Preserving Digital Scholarship in Perseids: An Exploration

Fernando Rios
Data Managment Services, The Sheridan Libraries
Johns Hopkins University
rios@jhu.edu

Bridget Almas
Perseids Project, Tufts University
bridget.almas@tufts.edu

October 10, 2016

1 Introduction

Software is an important part of many kinds of scholarship. However, it is often an invisible part of the knowledge generation process. As a result, software's lack of visibility within the scholarly record inhibits the understanding and future use of the scholarship which is dependent on it. One way to mitigate that outcome is to preserve not only the final result but also the actual platform, services and tools upon which it depends.

In order to guide preservation of these platforms and services, Data Management Services at Johns Hopkins University is exploring several aspects of software preservation, one of which is investigating how preservation needs can be determined for particular projects such as Perseids. The Perseids Project at Tufts University is a web-based platform that is being used to produce new forms of digital scholarship for the humanities. Consequently, examining how this scholarship might be preserved by preserving the underlying software is of practical importance.

One of the outputs of the Perseids Project has been a series of prototypes of new forms of data-driven publications and digital editions. The data for these online publication prototypes have been produced through the use of a variety of software tools and services that combine dynamically provided data through orchestrated calls to web services. The software tools and underlying services have gone through several iterations of development throughout the lifetime of the project and publications have been produced at different stages of that development. This scenario poses a series of interesting challenges for preservation of these digital publications, the underlying data, and the tools and services that are intrinsic to them.

 DOI: 10.5281/zenodo.159569

2 Objectives

This exploratory project had two objectives. The first was to give structure to thinking about how the data-driven publications and digital editions enabled by Perseids could be preserved. The primary concerns were what should be considered in determining how to adequately capture the collection of services and tools that comprise Perseids? Should the entire collection even be captured? The second objective was to develop and trial a set of questions, presented in the form of a questionnaire, that could be used to elicit information to help address the first objective.

3 Methods

The Perseids platform and the publications produced on it rely upon complex pieces of software with many moving parts. In order to begin addressing the question of how such a platform and its publications might be preserved, we had several informal discussions of what the major parts of Perseids were, along with general approaches to preservation and the associated challenges. We focused our investigation on a prototype digital publication that was developed on an early version of the platform and that used versions of the annotation tools and services from Perseids which have since been significantly updated or replaced since the prototype was first produced.

In order to understand how we might proceed with a potential software preservation activity, we decided it was important to answer three questions. First, we agreed it was important to have clarity on what the purpose of preservation is and who would benefit. Second, we determined that understanding what the pieces of the software are and how they are interdependent was critical. Third, we decided that being clear on what the costs versus benefits of preserving the Perseids software were, in relation to alternative approaches (e.g., website capture), was the most important question to address, from a practical perspective.

To structure the information, we used two questionnaires developed by Fernando for the purpose of providing consulting services for software archiving by the Data Management Services group at Johns Hopkins University. The first questionnaire asked very general questions in order to appraise the state of the software and gauge any potential gaps which may hinder its preservation and future reuse. Questions included asking the purpose of the Perseids project, its intended audience, the state of user- and developer-oriented documentation, general information about external software dependencies, and questions meant to gauge the general attitude with respect to software preservation and credit. After Bridget completed the questionnaire, we decided to move forward with determining what might need to be done in order to preserve the scholarship that the target use case represents (i.e, the prototype digital publication) and how it might be carried out.

To do this, a second, more focused questionnaire was developed (by Fernando, using feedback given by Bridget on the first questionnaire) in order to get us thinking about the specifics of preservation, including most importantly, the why. Figure 1 shows the sequence in which

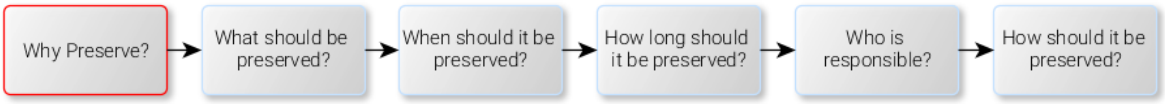

Figure 1: The sequence in which different aspects of preservation were addressed. The "Why" is the most important aspect.

different aspects of preservation were addressed. The questions are loosely grouped by what kind of information they capture: why, what, when, how long, who, and how.

Although the questionnaires are still undergoing refinement and are not (yet) publicly distributed, a brief description of the information captured by the questionnaire we used is shown in table 1.

4 Lessons Learned

We learned, first, the importance of sorting through the "why" and "what" to identify those pieces of software which warrant preservation activity and to define exactly what approach to take to preservation. Having the framework of the questionnaire to guide our thinking about those issues helped to focus what felt at the beginning like a daunting task.

Bridget entered into the discussions with Fernando with a pragmatic motivation: as development progresses on Perseids, having to support multiple earlier versions of services in order to support the prototype publications becomes increasingly unmanageable. We wanted to be able to retire the earlier service versions that these prototypes depend upon, but the cost versus benefit ratio for upgrading prototype code does not always allow for that. In considering the options for preserving a functioning version of a prototype, some of which themselves imply a fair amount of work (such as creating and preserving a Docker container image of all the supporting pieces), thinking about the the true purpose for preservation helped to put the problem in perspective and also to identify gaps in our planning and preservation capabilities.

While each of the suggested motivations from Fernando's questionnaire could be considered to be an ideal to which to aspire in general, when held up against the specific software, they didn't all make practical sense. For example, while in theory, reproducibility of the exact display of the annotations and textual data from our target use case seemed desirable, we had to ask if that was essential for preserving and reproducing the scholarship. The answer to that might have been yes if we had amassed large quantities of data for the use case, and expanded it beyond the initial prototype. But as we have not yet been able to do that, and the tools and services in question have since evolved, the small dataset we have accumulated for our publication would be better reproduced and expanded via newer tools. With this consideration in mind, it seems the remaining value of the prototype code would be as a demonstration of a methodology for annotation and a proposed service-based infrastructure to support that methodology. The code itself is of less consequence than a documentation of the ideas and dependencies would be.

Why	Questions in this part revolved around really thinking about the true purpose of preserving software (e.g., enabling reproducibility, reuse, or continued access to scholarship) as well as the intended audience.
What	This section attempted to help us think through two things. First, at what level of granularity should the software be described and preserved in order to fulfil the preservation goal? This is important because different goals may require different levels of granularity in the description of the software. An example of a highly granular description is describing not only the software as a whole but also describing and documenting the individual pieces that comprise it as well as their interrelationships. Once an appropriate level of granularity was determined, a series of questions elicited information on those pieces.
When	This section attempts to determine what an appropriate time to preserve software is. For normal grant-funded projects, this will likely be at the end of the project or at the time of publication.
How Long	This part simply asks at least how long should the software be preserved. It is a simple question with a potentially difficult answer. Ideally, the answer is 'a long time' but the longer the time span, the more effort must be made to ensuring the software remains not only accessible but also usable. Therefore, it is important to come up with a number based on available resources.
Who	This section is meant to determine who is responsible for not only the software but also who bears responsibility for archiving it, making it citable, assigning unique identifiers etc. This section also is meant to help in identifying a suitable archive where it may be stored.
How	This section elicits what approach seems reasonable to preserve the software (e.g., by archiving the source code as-is, using virtualization or emulation technology, or by continued development). In addition, this section determines the kind of documentation that will be included and how it will be attached to the software (e.g., readme file, wiki, structured metadata). Although not part of the questionnaire, the Pathways of Research Software Preservation [1] gives an overview of how different parts of research software might be preserved and how different approaches are related.

Table 1: The information we wish to capture with the questionnaire.

This problem is discussed in the context of scientific workflows in "Techniques for Preserving Scientific Software Executions: Preserve the Mess or Encourage Cleanliness?" [2]. The authors found that preservation of distributed environments is still very much an open question and they suggest various approaches. In our case, a Docker image would allow an end-user to see the prototype functioning as it did when published but would provide little insight into the methodology or infrastructure. As we don't intend to reproduce this environment exactly, we might consider just preserving the "working principle", providing a description of the setup, using a controlled vocabulary.

It also became clear, in reviewing the questionnaire, that simply having code in GitHub or other open source versioning repository is not sufficient. All code we write is available in the project's GitHub repository. However, because of the complex history and dependencies of open source software development, what exists in the repository represents, in many cases, only the tip of the iceberg. In addition, the GitHub repository, as it currently stands, doesn't present a true picture of all the people who contributed intellectually to these efforts, because the code is just one piece of the puzzle. As discussed in Matthew Turk's excellent post, "The Royal 'We' in Scientific Software Development" [3], we need to do a much better job of recording and crediting this intellectual work. Further, we need to be cognizant of the need to to this as the work takes place. An ontology such as TaDiRAH [4] would be worth considering here.

The "who" section of the questionnaire also raised some interesting questions. Where does the responsibility for preservation lie, between the software developer and the scholar? Many of the use cases we work on in Perseids are not explicitly funded projects in and of themselves. Our approach has been to try to do as much as possible to serve as many real scholarly workflow needs as possible. This has provided the opportunity for us to explore various questions around what humanities infrastructure needs to support, while hopefully still also providing real value to our users. At the same time, we have learned that without adequate planning for governance and sustainability, things can and do fall through the cracks. Prototype code which we have developed, such as for the use case we examined here, does not always have a clear owner. For future projects of this nature, we need to take the time at the beginning to ask ourselves these questions about who will take ownership and responsibility for ensuring the preservation in order to eliminate this ambiguity.

5 Conclusions and Next Steps

Although data preservation and sharing has received much attention from funders, publishers, libraries and research communities in the past 10 years or so, methods, tools, and best practices for preserving and curating the software associated have not been as fully developed. The evaluation of the Perseids project served to contextualize some of the ideas and workflows around capturing information to enable the archiving of research software that are being developed in the Data Management Services group at Johns Hopkins University. From the Perseids Project's perspective, the iterative approach we took gave us a clearer idea of the unique requirements and challenges of preserving the scholarship embedded in this software.

We learned that while having an ideal to shoot for is good, the ideal isn't always the best or

most practical approach. We have, however, identified some concrete next steps we can take to move closer to where we would like to be with preservation of the platform components and outputs.

First, we will explore ontologies and approaches for describing the distributed infrastructure we have envisioned for our publications. We have started with an analysis of the OntoSoft Ontology [5], although at first glance, it does not seem possible to express with it all the layers of intent and dependencies in our environment. We also intend to explore the Linked Resource Model ontology [6] developed by the Pericles-EU project for this purpose.

In order to preserve the end-user experience of our publications, we expect to use Webrecorder.io service to create web archive snapshots of their current state. This will allow us to preserve the visual representation of the scholarly output without a dependency upon the software behind it being available in perpetuity.

Finally, we hope to do a better job planning for the sustainability and stewardship of future undertakings on the platform from the outset, including identifying all participants and the nature of their contributions.

References

[1] F. Rios, "The pathways of research software preservation: An educational and planning resource for service development," *D-Lib Magazine*, vol. 22, no. 7/8, 2016. DOI: 10.1045/july2016-rios (cit. on p. 4).

[2] D. Thain, P. Ivie, and H. Meng, "Techniques for preserving scientific software executions: Preserve the mess or encourage cleanliness?" In *Proceedings of the 12th International Conference on Digital Preservation (iPRES)*, Nov. 2015. [Online]. Available: http://dx.doi.org/doi:10.7274/R0CZ353M (visited on Nov. 9, 2015) (cit. on p. 5).

[3] M. Turk, *The royal "we" in scientific software development*, Accessed September 15, 2016, Jul. 2016. [Online]. Available: https://medium.com/@matthewturk/the-royal-we-in-scientific-software-development-9deea495b3b6 (cit. on p. 5).

[4] C. Schöch, J. Perkins, and L. Borek, *TaDiRAH: Release version 0.5.3*, Oct. 2015. DOI: 10.5281/zenodo.32492. [Online]. Available: https://doi.org/10.5281/zenodo.32492 (cit. on p. 5).

[5] Y. Gil, V. Ratnakar, and D. Garijo, "OntoSoft: Capturing scientific software metadata," en, in *Proceedings of the Eighth ACM International Conference on Knowledge Capture, Palisades, NY*, ACM Press, 2015, pp. 1–4, ISBN: 978-1-4503-3849-3. DOI: 10.1145/2815833.2816955. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2815833.2816955 (visited on Apr. 12, 2016) (cit. on p. 6).

[6] J.-Y. Vion-Dury, N. Lagos, E. Kontopoulos, M. Riga, P. Mitzias, G. Meditskos, S. Waddington, P. Laurenson, and I. Kompatsiaris, "Designing for inconsistency – the dependency-based PERICLES approach," in *New Trends in Databases and Information Systems*, ser. Communications in Computer and Information Science, T. Morzy, P. Valduriez, and L. Bellatreche, Eds., vol. 539, Springer International Publishing, 2015, pp. 458–467. DOI: 10.1007/978-3-319-23201-0_46 (cit. on p. 6).