1-2019

Addressing the Technical Challenges of Open Educational Resources

Steven Ovadia
CUNY La Guardia Community College

Addressing the Technical Challenges of Open Educational Resources

Steven Ovadia

abstract: As open educational resources (OER) becomes a more established concept, its growth and continued success not only center around the creation of new content but also depend upon the development and evolution of existing content, licenses permitting. The evolution of and persistent access to existing content requires easily editable works and the ability to download local copies of OER files so that no technical limitations hinder what the license permits. Such measures as text files, OpenDocument formats, and version control present a way to keep OER content editable and shareable, in compliance with licenses. Librarians are uniquely positioned to help users understand those solutions.

Introduction

As open educational resources (OER) becomes more popular and accepted, its continued spread will depend not only on the production of new work but also upon the refinement and improvement of existing content, licenses permitting. This development of and continued access to existing content requires works that can be easily edited and OER files that can be easily downloaded as local copies. Without those technical permissions, a license offers only theoretical rights, with users unable to exercise their rights due to the technical limitations of a file. Until the OER community addresses these issues, the technical challenges will prevent more widespread adaptation of material. These barriers impact OER creators, consumers, and curators. Librarians, with their expertise in pedagogy, technology, and copyright, are well equipped to address these challenges.

Before detailing the technical challenges of OER, both OER and openness should be defined. As Jan Hylén wrote, there is unavoidable ambiguity to these terms, in part because OER is still a relatively new concept.[1] The definitions used here should be considered working definitions for the purposes of this paper.

The United Nations Educational, Scientific and Cultural Organization (UNESCO) first used the term *open educational resources,* recommending a definition of "the open provision of education resources, enabled by information and communication technologies, for consultation, use and adaptation by a community of users for non-commercial purposes."[2] Education resources will be defined as any and all material related to a class, such as syllabi, assignments, videos, slides, and textbooks.

This paper uses the "Four R's of Openness" developed by John Hilton III, David Wiley, Jared Stein, and Aaron Johnson. Hilton and his coauthors expressed openness as a continuum of four levels of openness: (1) reuse, (2) redistribute, (3) revise, and (4) remix.[3]

Reuse means providing access to the material. Reuse is the most basic level of openness, and it involves not just being accessible online but also granting users the right to download content, so that it is always ready for use. Hilton, Wiley, Stein, and Johnson refer to this as *source-file access*. For example, a cellular biology reading from a website that cannot be downloaded cannot be relied upon to be reusable. If the website disappears or if someone trying to access the material loses Internet access, then reuse becomes impossible.

Redistribute means the work can be shared with others. The ability to download files, rather than just link to them, also makes this process more stable. Content creators might allow individuals to download their work but might not allow it to be shared or redistributed. An instructor wishing to share a cellular biology article with students might lack the legal right to e-

mail the article to the class. Conversely, if the professor has no technical means to download the article and share it, then a license allowing reuse is technologically impossible to fulfill.

Revise means the work can be altered. Like reuse and redistribute, this is another scenario where the actual file is more useful than a link to a website. While text can often be copied and pasted and images can usually be saved and downloaded, it is much easier for someone trying to revise content to have all the necessary files in an easy-to-download, editable package, rather than trying to reconstruct hosted content locally. In the example of the cellular biology reading, the right to revise would allow the instructor to alter the article, perhaps changing the order of ideas or adding better images, and then redistribute the revised work to students.

Remix means that multiple works can be combined. Like revise, this process is made easier with access to the electronic files, rather than links pointing to hosted content. For example, combining two articles on cellular biology, one with beautiful pictures and the other with clear text, would form one comprehensive article that takes the best elements of each remixable article.

Each level of openness depends upon the step before it. The cellular biology article cannot be remixed if it cannot be reused. Work that cannot be reused also cannot be redistributed. Licensing controls the degree of openness and what may be done with the work. The license is the legal mechanism controlling how the work can be used, as copyright expert Michael Carroll explains:

> Copyright law supplies the baseline terms of use for almost all information on the Internet. These terms can be altered if the copyright owner grants a license or permission to do something that would otherwise infringe on copyright. Traditionally, copyright owners granted licenses to specific persons or entities. More recently, copyright owners seeking to grant permission to everyone have issued public licenses broadening the range of permitted uses, subject to certain conditions. [4]

Carroll notes that the licensing concept developed by the nonprofit organization Creative Commons is one mechanism for a content creator to control how the four R's apply to a given work. There are other types of licenses, but Creative Commons will be discussed here, purely for the purposes of example. Most licenses will regulate the usage of a given work in a similar way, using their own terminology. Creative Commons uses four parameters that can be combined with each other to create different licenses.[5] Every Creative Commons license must include attribution, meaning whoever is using or reusing the work must give credit to the creator. A provision called ShareAlike requires the work, whether redistributed, revised, or remixed, to keep the terms of the original license. For instance, once a work is revisable under a ShareAlike license, all future works derived from the original must also be revisable. The NonCommercial rule prevents a work from being used for commercial purposes. Finally, NoDerivatives forbids a work from being revised or remixed, meaning the work can only be reused or redistributed, but not changed.

One current challenge of OER is that the license may permit one action, such as revision, but the technical underpinnings of the OER work itself may prevent what the license allows. Consider the example of a crime statistics website, used for a class, that subsequently disappears. A license permitting reuse allows the user to have a personal copy of the data. Without the ability to download a personal copy, however, the instructor who depended upon the site is left with only a theoretical right if the site becomes unavailable. Consider a YouTube video that has a license allowing the revision of a video. How does a user revise a YouTube video without the original, editable file? Such inconsistencies often prevent the wider adaption of OER materials.

Hilton and his coauthors acknowledged the technical issues that impact the openness of a resource, recommending "that OERs are designed in such a way that users will have access to

editing tools, that the tools needed will not require a prohibitive level of expertise, and that OERs are meaningfully editable and self-sourced."[6] Ilkka Tuomi drew comparisons between OER and open source software, but a key distinction between the two is that open source software requires the underlying source code be made publicly available. OER has no such formal requirement.[7] The Organisation for Economic Co-operation and Development declared:

> To adapt or modify a digital resource it needs to be published in a format that makes it possible to copy and paste pieces of text, graphics or any published media. This means that noneditable formats, such as Flash (.swf) and Adobe Portable Document Format (.pdf), do not qualify for a higher level of openness. Examples of more open formats are HTML [Hypertext Markup Language], ODF [OpenDocument Format], RTF [Rich Text Format], SVG [Scalable Vector Graphics], PNG [Portable Network Graphics] and others. However, these formats are more difficult to use and thus exclude people lacking the necessary skills.[8]

Richard Baraniuk also identified this challenge, writing, "Unfortunately, widely used OER formats like PDF yield materials that are open in theory but closed in practice to editing and reuse, rendering them often merely 'reference' materials that are to be seen and not used. This stifles both innovation on the materials and also community participation."[9] As these writers suggest, OER is not merely about accessing material, but about sharing ownership of content. Without access to the files used to create OER, modifying and sharing becomes challenging, if not impossible.

Technical Challenges

Reuse and redistribute are closely related because, in a technical sense, Web servers are the mechanism for distributing Web-based content. An instructor can e-mail students a link pointing to specific website content, but if the content is merely displayed and cannot be downloaded (or is difficult to download), then the instructor has only highlighted the content. If the instructor e-mails a file to students, or allows students to download it, however, the instructor becomes the

distributor. Content must be accessible for it to be reused and redistributed. People who have downloaded a file always have the right of reuse and redistribution for as long as they have the source file. They are not dependent upon a server letting them see the content because they can keep that content on their own device.

While the ability to download and share files goes a long way toward providing perpetual access to OER content, there is another potential issue: file format. If a document or piece of media is not in an open format, meaning one that can be accessed by a nonproprietary program, there is no guarantee that students or instructors will be able to access the work. For example, many instructors upload their OER content in the .docx format used by Microsoft Word. As of this writing, most non-Microsoft word processors, such as LibreOffice, can open and translate .docx files so the files are readable. But if Microsoft were to change the format so that only Microsoft products could open them, users would lose access to these files. The content would be reusable, but only by people who have compatible software. The danger exists with any proprietary format, from Flash to QuickTime. Any file that requires proprietary software to open it is not truly open, since lack of the correct software (whether for financial or technical reasons) can prevent access.

The more proprietary formats there are in the OER ecosystem, the greater the chance they might someday become outdated and inaccessible, or will require technical skill and patience to access. Matthew Kirschenbaum's book *Track Changes* used the work of the American poet Lucille Clifton as an example of the dangers of proprietary formats.[10] Clifton wrote using a Magnavox Videowriter, a word processor with built-in screen and printer sold in the 1980s and now obsolete. As a result, her work became accessible only after much effort:

> The proprietary formatting of Clifton's Magnavox Videowriter diskettes posed a significant challenge in terms of capturing and processing their data. Forensic images of each disk have been

> captured but cannot currently be accessed and rendered. Due to these limitations, the original Magnavox Videowriter was used to print out copies of each of the diskettes' files, which were then scanned using OCR [optical character recognition] technology to create searchable PDF files.[11]

Looking ahead 10 or 15 years, there might not be enough time, interest, or funding to unlock all the content trapped in outdated, no-longer-supported proprietary formats. A work's license can decree the work reusable, but if there is no software to open it, the work is not actually reusable in practice.

File format also plays into the technical issues of revision and redistribution. Revision and redistribution are the ability to edit a work. However, just because a license grants the right to edit a work does not make it technically feasible or simple. Tel Amiel wrote that most OER repositories "are focused on the distribution and dissemination of resources and [provide] little guidance or tools for those who wish to make revisions or remix existing resources."[12] The focus on distribution over revision amplifies the challenges of revising and remixing OER content.

For instance, many instructors release OER as PDF files. This is an accessible format, with most devices and operating systems able to open the files. However, as mentioned earlier, the editing of such documents can be problematic. The content of many PDFs can be copied and pasted into other tools, such as text editors and word processors, but the files lose all formatting and images in the process. The license might make editing the work legally possible, but the technical limitations of the PDF format make such editing challenging. It may not be hard to reformat a short document or assignment, but a longer document, such as a textbook, could require a huge investment of time. The amount of effort required to edit the content might prevent the work from being used for OER purposes.

PDFs are not the only example of this situation. Revision can also be an issue with audiovisual material. For example, with a YouTube video, even if there were a mechanism to

easily download videos, someone who chose a license allowing work to be edited still extends only a theoretical right, not a practical one. It is challenging to edit audio or video that has already been published. Instead, someone serious about reworking a video would want the raw files, so that they might do things like redub audio and reorder shots.

A similar issue exists with charts. While a chart's license may allow for revision, unless there is access to the underlying data used to build the chart, it can be challenging, and even impossible, to revise. If someone builds an OER textbook using charted data, someone else might want to rework those charts for his or her own students, perhaps using different variables in the revised charts. With just an image of a chart, the new chart will need to be totally rebuilt, with the variables manually transferred from the image into whatever will be used to build the new chart. With access to the data used to build the chart, however, it becomes much easier to create new charts based upon the same data. Amiel summarized the issue, writing, "Making the original source file available in an open format greatly increases the potential for revision and remix of existing materials."[13] Likewise, Daniela Luzi, Rosa Di Cesare, Marta Ricci, and Roberta Ruggieri advocated for the use of flat files, data files that can be opened with a text editor, in their study of open data in repositories: "The use of flat files, that is files that transform a record of a database into text, can be easily exchanged because they are not connected with proprietary systems."[14]

Another challenge to reuse and remix is technical issues from the end user's perspective. Gráinne Conole reviewed OER case studies. Examining the OpenLearn Project, an effort by the United Kingdom's Open University to make its educational resources available free on the Internet, Conole discovered little evidence of content being reused and reposted back to the site.[15] She attributed this to "both a technical (lack of understanding of XML) and pedagogical

(lack of experience of redesigning and not wanting to alter existing perceived 'good' content) barriers to reuse."[16] OER participants need to understand not just the creation of formats but also the technical use (and reuse) of them. Without that understanding, OER content might be used and accessed, but not revised or remixed.

Overcoming the Technical Challenges

None of these technical challenges are insurmountable. Rather, they represent a new component to OER. In addition to thinking about content, which is obviously important, OER producers can release their material in ways that minimize these technical limitations.

Text Files

The broader challenge hindering the four R's of OER are file formats that impact access, the ability to edit, or both. In terms of text-based documents and assignments, one simple solution is the text file, an open format that is accessible on just about any device or operating system. Also, because text files cannot be formatted, meaning they cannot include things like bullets, underlining, font changes, and other changes to the appearance of the text, the text is modular and, in some respects, easy to work with. Material copied and pasted from a PDF might have line breaks that would need to be manually fixed, but a text file would have no such formatting to hinder moving text between applications. Karl Stolley's "Lo-Fi Manifesto" explicitly identified text files and text editors as tools for creating content that is "modular and swappable, and can be combined or replaced as needed."[17] For users to engage with OER text at the revise or remix level, they need to manipulate it, and text files are an easy way to ensure the text can always be modified.

If the content creator wants to preserve formatting, there are ways to do so using text files. One way involves using Markdown, a simple syntax that allows third-party tools to

transform plain text files. For instance, while words cannot be italicized in a text file, Markdown allows the use of asterisks to indicate a word should be italicized. Within the text file, the italicized word or sentence *might look like this* but, once transformed, *might look like this*. Markdown is plain text that can be read and edited on just about any device with a text editor, now and for the foreseeable future. This makes text accessible to just about anyone. When the Markdown text is viewed in a browser or printed out, the text becomes formatted. Figure 1 shows a Markdown document, with the left pane showing the raw Markdown and the right pane previewing it as formatted text.

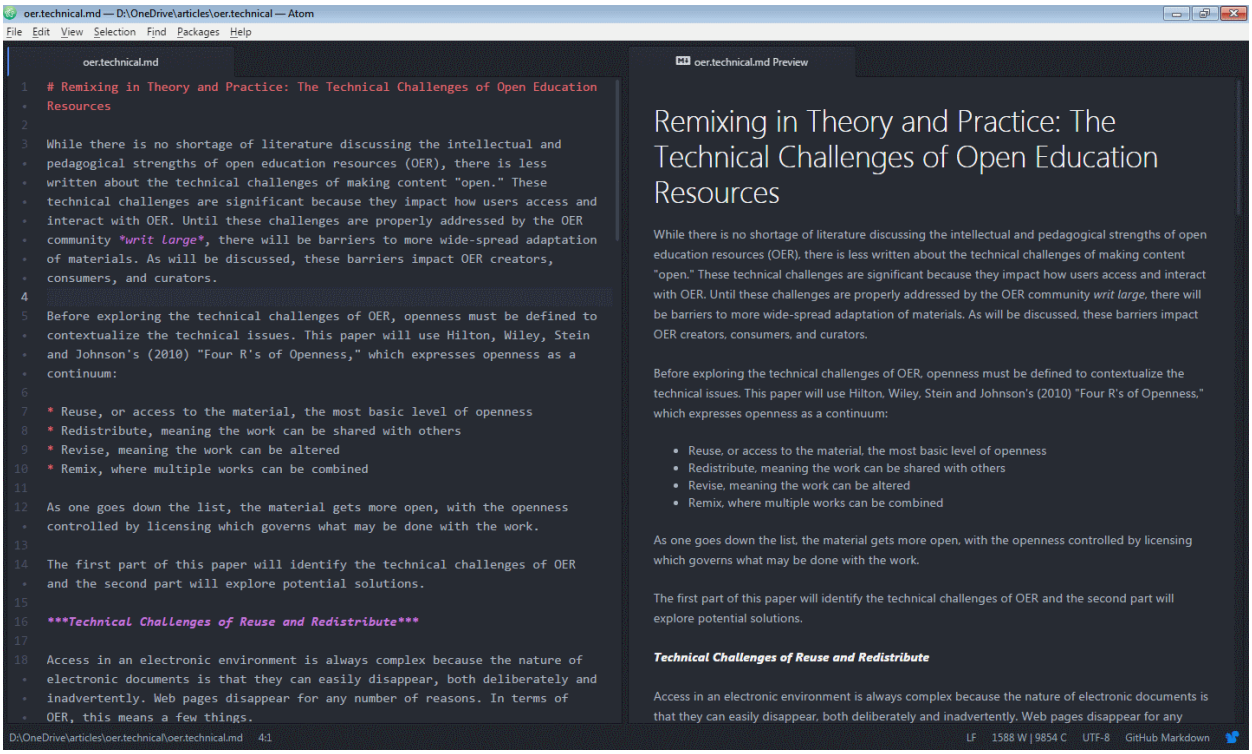

Figure 1

The ability to transform Markdown also makes it useful because it separates content from format.[18] For instance, rather than keeping a presentation in a format such as PowerPoint, a third-party tool can change properly formatted text into a presentation. The Pandoc project (http://pandoc.org/) allows markup languages to be changed into other formats such as PDF,

HTML, and OpenDocument, a format discussed later in this paper.[19] With Pandoc, the user runs a snippet of command-line code to change a text-based file into a different file format. This makes it easier to reuse and remix content, although understanding and executing these commands requires a certain level of technical expertise. For example, someone might have a Markdown document outlining a few key composition principles. That Markdown document can be changed into a PDF, to be printed and shared with a class, by using Pandoc. Using a different Pandoc command and that same Markdown document, it becomes a presentation slide. If the instructor is so inclined, the document can be changed to a PowerPoint slide or an HTML-based presentation format. If that same Markdown document were lengthened, the professor might eventually use Pandoc to transform it into an EPUB, which is an e-book format. In these examples, the format of the work does not dictate how it is used. Instead, the content dictates usage, with the end user choosing an appropriate format. It is possible to change a PowerPoint into a class assignment, but it is much easier when the document is in transformable plain text. Plain text is always accessible, across varying levels of bandwidth and technology, and will likely remain accessible for some time to come.[21] Plain text will also come into play when discussing distributed version control later.

OpenDocument Format

Plain text is not always an option. For instance, plain text cannot replace audio and visual materials. But even text-based files sometimes require more sophisticated formatting to make sense. Such files could be graphic-intensive slides or word-processed documents. They could also be books with sophisticated design aesthetics, as seen in such fields as graphic design. In these situations, it makes sense to use the OpenDocument format, which is a standard for document files. OpenDocument allows them files to be opened and edited by a variety of tools

by “providing a standard format for storage and exchange of office documents.”[20] These files can, in theory, always be opened, since the standard used to create the file is publicly viewable. The files, which can be anything from a word-processed document to a presentation, spreadsheet, or graphic, remain openable and editable across different programs and platforms, and across time. OpenDocument not only future-proofs the content against a file format becoming unsupported but also allows people using different operating systems and software suites to access and edit the files, without having to purchase anything. The OpenDocument format thus makes sure content will always be both editable and accessible. In fact, one of the purposes of OpenDocument was to “preserve the structure of the document to allow re-editing (for example, footnotes must be stored as structured footnotes, not just as text in the document that looks like a footnote).”[22] This helps to prevent a situation like Clifton’s, where unlocking the content required a great deal of labor.

Another advantage to working with OpenDocument formats and files is that a local copy of the file is stored on the local network. This file can then be shared, rather than or in addition to uploading the file into a cloud-based service, such as Google Docs. Making the actual file available to users allows them to download it themselves, not only keeping it accessible in case the hosting service disappears but also allowing users to redistribute the work without depending upon any other servers. The work remains redistributable as long as copies of the files exist. As mentioned previously, the danger of sharing a link is that if the site hosting the link disappears, so does the content—regardless of what the license indicates about the openness of the content. When users can download the file, though, they can do whatever the terms of the license allow. The ability to download a file, rather than just to access it, empowers the end user.

OpenDocument files provide a stronger likelihood of a file remaining revisable and remixable, license permitting.

Distributed Version Control

Revising and remixing can be an important part of OER work, but tracking changes to a work can be challenging. William Wong identified "a journaling system for tracking changes" as an important component of open source textbooks.[23] Many OER hosting platforms allow users to upload variations of a work, but this depends upon the user uploading the new version to the same platform after changing the work. An automatic notification system may not alert the original content creator that the work has been changed. Even on platforms where that functionality does exist, the original content creator still must review the modified work to see how others altered the file and what was kept from the original. Another issue is that a person revising or remixing a work, or both, might not want to upload the new version to the platform on which it was found. For instance, the person might instead wish to use an institutional repository. While there are many potential barriers to sharing revised and remixed content with the original creators, this section will address these three: (1) notifying the original content creator that the content has been modified; (2) showing the original content creator how the work has been revised and remixed; and (3) linking the changed content to the original.

Distributed version control addresses all three of these challenges. Distributed version control is most commonly used by software developers to collaboratively develop and refine code in much the same way certain licenses allow OER content to be developed and refined. *Distributed* means that people work across geographic areas, as opposed to working on a single file in the same place. This system allows developers around the world to work on the same set of code. Version control tracks changes, much as a word processor can track changes to a

document. Version control keeps track of how a text-based work has changed, allows end users to easily see those revisions, and permits them to revert to a previous version, if needed. This coding workflow potentially applies to OER content, with the caveat that distributed version control typically only works with plain text. That is one reason plain text is so important to making OER content accessible and revisable. Srikesh Mandala and Kevin Gary argue, "The problem of an instructor obtaining, customizing, and integrating curricular content from multiple sources is not unlike the open source problem," so it makes sense to use a solution from the open source world.[24]

Distributed version control can be thought of as two parts: one part is the software that tracks and controls the versioning of documents, and the other part is the repository where the content is held. The most commonly used version control is software called Git (itself an open source tool). There are many repositories where content can be held. Currently, GitHub is one of the more popular, but there are alternatives.

The simplest way to show how Git could work in an OER context might be to demonstrate a sample workflow from start to finish, sharing the perspectives of the content creator (Creator) and the content modifier (Modifier). The process begins with the Creator having uploaded an OER syllabus into a publicly viewable repository and having assigned that syllabus a permissive license.

The Modifier sees the syllabus online and has some thoughts on how to improve it, so he or she installs Git. The Modifier forks the syllabus, making a copy of the Creator's repository (allowing the original work to remain intact). The Modifier downloads the forked repository and now has all of the Creator's repository-hosted files copied onto his or her local computer. If the

Creator's repository disappeared, the local files would still be available to anyone who already downloaded them (including the Modifier).

The Modifier edits the syllabus on his or her own computer, changing the content so it works for the intended purpose. The Modifier can offer these changes back to the Creator, using what Git calls a *pull request*, with the Creator and Modifier discussing the alterations the Creator wants to accept into the original syllabus. This conversation takes place in the comments of the pull request. The Creator can also accept only the changes he or she wants. Whenever and however that process takes place, the Creator can eventually merge the changes into the original document, accepting the Modifier's revisions into the original work in the online repository. The Modifier can also keep the changes as a separate fork of the original, making it a second work derived from the original.

At the end of this process, the original syllabus has been modified. The changes made to the Creator's work are all visible online in the repository. Anyone disagreeing with the changes can simply download previous versions of the work, which are all accessible in the repository. Figure 2 shows the history of a file within a Git repository with dates and *commit messages*, notes describing the nature of changes made to files.

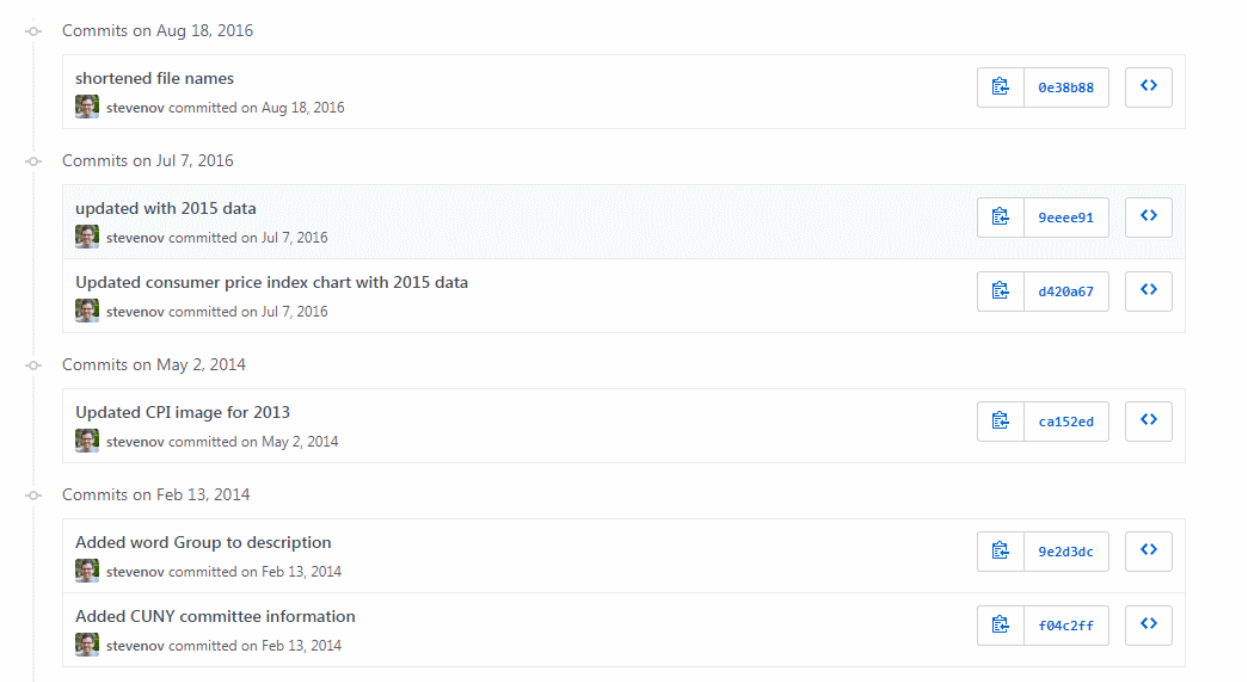

Figure 2

Figure 2 shows how the file evolves with just two participants, Creator and Modifier. The work shifts and evolves as more Modifiers enter the process. Consider the forks and changes and enhancements, all tied together within the repository. The history of the work is there for anyone to see, much like the history of a Wikipedia entry. The Git methodology allows for work to easily be revised and remixed while also ensuring it remains redistributable via the local copies saved on individual machines. If a repository disappears, the work survives, existing on local computers and perhaps eventually in new repositories.

The question then becomes, why is this workflow not more commonly used for OER? One reason is that Git can be complex and confusing to use, relying on idiosyncratic syntax and commands, such as *pull* and *fork*. While there are some graphical tools, understanding how Git works with files and changes still requires technical know-how. Mandala and Gary aptly noted that for version control to work with educators, its complexity must be hidden from them.[25] To contribute changes to a Git repository requires at least three commands: (1) *git add* identifies the

files that will be uploaded to a repository; (2) *git commit* is similar, but allows the user to add a message describing the changes; and (3) *git push* finally uploads the file or files to the repository. This workflow might make sense to some educators but would likely confuse and frustrate others.

Another consideration is that OER content within Git repositories has not yet hit a critical mass with educators. As more OER content migrates into these repositories, more educators might become motivated to learn how to use Git. Until Git is simpler to use, however, it will likely remain more a theoretical option than a realistic one.

Git creates a detailed record for all changes made to a work and links between different versions of a work. This information is publicly available within a given repository. In terms of pedagogical research, Git provides an amazing insight into the evolution of a work that could be helpful to researchers studying how, and perhaps even why, OER content evolved. This kind of information is much more difficult, if not impossible, to track in other OER platforms, a compelling reason to encourage educators to use Git (and one that speaks to archivists). Stolley also discussed the importance of iteration, which can be both demonstrated and facilitated by version control, showing the "slow and steady improvement of existing work as well as experimentation and parallel, alternate approaches to production."[26] This represents an opportunity to show not just the work but also how others have changed it, with that evolution perhaps informing future directions for the content. Someone could see an idea in the commit history of a file that might not have worked at the time but could be more successful in the present. That type of document view is impossible with a static document that captures the current moment but not the iterations that led to that moment.

Recommendations

In terms of removing the technical friction points, a few ideas might address some of these challenges. These ideas include an easier interface for Git, user education about the nature of files and their technical limitations, and document formatting standards.

An Easier Interface for Git

As a command line program, Git requires the user to know several basic commands. As repository work grows more complicated, so do the commands. An easy-to-use graphical interface to Git would make it a more practical tool for revising and remixing OER content. There are already Markdown editors with graphical user interfaces. If one (or some) of these editors easily integrated with Git, it would make both Git and Markdown more viable tools for educators. GitHub, a popular repository, has a graphical tool, but using it ties users to GitHub. There are other third-party Git graphical interfaces, but they require a strong understanding of the intricacies of Git and might not be much easier to use than the commands. There is a need for a graphical tool that harnesses the complexity of Git while shielding users from that complexity.[27]

User Education

While the legal aspects of traditional OER training are important (and a place where many librarians have already made their mark), there is also a need for users to understand how file format impacts the ability of others to work with their file. Cheryl Cuillier and her coauthors did a thorough job of explaining this in their guide to modifying an open textbook, but the importance of file format should ideally be understood prior to the decision to make content available as OER.[28] For instance, it would be useful if educators created all pedagogical material in open, editable formats, making the material that much easier to transition into OER

repositories once the decision is made to make the work available as OER. Many users also need education around the use and reuse of these files, so they understand how to work with what might, to some, be new file formats. This kind of training aligns with the education already being done in many academic libraries.

Document Formatting Standards

Document formatting standards could also help remove technical challenges by giving faculty guidelines on how to build and create OER content. The standards could be everything from plain text file templates, to suggested lists of open file formats, to recommended best practices for creating and saving files. Such content creation is much more manageable with a tool like Git, which only tracks plain text files, although it does allow users to upload (and download) media files and can track changes to these kinds of files via notes made when users change images and re-upload them to the repository. Recommending users write in plain text and use Markdown (or some other agreed-upon markup) would help them become familiar with accessing, editing, and creating these kinds of files. Darrell Porcello and Sherry Hsi discussed the importance of common metadata to make science, technology, engineering, and mathematics OER content easier to find.[29] Similar alignment around file format would make OER content easier and more consistent to work with. Providing templates would make sure the format allows revision and remixing of content and would also train OER content creators about these kinds of files.

Academic repositories, which typically host completed work (even prepress publications can be considered finished), often as PDFs, could also make more of an effort to host open, modifiable formats. They might perhaps even link out to Git repositories holding the evolving work, reinforcing the idea that many OER works can never be considered completed.

A common thread to addressing these technical issues is making formats and tools easier to work with. These technical barriers exist, in part, because many OER content creators are educators without formal technical training. It might be unrealistic to expect faculty interested in OER to learn Git, but it is perhaps more realistic to encourage the development of tools that make something like Git as easy to use as a word processor. This kind of work requires people who understand OER from pedagogical, legal, and technical perspectives. Many librarians already fit this description. Mark Eaton studied how librarians used GitHub for code and found they had greater reach and productivity than a comparison group.[30] Librarians can use those same skills to help share content in Git repositories. As OER grows more popular and attracts more attention, more people fluent in the various aspects of OER, librarians and nonlibrarians alike, will enter the ecosystem. They will improve and even create new OER tools to facilitate sharing from a technical as well as a legal standpoint. Until then, however, user education and standards are the best strategy for ensuring OER content gives users all the technical rights extended by the license of the work.

Conclusion

Any incongruity between what a license permits and what a file format allows is a barrier to expanding the reach of OER materials. Plain text files shared via Git go a long way toward making sure end users can work in accordance with the intent of the license. This is important because, for OER to grow, content not only needs to be created and shared but also needs to be revisable. The switch from sharing content to thinking about how it will be used is important because truly open content allows users to engage with work in different ways, from editing to remixing. This type of engagement requires file-level access to content, not just the ability to call up a Web page. If OER content is merely placed online, without considering how the content can

be used, it will be challenging for other users to share ownership in it. The content never truly belongs to the users who find it because of the limitations of how it can be used. To realize the full potential of OER, users need to do more than access it: they need to engage with OER content in a meaningful, transformative way—as always, license permitting.

Steven Ovadia is professor and deputy chief librarian at LaGuardia Community College, an institution of the City University of New York (CUNY) system; he may be reached by e-mail at: sovadia@lagcc.cuny.edu.

Notes

1. Jan Hylén, *Open Educational Resources: Opportunities and Challenges* (Paris: Organisation for Economic Co-operation and Development, 2006), https://www.oecd.org/edu/ceri/37351085.pdf.
2. United Nations Educational, Scientific and Cultural Organization (UNESCO), *Forum on the Impact of Open Courseware for Higher Education in Developing Countries: Final Report* (Paris: UNESCO, 2002), 24, http://unesdoc.unesco.org/images/0012/001285/128515e.pdf.
3. John Hilton III, David Wiley, Jared Stein, and Aaron Johnson, "The Four R's of Openness and ALMS [access to editing tools, level of expertise required, meaningfully editable, and source files] Analysis: Frameworks for Open Educational Resources," *Open Learning* 25, 1 (2010): 34–44.
4. Michael W. Carroll, "Creative Commons and the Openness of Open Access," *New England Journal of Medicine* 368, 9 (2013): 789.
5. Creative Commons, "About the Licenses," https://creativecommons.org/licenses/.

6. Hilton, Wiley, Stein, and Johnson, “The Four R’s of Openness and ALMS Analysis,” 43.

7. Ilkka Tuomi, *Open Educational Resources: What They Are and Why Do They Matter* (Paris: Organisation for Economic Co-operation and Development, 2006), 7, http://www.meaningprocessing.com/personalPages/tuomi/articles/OpenEducationalResources_OECDreport.pdf.

8. Organisation for Economic Co-operation and Development (OECD), *Giving Knowledge for Free: The Emergence of Open Educational Resources* (Paris: OECD, 2007), 34, https://www.oecd.org/edu/ceri/38654317.pdf.

9. Richard G. Baraniuk, “Challenges and Opportunities for the Open Education Movement: A Connexions Case Study,” chap. 15 in Toru Iiyoshi and M. S. Vijay Kumar, eds., *Opening Up Education: The Collective Advancement of Education through Open Technology, Open Content, and Open Knowledge* (Cambridge: MIT Press, 2008), 230.

10. Matthew G. Kirschenbaum, *Track Changes: A Literary History of Word Processing* (Cambridge, MA: Belknap Press, 2016), 213–14.

11. Emory Libraries and Information Technology, Emory Finding Aids, “CLIFTON, LUCILLE, 1936–2010. Lucille Clifton papers > Born digital materials,” https://findingaids.library.emory.edu/documents/clifton1054/series11/.

12. Tel Amiel, “Identifying Barriers to the Remix of Translated Open Educational Resources,” *International Review of Research in Open and Distance Learning* 14, 1 (2013): 126–44.

13. Ibid., 138.

14. Daniela Luzi, Rosa Di Cesare, Marta Ricci, and Roberta Ruggieri, “Enhancing Diffusion of Scientific Contents: Open Data in Repositories,” *Grey Journal* 8, 2 (2012): 78.

15. Gráinne Conole, *Designing for Learning in an Open World* (New York: Springer, 2013).

16. Ibid., 231.
17. Karl Stolley, “The Lo-Fi Manifesto, v. 2.0,” *Kairos* 20, 2 (2016) http://kairos.technorhetoric.net/20.2/inventio/stolley/.
18. Steven Ovadia, “Markdown for Librarians and Academics,” *Behavioral and Social Sciences Librarian* 33, 2 (2014): 121.
19. Albert Krewinkel and Robert Winkler, “Formatting Open Science: Agilely Creating Multiple Document Formats for Academic Manuscripts with Pandoc Scholar,” *PeerJ Computer Science* 3, e112 (2017), http://doi.org/10.7717/peerj-cs.112.
20. Barry Leiba, “From the Department Editor,” *IEEE* [Institute of Electrical and Electronics Engineers] *Internet Computing* 13, 2 (2009): 84.
21. Markdown is not the only plain-text markup language. There are others, such as AsciiDoc and LaTeX, which is often used in science and mathematics.
22. Rob Weir, “OpenDocument Format: The Standard for Office Documents,” *IEEE Internet Computing* 13, 2 (2009): 85.
23. William Wong, “Open-Source Textbooks Need to Be Configurable,” *Electronic Design*, April 26, 2017, http://www.electronicdesign.com/embedded-revolution/open-source-textbooks-need-be-configurable.
24. Srikesh Mandala and Kevin A. Gary, “Distributed Version Control for Curricular Content Management,” IEEE Frontiers in Education Conference, Oklahoma City, October 23–26, 2013, 802.
25. Ibid., 802.
26. Stolley, “The Lo-Fi Manifesto, v. 2.0.”
27. Mandala and Gary, “Distributed Version Control for Curricular Content Management.”

28. Cheryl Cuillier, Amy Hofer, Annie K. Johnson, Kathleen Labadorf, Karen Lauritsen, Peter Potter, Richard Saunders, and Anita Walz, *Modifying an Open Textbook: What You Need to Know* (Minneapolis: Open Textbook Network, 2016), https://press.rebus.community/otnmodify/.
29. Darrell Porcello and Sherry Hsi, "Crowdsourcing and Curating Online Education Resources," *Science* 341, 6143 (2013): 240–41.
30. Mark Edward Eaton, "A Comparative Analysis of the Use of GitHub by Librarians and Non-Librarians," *Evidence Based Library and Information Practice* 13, 2 (2018): 40.