD, since they've hacked the way browsers show their links. Nevertheless, the redirects are there – because, of course, Google got to be a \$1.6trillion company by following the approach of surveillance capitalism.

Figure numbering does require a little of ML because there has to be some way to refer to the figure in the text, even if the number is not known by the author at the time of writing.

External link checking can be complicated by the fact that sites don't like to be visited many times in quick succession from the same IP address – that looks like a DOS attack.

Showing the color-changing HTML is useful so that we can see if the author is using color to convey meaning, a bad a11y practice!

I have used the outliner so many times to figure out when the book had some structural problem, it turns out to be fabulously useful!

Note the manifest allows one to specify a custom CSS file, Front and Back Matter sections, and the GD Parts and sections.

Image files are already on PB by this point, as explained earlier, and the local HTML files to be uploaded have correct URLs to their locations in the PB site

I'm showing you what I would do on my laptop running Linux (of course). Here %
command means run the command in a terminal window (yay, CLI!)

My celebratory beverage is espresso.

Note there is also a program 00download which enables one to download a snapshot of
all the PB HTML, for copying, modifying, etc., working directly with the HTML. Slide 21/

Traditionally, one picks a fun name for a FLOSS project. Since this software is all about going from **Google Docs** to (2) **Pressbooks**, I thought of **GoDot2P**. There was then this nice openly licensed image I could modify and use as cover art!

And also when my home network is slow and it is taking a while to upload a large book to PB, I can darkly mutter “is that Godot fellow *ever* going to get here?”

I really hate Google, as should be apparent – although I use their products, quite a bit, I confess it!

There are lots of other tools which have all of the advantages cited here, but maybe fewer of the disadvantages.

I kind of hate PB, for a specific technical reason I can explain if anyone is interested ... but many people like it, so I can put that negative feeling aside.

You have to love it's learning curve!

Slide 24/ 25