

Documentation to the People: Building Empathy into Technical Documentation for Digital Archiving

Short Paper

L. Work

University of Virginia
160 McCormick Road
Charlottesville, VA 22904 USA
lw2cd@virginia.edu

H.E. Kelly

Indiana University
1320 E 10th Steet
Bloomington, IN 47401 USA
heidowdi@indiana.edu

ABSTRACT

The field of digital archiving is situated in a tenuous position between the archives and information technology, where the humanity of archival labor is often erased on one side and that of users may be ill-considered on the other. To that end, the quality of technical documentation is paramount to the success of digital archiving efforts, but it often falls victim to a lack of empathy for practitioners, users of our tools and collections, and those represented in the archives. This paper aims to provide initial community suggestions for creating better, more useful documentation built around the concept of empathy.

KEYWORDS

Documentation, Digital Archives, Digital Curation, Ethics

ACM Reference format:

L. Work and H.E. Kelly. 2017. Documentation to the People: Building Empathy into Technical Documentation for Digital Archiving. In *Proceedings of iPRES 2017, Kyoto, Japan, September 2017 (iPRES2017)*, 5 pages.

1 INTRODUCTION

In *Rethinking Repair*, Jackson presents two potential worlds: one is “always-almost-falling-apart”, while the other is one of constant progress, reinvention and improvement. The crux, he argues, is repair; it is the ongoing maintenance of our technologies [15, p. 222]. Indeed, we must “love our monsters” [19], but so often the focus on innovation and newness overshadows the maintenance needs of existing technologies. In the field of digital archiving- which, for practical purposes, we consider here as a related component of the greater digital preservation field- we feel the impact of technological degradation acutely and yet our own maintenance work is lacking. Ongoing development of technical solutions (including software, applications, and hardware), to the challenges faced in managing born-digital content within libraries and archives often neglects repair work. This is especially clear in the realm of technical documentation, a piece of digital archiving work that is not often incentivized. Further, often when technical documentation work is undertaken, it is done so to reflect only the knowledge, experience, and needs of the staff and institutions creating such documentation, which thereby leaves out large populations that should be represented in digital archiving work.

Drabinski [11], based on the seminal work of Bowker, argues that the fundamental act of documentation is one of building a professional narrative. In this case, we argue that the development of better, more empathic technical documentation practices is one of the most fundamental elements needed for building a more diverse field, which will in turn make us better at representing both our users and those who are included in our archives.

The field of archival science has recently explored the issue of empathy as part of the archivist’s role (e.g., Caswell & Cifor). We aim to build on this exploration by arguing that the same critical lens should be focused on our digital tools and practices. Additionally, there are extensive theoretical studies on representation in technology and in the online realm (e.g., Nakamura; Wolmark) and we seek to build a bridge between that theoretical work and the functional practice of those undertaking technical work in digital archives. There is a need for sustained inquiry regarding how we address issues of representations of race, gender, and inequality in technical arenas [17], and we argue that such an inquiry should be integrated into every aspect, rather than separated as its own subfield. The purpose of this paper is to provide initial suggestions for the digital archiving community in order to develop a more human-focused, empathic approach to technical documentation. We argue that by creating documentation that is underpinned by empathy for ourselves, our users, and those in our archives, we will improve our community, our tools and software, and our practices.

2 THE STATE OF DIGITAL ARCHIVING TECHNICAL DOCUMENTATION

2.1 Current State

A discussion of the current state of technical documentation in digital archiving must begin with the fact that, while documentation is ubiquitously known to be of importance from both a technical standpoint as well as a preservation standpoint, it is generally not prioritized. On an individual tool level, documentation is often prone to institutional specificity and the single-point-of-failure issue [12]. Specifically regarding the implementation of open source software (OSS), libraries cited poor documentation as second only to the need for skilled staff that could provide support for an OSS system in terms of barriers [9, 26]. In reviewing the principles of various technical development communities, this dearth of focus on documentation

is apparent. The Hydra community, for example, only mentions documentation in the context of code in its overarching principles, and does not provide much guidance to developers in creating usable documentation [28]. The Spruce Mash-Up Manifesto asserts that development projects should “make it easy to use, build on, re-purpose and ultimately, maintain” but does not specifically highlight the need for good documentation [27]. These well-intentioned principles serve a point by Bayer and Muthig, that often documentation processes are defined without justification, so those producing it are not clear about “what is being done with the documents they produce and for whom they are producing them” [3, p. 2].

In terms of documentation to support digital archival efforts at the institutional level, digital preservation policies tend to be the most ubiquitous, as growth in both administrative policy focus and digital content in cultural heritage institutions have helped solidify the need for these policies to exist [25]. Standards and certification also necessitate related types of high-level documentation that aid practitioners in understanding their institution’s specific landscape for digital archiving, such as collection development policies and mission statements. For example, many larger institutions strive to meet the requirements for Trustworthy Digital Repositories (TDR), for which there are extensive explanatory documentation requirements. Levels of TDR-readiness vary widely, however, and so does the quality of organizational and technical documentation that ultimately supports digital archiving.

At a digital archives practitioner level, the poverty of technical documentation is felt acutely in systems and tools that require the content creators and users to change their needs in order to be able to use the technical solution. However, the field is becoming more adept at requesting user-focused development with better documentation. In creating a list of Minimum Necessary Requirements for Developing OSS Tools for Digital Preservation, attendees of the OSS4Pres 2.0 workshop cited “Provide publicly accessible documentation” as the first item of necessity. “Keep documentation up-to-date and versioned” followed that requirement, as well as the need to provide “a documented process for how people can contribute to development, report bugs, and suggest new documentation.” Participants also suggested well-documented integrations with other tools and systems, and the need for validated use cases [10]. All of these items refocus development projects on the needs of users and future stewards of the tools and software.

2.2 Relevant Developments in the Field

In the larger field of digital preservation, momentum has been building in terms of reconsidering traditional beliefs and practices, embracing concepts like ethical collection practices and community building. These initiatives reflect the growing interest and need for preservation across communities, such as activists, laborers, researchers, artists, and others. New efforts by some of the digital archiving projects discussed below also reveal the

diversity of both the user base intended to use the tools, software, and workflows being developed by the digital preservation community, as well as that of the new voices that digital archivists are striving to include in archival collections. Where archives of the past have often represented those with a certain level of power, recent projects have been working to establish the visibility of those accidentally and purposely erased from previous archival memory.

3.2.1 Documenting the Now. Documenting the Now (DocNow)¹, a Mellon-funded project that began in 2016, aims to collect and preserve social media associated with historic events. The project is unique in its focus on representing the needs and desires of content creators whose content it aims to preserve, delving deep into the new ethics of digital archiving. DocNow has specifically addressed issues such as the ethics of collecting data in the first place if it puts content creators at risk of harm or imprisonment. The project’s work is also unique in that much of the discussions are openly documented and available on platforms such as DocNow’s public Slack channel, which will allow the decisions made by the project’s team and collaborators to gather wider input from the community, and to have a strong and lasting impact of future ethics embedded in tools and practices for digital archiving.

3.2.2 Mukurtu. Mukurtu, an online digital content management system, has a stated goal “to empower communities to manage, share, preserve, and exchange their digital heritage in culturally relevant and ethically-minded ways” [20]. The project has contributed to the accessibility of technical solutions for digital archiving that ensure alignment with cultural practices. Began as a collaboration between researchers and members of the Warumungu community, Mukurtu provides access controls that align with the needs of cultural protocols, as well as traditional knowledge labels to prevent misinformation or appropriation of cultural objects and memory [21].

3 THE NEED FOR EMPATHY IN TECHNICAL DOCUMENTATION

3.1 Defining Technical Documentation

Our use of the term technical documentation covers that of tools, software, and workflows. Based on the Open Preservation Foundation’s (OPF) software maturity model, we define this type of documentation as including “source code, comments, technical documentation, installation manuals, user documentation” [23].

The concept of what is good in terms of documentation varies depending on the use cases and context in which that documentation is created [3]. In laying out a method for assuring the quality of documentation, Knodel and Naab [22], define content and representation as the two main elements for review. Good documentation knows its audience and purpose in terms of content, and has high levels of consistency, understandability,

¹ www.docnow.io

completeness, traceability across documents, and extensibility. The failure of documentation, on the other hand, is complex, but there are two main issues at play in the context of digital archiving. First, in archives, we often erase the impact of our work by not providing public documentation of our regular practices and ongoing work [1]. Second, in the context of digital work, Harihareswara [14] contends that the larger technology field “systematically undervalues the jobs and roles that require empathy and has deeply gendered associations with hospitality and empathy.” In digital archiving, then, it can be argued that we are prone to downplaying both the human element of our own work and the humanity of our users.

3.2 Empathy

In order to provide tools, software, workflows, and organizational solutions that actually work for people undertaking digital archiving tasks, reframing technical documentation in terms of empathy is paramount. While the term is somewhat diffuse in its meanings, here we utilize Coplan’s definition, which is “a complex imaginative process in which an observer simulates another person’s situated psychological states while maintaining clear self-other differentiation” [7, p. 5]. Additionally, we attempt to apply concepts from Kouprie and Visser [18], who demonstrate how a framework of empathy can be constructed as a pragmatic process by using the phases of “discovery”, “immersion”, “connection”, and “detachment” to enhance the empathy of designers for users; with the ultimate goal of improving the use and adoption of tools by others who are unlike themselves.

Focusing on empathy in this way requires imagining the needs of another person whose goals and lived experience are very different from one’s own, while still maintaining critical distance. This is key, as the differences between stakeholders using technical documentation within digital archiving are what will begin to push the field into a more comprehensively human direction. As Jules [16], aptly pointed out in the National Digital Stewardship Alliance (NDSA) Digital Preservation 2016 keynote, we must embrace the “unbearable whiteness” of our field, going beyond just race to address underlying issues of overemphasis on “professionalism” and “standards and technical know how.”

4 BUILDING BETTER DOCUMENTATION

There is an opportunity for those working in digital archives to take an approach to technical documentation that allows the field to move closer toward the inclusion of a wide community of practitioners of varying skill sets and experiences. Additionally, it is an opportunity to contribute to a more thorough understanding of our own developing practices, standards, community needs and sustainability. It is also important to acknowledge the ongoing discussion of adoption and technical documentation related to the OSS and library community [6, 9, 12, 26], which will undoubtedly continue to help shape the discussion of digital preservation community documentation at large.

4.1 Empathic Community Guidelines

In codifying community technical documentation guidelines based on empathy, we seek to provide initial steps that reflect both the

need for improved documentation related to the adoption, maintenance and use of software and technical resources, as well as the lowering of barriers to find ways to better connect with the myriad users and preservation communities that create, maintain, teach and use the various systems and tools that further our work. These suggestions serve as a basis to think more comprehensively about everything from how-to documents for users to educational or technical workflow documents.

4.1.1 Establish a Method to Receive and Integrate Feedback, Iterations, and Updates. As Dowding et al. [10] argued, establishing users to verify tool, software, and workflow use cases is key, and we argue here that providing a mechanism for receiving ongoing feedback will add essential value. This concept involves some of the aspects of Kouprie and Visser’s [18] first two phases of “discovery” and “immersion”; user feedback and study is imperative for empathic design. If a project or tool is lucky enough to elicit user feedback and comments, this input should be used wisely. Organizations could take time each month to cull questions and answers from participants within a Google Group, listserv, Slack channel or internal meetings. This information could then be used to build out documentation and identify gaps in the information that is provided to the greater community. This interactive, iterative community approach has been shown to positively impact technical documentation [8].

The BitCurator community provides a robust example of how this type of approach could benefit both users and maintainers of the software. This community sees detailed public questions in their dedicated Google Group based on practical, user case studies on a regular basis that are answered by staff or other users with a good deal of care and technical expertise. BitCurator also provides a detailed “Quick Start” guide that accompanies their software that attempts to address the myriad issues than can arise with installation of the software [4]. This guide demonstrates an iterative approach to documenting, as a new version is released along with new versions of the software, and a wiki with documentation is also maintained.

However, though the community is very active, questions and answers with useful information are eventually buried within years of Google Group pages, and are not always included in the newest “Quick Start” guide or other technical documentation for the software. These questions range from troubleshooting hardware to detailed error reports and use cases. This not only raises questions about the sustainability of this type of community documentation approach, but can leave users - especially those new to digital forensics or digital archiving - to sift through years of questions to unearth a relevant solution. This approach can also create redundancies in answers from BitCurator project staff, and sometimes leaves questions unanswered. To address these issues, and to make use of the invested user and staff community, a regular, automated or scripted solution for a project as large as BitCurator could be used to extract pertinent or new questions and answers for use in iterative technical documentation such as the “Quick Start” guide. This way, documentation could better refine common issues gathered and observed from the larger community, define use cases, and identify patterns or gaps that technical documentation needs to address. For smaller communities or projects, this could be achieved manually with similar results.

4.1.2 *Better Explain Errors and Provide Examples.* Many of the following suggestions relate to the “connection” phase of the empathic process as defined by Kouprie and Visser [18], in that those writing technical documentation may connect with users by recalling their own experiences first learning code, or a new tool, and then using those experiences to aid documentation design. Empathy for users, therefore, may be translated pragmatically in technical documentation to include:

Providing clear explanations of common errors that those who are not yet well versed in a tool or project may encounter, and what these errors mean. This specific approach means that common user errors can be learned from and decoded via technical documentation, allowing for greater understanding of underlying concepts, or the structure of a tool or program. An example of this type of empathic approach can be found in the documentation and exercises in *Learning Python the Hard Way*, an online coursebook for learning the Python programming language. The course details common questions and errors from new users with each exercise, along with why and how these errors might occur [24]. Additional related examples include fairly minimal efforts that nevertheless can translate into highly useful concepts for diverse user bases, such as clearer and more verbose error codes that can allow users to seek distinct further information [5].

Providing pragmatic, specific examples that reflect a variety of skill sets and functions related to a concept, tool or software. Providing examples can benefit both those developing tools or software by serving as use cases or user stories, as well as those wishing to learn the tool itself. An example of this approach can be found in technical documentation for some application programming interfaces (APIs), such as the Digital Public Library of America², where samples and guides provide step-by-step instructions for the most basic functionality of the API, along with notes about why and how specific elements were used.

4.1.3 *Lower All Possible Barriers.* “It is open but not available.” Davidson and Casden [9] express this sentiment in recognition of the wide variety of resources and technical skill levels that are the reality for most cultural heritage organizations outside of large research institutions, and how this reality affects the adoption and use of open source software and tools. The authors argue that software cannot truly provide utility to a large and diverse community until gaps are identified and programmatically addressed to further lower the barriers of adoption. Perhaps most importantly, they pinpoint that “[c]ommunity members who would help address many other types of users (serving many other types of institutional users) never have the opportunity to participate” when barriers to do so continue to be too high, and therefore the more inclusive community desired remains unrealized. Though technical documentation cannot solve all related barrier issues, parallel supporting arguments can be made in the form of improving the type and scope of documentation for these and other projects that reflect the needs and resources of a technologically-diverse digital archives community. In addition to intuitive programmatic changes and thorough testing with a variety of users, suggestions for documentation that also aids the continued lowering of implementation barriers include:

Moving from a lack of specificity to extreme detail. Technical documentation should show users more granular detail for everything from creating exact command line strings to visual examples that demonstrate settings for common operating systems. While this approach may undeniably take more time initially, this specificity will help developers, maintainers, and users, as it introduces a way of creating standard, documented steps that can be more easily reviewed if something goes wrong. This approach can then enable users to feel more comfortable investigating issues independently in the future or assisting others in the community with use. The web archiving subscription service Archive-It and BitCurator community documentation are good examples of this type of approach, and understandably so; their contributing feedback communities of users range widely, and both have worked through several iterations of development and documentation.

Defining terms clearly and avoid elusive language or jargon. The word “instance” provides an excellent example of a term used with regularity between developer, library, and various user worlds that can have a variety of meanings in different contexts. For a developer, this term may be used more frequently to refer to an object belonging to a certain class, whereas users may be more likely to use “instance” as a singular example. A reference guide or glossary for terms is a familiar concept, and would be a helpful addition to consider. Avoiding jargon within technical documentation is also imperative, as specialized language emphasizes differences and prioritizes the professionalization that Jules [16] argues against.

4.1.4 *Establishing Better Timing for Documentation.* In a study examining open source developer documentation, Dagenais and Robillard found that when developers were encouraged to update their documentation with each code change, the practice “...led to a form of embarrassment-driven development, which in turn led to an improvement in the code quality” [8, p.127]. Kouprie and Visser also suggest that, “a process of empathy in design practice requires a structured investment of time” [18, p. 447]. This investment of time and reflection informs the final emphatic phase of “detachment”; when the designer steps back out of the user world with an increased understanding that is then used to drive new design insight. Both concepts compel the idea that early, iterative documentation can help improve the work being done whether it is related to community code or use cases. Often, technical documentation is something that happens after all other work is done, and it is difficult for most to provide imperfect or “unfinished” work, even for internal use, much less for public consumption [2, 26]. However, it has been shown that following this type of strategy can lead to developers who, “...document their changes as quickly as possible after realizing that they often improved their code while documenting” [8, p. 130].

There is opportunity in the digital preservation community, that writing empathic technical documentation early and often will strengthen the overall effectiveness of approach. This means that if draft documentation can also be considered a means to not just detail strategy or craft code, but also to structure empathic phases by integrating user feedback, recording errors and providing detailed examples, and critically reviewing approaches to an issue, it has an improved chance to more fully serve the greater community. This draft technical documentation need not be

² <https://dp.la/info/developers/codex/>

perfect; the point is to get the basic, but tested, concepts down and allow them to be reviewed [2, 13].

4.1.5 *Improve Community Availability*. At the iPRES 2015 OSS4Pres workshop, one of the most important issues highlighted by participants related to the successful adoption and use of open source tools was the need for a publically-available, centralized technical documentation source that related to the more granular, problem-solving activities and issues that can arise when integrating OSS into digital preservation workflows. Requests for “end-to-end workflows” and “case studies” prevailed, alongside requests that documentation need also be kept up to date and versioned [10, 12]. Building on these recommendations, particularly that a public, centralized hub need not be built from scratch, but could instead be modeled after existing wikis such as the Community Owned digital Preservation Tool Registry (COPTR)³, a community hub might also include efforts to integrate with both international and national digital archiving and preservation initiatives such as the NDSA, the OPF, or the Digital Preservation Coalition. This approach would increase communication channels and encourage a wider variety of practitioners and communities to write, search and provide feedback for technical documentation. However, it is equally important that access and submissions to such a hub also stem from smaller or burgeoning sources of documentation such as the Society for American Archivists Electronic Records blog⁴ or community digital archiving efforts, as these sources will further enrich it. Providing examples of useful technical or instructional documentation, as well as calls for specific documentation types might also stem from the hub to fill gaps as defined by the user community.

5 Conclusion

These community technical documentation guidelines represent the beginning of a broad structure based on empathy that can be built around the various digital archiving communities. Further work remains to be done, particularly around the need for public examples and testing of diverse approaches to empathic documentation, as well as further codification of building an empathy framework. However, approaching documentation in this way is necessary to improve sustainability, innovate practices, and remove technological barriers to better serve our growing community.

REFERENCES

- [1] Hillel Arnold. 2016. Critical Work: Archivists as Maintainers. (August 2016). Retrieved February 7, 2017 from <http://hillelarnold.com/blog/2016/08/critical-work/>
- [2] Dale Askey. 2008. We Love Our Open Source Software. No, You Can't Have Our Code. *Code4Lib Journal*, 5 (Dec. 2008). Retrieved February 15, 2017 from <http://journal.code4lib.org/articles/527>
- [3] Joachim Bayer and Dirk Muthig. 2006. A view-based approach for improving software documentation practices. Engineering of Computer Based Systems, 2006. Proceedings of ECBS 2006: 13th Annual IEEE International Symposium and Workshop. March 27 - 30, 2006: Potsdam, Germany. Retrieved March 6, 2017 from <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1607376>
- [4] BitCurator Consortium. 2016. BitCurator Quick Start Guide (November 2016). Retrieved March 3, 2017 from <https://wiki.bitcurator.net/downloads/BitCurator-Quickstart.pdf>
- [5] Rich Bowen. 2015. RTFM? How to write a manual worth reading. (May 2015). Retrieved February 18, 2017 from <https://opensource.com/business/15/5/write-better-docs>
- [6] Dan Chudnov. 1999. Open Source Library Systems: Getting Started. Retrieved February 6, 2017 from <http://www.oss4lib.org/readings/oss4lib-getting-started.php>
- [7] Amy Coplan. 2014. *Empathy: philosophical and psychological perspectives*. Oxford University Press, Oxford, UK.
- [8] Barthélémy Dagenais and Martin P. Robillard. 2010. Creating and Evolving Developer Documentation: Understanding the Decisions of Open Source Contributors. In *Proceedings of the 18th ACM SIGSOFT international symposium on Foundations of Software Engineering (FSE '10)*. ACM Press, New York, NY, 127-136. DOI:<http://dx.doi.org/10.1145/1882291.1882312>
- [9] Bret Davidson and Jason Casden. 2016. Beyond Open Source: Evaluating the Community Availability of Software. *Code4Lib Journal*, 31 (Jan. 2016). Retrieved February 10 from <http://journal.code4lib.org/articles/11148>
- [10] Heidi Dowding, David Wilcox, Johan Mören, Phillip Tømmerholt, Johan van der Knijff, Angela Dappert, and Sarah Romkey. OSS4Pres 2.0: Building Bridges and Filling Gaps. (October 2016). Retrieved March 7, 2017 from https://docs.google.com/document/d/1R_Tyrbc4Y90-cFLlKtZm_QURYgBLyRHcjDrSjrSzk0/edit?usp=sharing
- [11] Emily Drabinski. 2016. Valuing Professionalism: Discourse as Professional Practice. *Library Trends* 64, 3 (2016), 604-614. Retrieved February 8, 2017 from <http://muse.jhu.edu/article/613926/pdf>
- [12] Marty Gengenbach, Shira Peltzman, Sam Meister, Blake Graham, Dorothy Waugh, Jessica Moran, Julie Seifert, Heidi Dowding, and Janet Carleton. 2016. OSS4Eva: Using Open-Source Tools to Fulfill Digital Preservation Requirements. *Code4Lib Journal* 34 (Oct. 2016). Retrieved February 7, 2017 from <http://journal.code4lib.org/articles/11940>
- [13] Anne Gentle. 2012. *Conversation and Community: The Social Web for Documentation* (2nd. ed.). XML Press, Laguna Hills, CA
- [14] Sumana Harihareswara. 2015. User Experience is a Social Justice Issue. *Code4Lib Journal*, 28 (April 2015). Retrieved February 7, 2017 from <http://journal.code4lib.org/articles/10482>
- [15] Steven J. Jackson. 2013. *Rethinking Repair*. MIT Press, Cambridge, MA.
- [16] Bergis Jules. 2016. Confronting Our Failure of Care Around the Legacies of Marginalized People in the Archives. (November 2016). Retrieved March 7, 2017 from https://medium.com/on-archivy/confronting-our-failure-of-care-around-the-legacies-of-marginalized-people-in-the-archives-dc4180397280#_gw84zs68t
- [17] Beth E. Kolko, Lisa Nakamura, and Gilbert B. Rodman (Eds). 2000. *Race in Cyberspace*. Routledge: London
- [18] Merlijn Kouprie and Froukje Sleeswijk Visser. 2009. A Framework for Empathy in Design: Stepping Into and Out of the Users's Life. *Journal of Engineering Design* 20, 5 (Oct. 2009), 437-448.
- [19] Bruno Latour. 2012. Love Your Monsters: Why We Must Care for Our Technologies As We Do Our Children. *The Breakthrough* (Winter 2012). Retrieved February 9, 2017 from <http://thebreakthrough.org/index.php/journal/past-issues/issue-2/love-your-monsters>
- [20] Mukurtu.n.d. About. Retrieved March 7, 2017 from <http://mukurtu.org/about/>
- [21] Mukurtu.n.d. Learn About Mukurtu CMS. Retrieved March 7, 2017 from <http://mukurtu.org/learn/>
- [22] Jens Nodel & Matthias Naab. 2016. Pragmatic Evaluation of Software Architectures. Springer, Cham, Switzerland. DOI: 10.1007/978-3-319-34177-4
- [23] Open Preservation Foundation, n.d. Software Maturity. Retrieved February 10, 2017 from <http://openpreservation.org/technology/principles/software-maturity/>
- [24] Zed A. Shaw. 2016. Learn Python the hard way. (2016). Retrieved March 7, 2017 from <https://learnpythonthehardway.org/book>
- [25] Madeline Sheldon. 2013. Analysis of Current Digital Preservation Policies: Archives, Libraries, and Museums. (July 2013). Retrieved March 8, 2017 from <http://www.digitalpreservation.gov/documents/Analysis%20of%20Current%20Digital%20Preservation%20Policies.pdf?locl=blogsig>
- [26] Curtis Thacker and Charles Knutson. 2015. Barriers to Initiation of Open Source Software Projects in Libraries. *Code4Lib Journal*, 29 (Jul. 2015). Retrieved February 10 from <http://journal.code4lib.org/articles/10665>
- [27] Paul Wheatley. 2013. Spruce Mashup Manifesto. (April 2013). Retrieved March 7, 2017 from <http://wiki.opf-labs.org/display/SPR/The+SPRUCE+Mashup+Manifesto>
- [28] Matt Zumwalt, Richard Green, and Michael J. Giarlo. 2014. Hydra Community Principles. (January 2014). Retrieved March 7, 2017 from <https://wiki.duraspace.org/display/hydra/Hydra+Community+Principles>

³ <http://coptr.digipres.org/>

⁴ <https://saaers.wordpress.com/>