

Sustaining Software Preservation Efforts Through Use and Communities of Practice

Fernando Rios
University of Arizona

Monique Lassere
University of Arizona

Judd Ethan Ruggill
University of Arizona

Ken S. McAllister
University of Arizona

Abstract

The brief history of software preservation efforts illustrates one phenomenon repeatedly: not unlike spinning a plate on a broomstick, it is easy to get things going, but difficult to keep them stable and moving. Within the context of video games and other forms of cultural heritage (where most software preservation efforts have lately been focused), this challenge has several characteristic expressions, some technical (e.g., the difficulty of capturing and emulating protected binary files and proprietary hardware), and some legal (e.g., providing archive users with access to preserved games in the face of variously threatening end user licence agreements). In other contexts, such as the preservation of research-oriented software, there can be additional challenges, including insufficient awareness and training on unusual (or even unique) software and hardware systems, as well as a general lack of incentive for preserving “old data.” We believe that in both contexts, there is a relatively accessible solution: the fostering of communities of practice. Such groups are designed to bring together like-minded individuals to discuss, share, teach, implement, and sustain special interest groups—in this case, groups engaged in software preservation.

In this paper, we present two approaches to sustaining software preservation efforts via community. The first is emphasizing within the community of practice the importance of “preservation through use,” that is, preserving software heritage by staying familiar with how it feels, looks, and works. The second approach for sustaining software preservation efforts is to convene direct and adjacent expertise to facilitate knowledge exchange across domain barriers to help address local needs; a sufficiently diverse community will be able (and eager) to provide these types of expertise on an as-needed basis. We outline here these sustainability mechanisms, then show how the networking of various domain-specific preservation efforts can be converted into a cohesive, transdisciplinary, and highly collaborative software preservation team.

Submitted 15 December 2019

Correspondence should be addressed to Fernando Rios, Email: frrios@email.arizona.edu

The 15th International Digital Curation Conference takes place on 17 - 20 February 2020 in Dublin, Ireland
URL: <http://www.dcc.ac.uk/events/idcc20>

Copyright rests with the authors. This work is released under a Creative Commons Attribution 4.0 International Licence. For details please see
<http://creativecommons.org/licenses/by/4.0/>



Introduction

The importance of software preservation and its associated challenges and potential solutions have been broadly framed and discussed in the literature primarily in two contexts: (1) software that has cultural and historical importance (e.g., Michael Shrayner's Electric Pencil, the first word processing application for the home computer market), and (2) software created for computationally-oriented research (e.g., TensorFlow, Google's open-source machine learning library). Since the mid-1980s, scholarly literature about the former has mostly addressed the importance and challenges of software preservation (e.g., Swade, 1993; Swade, 1995; Rothenberg, 1999; Shustek, 2006; Lowood, 2013); much of this work has focused on the preservation of video game software (Monnens et al., 2009; McDonough et al., 2010; Kaltman et al., 2014); and the less formal work emerging from video game archives such as Stanford's Cabrinety Collection, the Internet Archive's Internet Arcade, and the University of Michigan's Computer and Video Game Archive. Scholarship about the preservation challenges surrounding research software broadly mirror those from the cultural heritage domain (Chue Hong et al., 2010; Matthews et al., 2010; Matthews et al., 2015; Chassanoff & Altman, 2019), but differs in its attention to issues of research reproducibility and re-use (Crick et al., 2014; Thain et al., 2015; Peer & Wykstra, 2016; Brown, 2017; Lynch, 2017; Rios et al., 2017; Baillieul et al., 2018).

Despite nearly four decades of the wide ranging discussion about software preservation, one theme remains consistent: the human element. Whether explicitly or implicitly, the limits and foibles of human beings routinely appear in software preservation scholarship, often leading archivists to develop solutions designed not only to make projects more efficacious, but also faster and less susceptible to human error. In the context of research software, for instance, open source software communities have garnered considerable attention recently as a workload and information-sharing solution, particularly for preservation workflows that include the labor of ingesting material, generally considered to be the most expensive part of digital preservation (Chue Hong et al., 2010; Hees, 2017; Rosenthal, 2018).

In this paper, we extend these discussions into the realm of sustainability, that is, once an institution has been convinced of software preservation's import, and an effective software preservation workflow has been developed, how are such accomplishments to be supported, managed, and grown over the lifetime of a given collection? We offer two tentative answers to these questions below. The first we characterize as "preservation through use," a model that has been used to successfully sustain the preservation efforts at the Learning Games Initiative Research Archive (LGIRA)—one of the world's largest research archives of videogames and their paratexts—for two decades. The second answer reorients the project of preservation such that people, rather than property, serve as the cynosure of all efforts. Such "communities of practice" both surface and sustain conversations around local software preservation challenges, and have proven quite effective for linking a highly diverse group of University of Arizona faculty, staff, and students in their efforts to preserve a wide range of software applications and their respective data files. Both preservation through use and communities of practice, we propose, are two sides of the same coin, linked by their structured and intentional approach to software preservation, their entrepreneurial symbiosis, and their commitment to broad-based knowledge sharing.

Preservation through Use and Communities of Practice

As difficult as it can be to build institutional interest in software preservation, sustaining engagement in this work often proves to be even more challenging. Below, we briefly outline two approaches that we have found effective for sustaining such engagement: “preservation through use” and the fostering of a local community of practice.

Preservation through Use

In short, “preservation-through-use” refers to the understanding that the act of preservation extends beyond the physical preservation of objects and further towards human capacity for memory-making. Memory-making prioritizes and leverages persistence of sensory-based, affective representations of the artifact and its uses through human interaction. All archives practice preservation through use to some degree. Some archives, for instance, allow the “use” of their objects only by a select and credentialed few. Consider rare book collections, for example, where the artifacts are largely kept away from curious eyes outside of specific, vetted research requests, and interaction with these artifacts is only permitted under employee observation. Other archives—like LGIRA—are far more accessible, encouraging object use and interaction by almost anyone with a probing question (Ruggill and McAllister, 2011). Preservation through use is orthogonal to preservation in the traditional sense, i.e., keeping physical objects in pristine condition and available only to a select few. It is thus meant as a complementary approach to traditional preservation, one that appreciates material preservation for future inquirers even as it focuses on asomatous preservation for inquirers here and now. For archives like LGIRA, preservation through use also makes good sense: without human interaction, “video games” are actually just packaged storage media. A game does not exist until a player powers up the preserved and articulated software and hardware. Preserving video games thus not only benefits from use—it requires it. Moreover, the deleterious physical effects of preservation through use on objects (e.g., wear and tear) has the preservative benefit of necessitating increased knowledge about objects’ inner workings. Repair and maintenance are thus an integral part not only of the preservation of an archive’s objects, but also of the community that literally keeps it working. Drawing on Matthew Kirschenbaum’s interpretation of Kenneth Thibodeau’s model of a digital object, what we aim to preserve is an interactive, commutable representation of the artifact as physical, logical, and conceptual objects (Kirschenbaum, 2008). Within these classes of distinction, a high degree of flexibility exists due to the modular nature of digital objects. While preservation-through-use valorizes interaction with the original artifact, it may do so by applying discrete physical and logical resources that change over time or actualize the concept of the original artifact and its original functionality in alternative ways. Distributing access to the original artifact digitally then requires emulation, a preservation tactic which allows one device to simulate the behavior of another device. Because emulation allows us to account for a variety of processes by which an artifact’s functionality can be recreated that expend software and hardware resources differently, it exists as a shared, cross-disciplinary approach to software preservation.

Fostering Communities of Practice

Communities of practice are increasingly being recognized as critical to the sustainability of software preservation efforts. To underscore this point, the Software Heritage Archive, a project aiming to collect all publicly available source code, has specifically stressed the need for support from partners in industry, government, education, and elsewhere. It specifically includes “Community” as one of the four ingredients for success (Di Cosmo &

Zacchioli, 2017; Abramatic et al., 2018). To address the need for software preservation-specific communities that are broad in scope, the collective action organization known as the Software Preservation Network (SPN) has functioned as a nexus for professionals interested in building, connecting, and sustaining communities interested in advancing software preservation for both specific needs and the general welfare (Meyerson et al., 2017). To that end, SPN spearheaded an initiative to foster communities of practice around the use of emulation technologies for software preservation, mainly to document and share lessons learned with the rest of the software preservation community (“Fostering Community of Practice in Libraries, Archives and Museums,” n.d.).

In 2017, we saw an opportunity to explore how active but niche communities like the one supporting LGIRA could be linked with other local software preservation communities, and in so doing could strengthen the SPN generally. That same year, we represented one of six institutions accepted into the SPN’s “Fostering Communities of Practice” (FCoP) cohort, having proposed to seed a community around the development of emulation-based workflows for game preservation and beyond. The application—“Through Use and Emulation: Increasing Institutional Knowledge of Software Preservation with Computer Game Archiving,” proposed that software preservation is necessarily a transdisciplinary challenge and thus must be approached by multi-disciplinary, collaborative, and highly communicative teams.

Linking Preservation through Use with a Local Community of Practice

Initiating a local software preservation community of practice at the University of Arizona in cooperation with LGIRA was relatively easy given LGIRA’s established network of stakeholders, its desire for increased local awareness, and the opportunity to tap into local expertise about topics that, for technical and organizational reasons, had exceeded LGIRA’s capacities (e.g., video game emulation and virtualization). It was clear that there was also an opportunity for the local community to surface their own software preservation challenges among colleagues expert not just in video game preservation but also in fields such as high performance computing, data visualization, and digital forensics. Our initial strategy for launching this community comprised of three meetings, with time dedicated not only to presenting topics such as video game preservation through use but also to allowing participants to share their backgrounds and challenges.

The first meeting of the group, christened the University of Arizona Software Preservation Interest Group (UA-SPIG), was intended to introduce the objectives of the group, brainstorm about how members could both contribute and be helped, and generally make connections across our large campus. Among the inaugural visitors were data librarians, physicists, electrical engineers, language researchers, archivists, and historians. One set of advantages to having a video game archive at the center of this project is that it offers familiar subject matter from which to launch more general inquiries, it is wondrous in its variety, and it offers a potent testimony to the transdisciplinarity of the video game industry. These factors made it relatively easy to establish LGIRA as a hub of interest and a conversational focal point for discussions about the complexities of software preservation writ large.

A short discussion session at the end of the first meeting yielded a number of insights into how local concerns (e.g., research data trapped on old computers and lab instruments) might usefully intersect with software preservation. And even though the discussion had been seeded with video game examples, most participants seemed to find it easy to jump from games to topics further afield. Indeed, one of the most fruitful topics concerned the *educational potential* of software preservation. One participant observed

that software archives like LGIRA give budding game developers a chance to examine the play mechanics of old games. Another noted that such archives could make an array of software available to new users (e.g., those in different fields) who would otherwise find it technically or financially inaccessible. A third participant pointed out that comprehensive software preservation programs at schools would significantly reduce the difficulty of retaining easy access to students' digital portfolios (e.g., digital art and music assets for games, 3D models and scans, source code). We also learned in this first meeting that the participants had no shared software preservation vocabulary. As a result, concepts such as "software heritage" and "workflows for preservation" were opaque to some and alienating to others. Consequently, we developed a short "Software Preservation 101" module (available at osf.io/2wrqc) for the next meeting that allowed us to introduce, in an accessible, pan-disciplinary manner, the what, why, and how of preservation.

UA-SPIG's second meeting began with the Software Preservation 101 module, followed by a short sticky-note activity meant to elicit potential topics of interest and opportunities for collaboration. The group agreed that establishing a base level of software preservation knowledge would aid transdisciplinary collaboration and strengthen the learning community. In response, UA-SPIG is currently developing educational material based on the (mostly) relatable content of the Software Preservation 101 module.

To broaden the conversation beyond video games and illuminate additional challenges that could inform emulation-based workflows in LGIRA, we organized a third session, as part of Research Bazaar Arizona 2019. Research Bazaar is a worldwide festival promoting crosscutting conversations to help researchers "up-skill" in modern digital research tools and techniques (ResBaz Tucson, 2019). Attendees included some individuals with some familiarity of preservation of digital artifacts but most were researchers from other domains whose research intersected with data science and computational methods. To make the session more accessible, we presented a condensed version of Software Preservation 101 along with a "show-and-tell" using a few notable items from LGIRA's collection to stimulate conversation. After the presentation, breakout groups were given a series of prompts: "What issues have you encountered while trying to run software that is more than five years old?"; "What approaches could be taken to address those issues?"; "Would emulation tools help address your issues and keep important software alive?" The breakout group discussions were lively, and our debrief after reconvening ranged from the necessity of preserving websites (especially the backend sections that the Internet Archive does not capture), the challenges of data ownership, the necessity of implementing records management guidelines that include the saving of printed offline copies of important source code, and the need to consider the effects of security patches on preserved software and its associated software and hardware dependencies.

An anecdotal examination of the relevance of the preservation-through-use concept to those whose domain lies outside of information management and preservation illuminates how that concept manifests itself in different communities. In the third session, perhaps reflective of the audience, nearly all of the cases identified dealt with active use of the software artifact with the mode of use being operational in nature. That is, people talked about the need for preservation in terms of "I need to do X with software Y and I don't care how that happens", contrasting with more conceptual and methodical approaches in the digital preservation community ("software Y needs to be carefully preserved to enable all of W, X, Z"). All of this is to say that the operational way of thinking about preservation appears to be the default mode of thinking in some, perhaps many, use cases and that the destruction of the original artifact is of little consequence if the goal is achieved (e.g., preserving experience or functionality over time). Perhaps the best example of this way of thinking is the way the free and open-source software (FOSS) movement operates. In FOSS development, the software artifact changes over time and those communities are often more interested in preserving functionality into the future, rather than keeping the original artifacts themselves. This operational way of thinking overlaps with preservation-

through-use in its goal of preserving the experience of interactions (whether they occur between humans and machines or between machines themselves). This suggests that sustainability approaches in open source development communities may be more broadly applicable to software preservation communities in general.

In sum, the three sessions served to link together communities of practice with the notion of preservation-through-use as the underlying fabric. Indeed, regardless of which community of practice any particular individual identified with, it appeared that software preservation resonated with individuals when it was encapsulated within the idea of active use.

Challenges in Sustaining a Community of Practice and Lessons Learned

The three sessions we organized to foster a software preservation community of practice at the University of Arizona illuminated two significant hidden challenges to making such a community sustainable. First, we discovered that just because attendees are interested in software preservation does not mean they share a baseline understanding about what such work entails. Although participants had a general belief that preservation was important, they often had little knowledge about what we considered basic concepts and terms. We found ourselves explaining, for example, the difference between source code and executables, what constitutes a preservation workflow, and the meaning of emulation in the context of software. The Software Preservation 101 module we created addresses these and other essential questions, allowing us to introduce the rudiments of software preservation to a wide array of individuals.

The second challenge we discovered is that participants without an archival or digital preservation background tend to think of software preservation in operational rather than conceptual terms. This means that instead of working to translate and transfer knowledge across domains in order to solve local software preservation problems (as archivists and preservationists are trained to do), they tended to work only within the frames of their respective disciplines—which generally do not have established methods for doing digital preservation. We believe this may be the most salient challenge for building sustainable software preservation communities of practice. The need for common conceptual frameworks, lexicons, and taxonomies in cross-domain collaborations is vital to software preservation if only because so many technical and disciplinary discourses are always at play within any such initiative. Establishing a baseline of concepts, tools, and techniques among all participants is vital, and it has been the recent focus of UA-SPIG's work. An example of this recent work is the hosting of a seminar to communicate the role of emulation in software preservation. The goal was to give attendees a common understanding of what it means to use emulation for preservation and present an in-depth walkthrough of what it takes to go from physical media, to imaging the media, to playing back the image in an emulator. The seminar took place several months after the third UA-SPIG session. A broad invitation was issued and although individuals from areas where software preservation may manifest itself (e.g., digital arts, information studies, museums and archives) were specifically invited, the relatively few attendees from those domains may indicate that software preservation is not on their radar, despite the notion that those fields have a demonstrable intersection with software preservation. Although there may be several reasons for that, one possible reason may be a lack of awareness of what software preservation is.

In light of these challenges, it is reasonable to ask: how will the efforts to sustain preservation efforts be themselves sustained? In other words, how will the community we are fostering live on? To address this challenge, we are aiming to not only hold more

independent UA-SPIG sessions but to also link with other local and relevant communities of practice. One such community is a local “hack” community under the umbrella of Research Bazaar. This community organizes regular “Coffee and Code” and “Hacky Hour” sessions intended for individuals to chat and get help around specific software or science problems.

Sustaining UA-SPIG

The third UA-SPIG session revealed that local “hack” communities may be able to provide the type of fertile ground that needed by groups like UA-SPIG to be sustainable by providing an established framework in which it can exist. That is, community meetups like “Hacky Hour” and “Coffee and Code” at the University of Arizona, provide an established, regular meeting location and time to work on problems related to software preservation. At the same time, the kind of enthusiastic individuals that choose to attend these kinds of casual meetups present an engagement opportunity with those that may have a latent interest in software preservation. At these meetups, individuals tend to focus on their own projects, however, a communal setting where conversations can be overheard, ideas proposed, and insights gained through osmosis is nevertheless valuable. The fact that these meetups are not dedicated to any particular problem, instead broadly focusing on individuals helping each other “do” computational research means that sub-communities like UA-SPIG can follow the ebb and flow of interest/availability of individuals without the need to constantly maintain and plan dedicated events. An informal poll of regular attendees at one Coffee and Code session revealed that there is interest in software preservation-related conversations, especially as they relate to tackling challenges related to the problem of preservation for research reproducibility. By leveraging ideas such as preservation-through-use as it relates to video games, and tools like emulation and the lessons learned we learned about disseminating software preservation knowledge, we aim to generate further awareness of software preservation, continue our conversations, elicit new interest, and tackle preservation problems.

Final Remarks

We presented preservation-through-use and building communities of practice as two ways to sustain preservation efforts. We then argued that they are interrelated. We then leveraged the preservation-through-use concept to begin building a community of practice around software preservation and illuminated challenges specific to sustaining a software preservation community. The main ones include the lack of a basic awareness of what software preservation is along with a lack of shared vocabulary to have cross-pollinating conversations and that once individuals have been exposed to software preservation, that they tend to think of it in operational terms.

With these challenges in mind, we aim to continue our work mainly in two ways. The first is by leveraging the local “hack” community through the Coffee and Code and Hacky Hour meetups to sustain the local software preservation conversations in a more cohesive environment. The second is a continued focus on creating awareness of software preservation in general through seminars and workshops. Although the challenges related to sustaining a local software preservation community are not unique, we believe that partnering with like-minded communities presents a more feasible path towards long-term sustainability of software preservation efforts. This approach has been demonstrated

at large scales through the broader efforts of the Software Preservation Network and as the partnership unfolds, we expect to demonstrate the same at local scales.

Acknowledgements

This work is supported by the Software Preservation Network project “Fostering Communities of Practice” which is funded by the Institute for Library and Museum Services [IMLS grant RE-95-17-0058-17].

References

- [journal article] Abramatic, J.-F., Di Cosmo, R., & Zacchiroli, S. (2018). Building the universal archive of source code. *Communications of the ACM*, 61(10), 29–31.
<https://doi.org/10.1145/3183558>
- [journal article] Baillieul, J., Grenier, G., & Setti, G. (2018). Reflections on the Future of Research Curation and Research Reproducibility. *Proceedings of the IEEE*, 106(5), 779–783.
<https://doi.org/10.1109/JPROC.2018.2816618>
- [web page] Brown, C. T. (2017, January 17). How I learned to stop worrying and love the coming archivability crisis in scientific software. Retrieved December 23, 2018, from <http://ivory.idyll.org/blog/2017-pof-software-archivability.html>
- [journal article] Chassanoff, A., & Altman, M. (2019). Curation as “Interoperability With the Future”: Preserving Scholarly Research Software in Academic Libraries. *Journal of the Association for Information Science and Technology*, 0(0).
<https://doi.org/10.1002/asi.24244>
- [report] Chue Hong, N., Crouch, S., Hettrick, S., Parkinson, T., & Shreeve, M. (2010). *Software Preservation Benefits Framework*. Software Sustainability Institute.
- [conference paper] Di Cosmo, R., & Zacchiroli, S. (2017). Software Heritage: Why and How to Preserve Software Source Code. *IPRES 2017 - 14th International Conference on Digital Preservation*, 1–10. Retrieved from <https://hal.archives-ouvertes.fr/hal-01590958>
- [web page] Fostering Community of Practice in Libraries, Archives and Museums. (n.d.). Retrieved June 28, 2019, from <http://www.softwarepreservationnetwork.org/fcop/>
- [web page] Hees, V. van. (2017, September 11). 10 Ways to keep your successful scientific software alive. Retrieved November 1, 2017, from Netherlands eScience Center website: <https://blog.esciencecenter.nl/10-ways-to-keep-your-successful-scientific-software-alive-61ac81f36a87>
- [white paper] Kaltman, E., Wardrip-Fruin, N., Lowood, H., & Caldwell, C. (2014). *A Unified Approach to Preserving Cultural Software Objects and their Development Histories*.
- [book] Kirschenbaum, M. (2008). *Mechanisms: New Media and the Forensic Imagination* (p. 3). Cambridge, MA: The MIT Press.

- [book chapter] Lowood, H. (2013). The Lures of Software Preservation. In T. Owens (Ed.), *Preserving.exe: Toward a National Strategy for Software Preservation* (pp. 4–11).
- [journal article] Lynch, C. (2017). Stewardship in the “Age of Algorithms.” *First Monday*, 22(12). Retrieved from <http://firstmonday.org/ojs/index.php/fm/article/view/8097>
- [journal article] Matthews, B., Shaon, A., Bicarregui, J., & Jones, C. (2010). A Framework for Software Preservation. *International Journal of Digital Curation*, 5(1), 91–105. <https://doi.org/10.2218/ijdc.v5i1.145>
- [journal article] Matthews, B., Crompton, S., Jones, C., & Lambert, S. (2015). Towards the Preservation of the Scientific Memory. *International Journal of Digital Curation*, 10(1), 196–209. <https://doi.org/10.2218/ijdc.v10i1.361>
- [report] McDonough, J., Olendorf, R., Kirschenbaum, M., Kraus, K., Reside, D., Donahue, R., ... Rojo, S. (2010). *Preserving Virtual Worlds Final Report* (p. 195).
- [journal article] Meyerson, J., Vowell, Z., Hagenmaier, W., Leventhal, A., Roke, E. R., Rios, F., & Walsh, T. (2017). The Software Preservation Network (SPN): A Community Effort to Ensure Long Term Access to Digital Cultural Heritage. *D-Lib Magazine*, 23(5/6). <https://doi.org/10.1045/may2017-meyerson>
- [white paper] Monnens, D., Armstrong, A., Ruggill, J., McAllister, K., & Vowell, Z. (2009). *Before It's Too Late: A Digital Game Preservation White Paper* (H. Lowood, Ed.). Game Preservation Special Interest Group, International Game Developers Association.
- [journal article] Peer, L., & Wykstra, S. (2016). New Curation Software: Step-by-Step Preparation of Social Science Data and Code for Publication and Preservation. *IASSIST Quarterly*, 39(4), 6. <https://doi.org/10.29173/iq902>
- [web page] ResBaz Tucson (2019). Retrieved June 28, 2019, from <https://researchbazaar.arizona.edu/resbaz/resbazTucson2019>
- [report] Rios, F., Almas, B., Jabloner, P., Contaxis, N., & Chassanoff, A. (2017). *Exploring Curation-ready Software: Improving Curation-readiness*. <https://doi.org/10.17605/OSF.IO/T9G3Q>
- [blog post] Rosenthal, D. (2018, March 13). The “Grand Challenges” of Curation and Preservation. Retrieved March 13, 2018, from <https://blog.dshr.org/2018/03/the-grand-challenges-of-curation-and.html>
- [Journal article] Rothenberg, J. (1999). *Ensuring the Longevity of Digital Information*. Retrieved from <http://www.clir.org/wp-content/uploads/sites/6/ensuring.pdf>
- [journal article] Ruggill, J. E., & McAllister, K. S. (2011). Computer Game Archiving and the Serious Work of Silliness. *Animation Journal*, 19, 10.
- [journal article] Shustek, L. (2006). What Should We Collect to Preserve the History of Software? *IEEE Annals of the History of Computing*, 28(4), 112–111. <https://doi.org/10.1109/MAHC.2006.78>

- [journal article] Swade, D. (1993). The Problems of Software Conservation. *Computer Resurrection*, (7). Retrieved from <http://www.computerconservationsociety.org/resurrection/res07.htm#f>
- [book chapter] Swade, D. (1998). Preserving Software in an Object-Centered Culture. In E. Higgs (Ed.), *History and Electronic Artefacts* (pp. 195–206). Oxford: Clarendon Press.
- [conference paper] Thain, D., Ivie, P., & Meng, H. (2015). Techniques for Preserving Scientific Software Executions: Preserve the Mess or Encourage Cleanliness? *Proceedings of the 12th International Conference on Digital Preservation (IPRES)*. <https://doi.org/10.7274/R0CZ353M>