Emulation Encounters: Software Preservation in Libraries, Archives, and Museums[1]

Amelia Acker
School of Information
University of Texas at Austin
aacker@ischool.utexas.edu

Abstract

This paper reports on early findings of research in 2019 following 3 small teams of information professionals as they implemented emulation strategies into their day-to-day work at a museum, a university research library, and a university research archive and technology lab. Findings are reported from workplace observations and semi-structured interviews with preservationists (N=25) as they implement software emulation programs in cultural heritage institutions that collect and preserve software for access. Results suggest that the distributed teams in this cohort of preservationists have developed different emulation practices for particular kinds of "emulation encounters" in supporting different types of use and users. I discuss the implications of these findings for digital preservation research and emulation initiatives providing access to software or software-dependent objects, showing how implications of these findings have significance for those developing software preservation workflows and building emulation capacities. This article suggests that there are different emulation practices for preservation, research access, and exhibition undertaken by preservationists in libraries, archives, and museums; and in examining particular visions of access these findings call into question software emulation as a single, static preservation strategy for cultural heritage institutions.

Keywords

Emulation practices, software preservation, access

[1] This is a pre-print-version of the text "Emulation Encounters: Software Preservation in Libraries, Archives, and Museums" that will be published in the Proceedings of the 83rd Annual Meeting of the Association for Information Science & Technology | October 24-28, 2020, forthcoming.

INTRODUCTION

Libraries, archives, and museums are at the beginnings of a significant change in providing access to digital cultural memory. This digital preservation change is driven by software, software-dependent media, and the increasingly affordable storage for digital artifacts and data collections. With the ever expanding reach of software-driven technologies in our lives, preserving it becomes necessary in the provision of information services, ranging from research data access to software-dependent information objects. But accessing obsolete software is time and resource intensive. Providing access to information through software emulation techniques will likely transform the culture, practice, and access experiences to digital cultural heritage as well as best practices for digital preservation professionals. As such, the study of emerging emulation and software preservation practices in professional communities is necessary for the library and information science community to identify capacity gaps and educational development opportunities. This paper reports on early findings of research in 2019 following three small teams of preservationists, librarians, archivists, and information professionals as they implemented emulation strategies into their day-to-day work at libraries, university research archives and museums.

Although there is a robust literature on digital preservation of information objects in cultural heritage institutions (Corrado, 2019; Meyerson, 2014), software preservation with emulation is understudied and continues to focus on best practices for format migration, digital storage, and description for access. Little research has examined the coordination, decomposition of tasks, or the work of implementing emulation programs that information professionals undertake *in situ*. While many scholars in human computer interaction (HCI) research and science and technology studies (STS) have examined the work practices of obsolete software and developing working strategies to confront legacy software amongst large teams for building and maintaining infrastructure initiatives (Borgman et al., 2014; Cohn et al., 2009; Kelty & Erickson, 2015), few consider software emulation as an information service provision that libraries, archives, and museums (hereafter “LAMs”) increasingly provide as part of their cultural heritage missions. Yet software and software-dependent media are increasingly collected, accessioned, and curated in LAM collections.

This paper argues that an analysis of preservationists’ implementation of software emulation can illuminate the requirements and design of emulation programs for access in cultural heritage contexts. As part of a larger study on software preservation and curation community building, I investigated the workflows and documentation by which small teams of preservationists implement and overcome emulation constraints in order to provide access to software using emulation practices. Software emulation strategies for access in libraries, archives, and museums are complex, time consuming, and frequently present unknown technical problems of trial, analysis, and testing to achieve reliable access to preserved materials. The preservationists that I interviewed overcame these challenges by envisioning access points that would serve various types of users, including themselves as archivists, librarians, and museum workers as well as the researchers and patrons they serve. Such practices and visions of access would then lead to, what I identify, as “emulation encounters”. Here, I focus on these visions of access by examining workflows of preservation teams and discuss challenges and concerns they reported across their institutions.

This paper focuses on answering these research questions**:** How are software preservation practices applied in different stewardship contexts and cultural heritage organizations? How are administration and technical workflows, in support of emulation work developed and deployed across different work sites? The research questions were designed to expand the study of software curation, digital preservation, and emulation practices in information institutions. In the next sections I provide background, motivation, and theoretical frame for this emulation preservation research. Then I discuss methods used in the design of the study and the interview data collected. These sections are followed by the major findings and discussion related to types of users and emulation encounters observed throughout the interviews and observational fieldwork.

BACKGROUND: DIGITAL PRESERVATION PRACTICES, SOFTWARE EMULATION, AND STUDYING WORKFLOWS AS PROCESS

Since the 1990s, librarians, archivists, and museum professionals have been concerned with the digital preservation challenges that software and software-dependent artifacts pose for accessing information (Garrett & Waters, 1996; Hedstrom, 1995; Rothenberg, 1999). In 2013, the United States Library of Congress convened a summit called, "Preserving.exe" to develop a national strategy for addressing software as cultural, historical, and scientific artifacts to preserve. The outcomes and the final report emphasize an increasing understanding that "software should be preserved, archived, for its own sake" (National Digital Information Infrastructure and Preservation Program, 2013, p. 23). Meyerson has characterized the current landscape as a result of two competing orientations or goals for understanding *what* is being preserved—on the one side is the problem of "software as a utility" and on the other is "software as evidence and information" (Meyerson, 2014).

For digital preservation techniques where machine readable digital data needs to be accessed, software is simply a utility and data migration is a sufficient digital preservation strategy. But if software is understood as information or evidence of a form or experience, then 'preserving software for its own sake' becomes a new digital preservation hurdle—both technical and theoretical for information professionals. Early accounts of software emulation, like Rothenberg's, argued that emulation is the "only reliable way to recreate a digital document's original functionality, look, and feel" (Rothenberg, 1999, p. 17). And while emulation does capture the functional and performative aspects of the software experience, most digital preservation practitioners agree that sustained support and capacity building are still needed. It is incredibly resource-intensive for information institutions with limited operating budgets, the breadth of technical expertise needed is high for working professionals, and user demand for accessing obsolete software from collections is still unproven. A recent survey of the digital preservation landscape found that some community members have begun to gain experiential knowledge of emulation for software preservation as they move "techniques from theory to practice" but that increasing knowledge of these preservation strategies is essential to addressing challenges and gaps in professional knowledge and dedicated operational resources (Rieger, 2018, p. 4).

Despite the increasing prevalence of software preservation in information institutions and concerns about these challenges amongst information professionals, scholarly attention to the software maintenance practices of librarians and archivists has been limited. Some notable recent examples include Chassanoff, Donaldson, and Kriesberg working on software curation in support of research data management practices in government, scientific repositories, and university research libraries (Chassanoff & Altman, 2020; Donaldson, 2019; Kriesberg et al., 2017). Even so, the coordination and social practices of software preservation projects often focus research on cyberinfrastructure initiatives, engineering teams, or space missions (Cohn et al., 2009; Mackenzie, 2006; Sim et al., 2009). These studies all prove one thing for researchers studying legacy software—that software maintenance is a team sport. But how should we study these processes amongst preservationists where maintenance in support of long-term collections access is a primary work responsibility?

For this research, I drew from STS studies and infrastructure studies perspectives that see software preservation as a collaborative work process between different sets of actors, technologies, and standards. Understanding software preservation as a collaborative process, allows us to examine emulation through workflows amongst small groups of experts in larger organizations (Bowker & Star, 2000). Since software emulation is a collaborative system that supports complex social interactions with people, and things, and people's (digital) things from the past being made present, we need a deep understanding of how people and groups work together in support of accomplishing tasks together. One way to observe this is through workflows of individuals and teams. For most STS and infrastructure studies scholars, as well as CSCW researchers, examining administrative and technical workflows means studying the processes of work in place (Kling, 1991), arrangements of people and technology, and then their practices (Star & Strauss, 1999) in the everyday—for example working with particular integrated library system software or a microfiche machine day-in and day-out as part of your work providing access to collections within teams of other information professionals in a library, archive, or museum.

As many information scholars have shown, software intensive digital work, such as preserving software-dependent artifacts, is often hidden, obfuscated or abstracted. So revealing this digital labor and making it known from a theoretical perspective that unpacks collaborative actions as socio-technical processes can lead to the benefits of transparency and accountability (Wolf et al., 2019). In addition to the process of using software, the maintenance work involved in digital preservation infrastructures is typically hidden, not documented, or not well known in hierarchical organizations such as universities or academic libraries (The Information Maintainers et al., 2019). Accountability measures for those involved in digital labor of maintenance and coordination can be essential to both the design and transformation of future and current systems because we can support, intervene, and iterate processes in more effective and transparent ways. Thus, documenting complex processes like software preservation workflows increases best practices for professional communities but also generates understanding of these emerging digital preservation processes for future training and research.

We can expect to see a number of changes in the digital work of preservation within LAMs as they being to provide more and more access to not only digital objects and research data, but software experiences and software-dependent objects through emulation techniques. In order to prepare for these changes, and theorize their significance to preservation and memory work, and to measure their impact in the information professions, we must understand how preservationists understand emulation *in situ* with grounded observations of software preservation workflows. Studying workflows in organizations as complex social interactions, and then depicting them as embedded in place are also important for accurately capturing the many layers of social and technical interactions in the adoption of new technology and practices of work (Bailey & Leonardi, 2015). Researching workflows as processes can also reveal how infrastructure is made up of humans in their experiential knowledge and supporting efforts coordinating and maintaining information systems over time and in teams (Lee et al., 2006). This theoretical framework also emphasizes understanding the articulation and coordination amongst teams or members in an organization, because articulating existing or ongoing processes helps people know what they are doing as part of working with complex systems, such as an information organization responsible for providing software preservation services (Wolf et al., 2019). Finally, empirical observations about workflow processes, like the software emulation practices I researched, also enables the transparency and accountability of those processes once they have become formalized in place and enacted (Dourish, 2001; Wolf, 2019).

METHODS

This research was conducted as part of *Fostering Communities of Practice in Software Preservation and Emulation* (FCOP), a two-year project aimed at broadening participation in software preservation through a cohort model (Software Preservation Network, 2017). The findings presented in this article are the result of the author's 16-month research project examining how preservationists in teams develop and implement software emulation programs into their everyday work practices in LAMs. In spring 2017, a call for proposals was circulated to advance digital preservation practice and explore the challenges providing access to software-dependent cultural heritage. Six teams were then selected to participate in the FCOP project beginning spring of 2018. The project teams, made up of institutions all throughout the United States, comprised a large cohort of preservationists. The cohort would have bimonthly "share" calls, public presentations, and knowledge sharing networking events in order to foster community learning about emulation beyond the project's end date. In 2018, I joined the project as a team researcher and followed online cohort activities until summer 2020. This article reports on field site observations and interviews from three cohort teams that took place in the summer of 2019. Research participants included archivists, catalogers, public services staff, metadata librarians, systems administrators, software developers, and administrators, amongst other titles.

My primary method of data collection was semi-structured and in-depth interviews with 25 employees of three field sites, where participants were members of preservation, access, or support teams. Teams included staff members, engineers, software developers, interns and managers for a total of 25 interviews (52% Female, 48% Male) for approximately 31 hours. Interviews typically lasted from 30 minutes to 1 hour, with the shortest being 18 minutes and the longest being 2 hours and 2 minutes. I received IRB approval from my institution's Office of Research Support and Compliance, fieldwork observation primarily occurred during summer of 2019. I embedded for three days at each field site to observe teams, workflows, and exhibitions of software emulation. The fieldwork observations combined with 25 one-on-one interviews resulted in over 70 hours of rich, interview data and hundreds of pages of field notes, photographs, participants' drawings (or "maps") of workflows, and workplace documentation.

This research involved observing people in their workplaces and recording interviews, so I developed a verbal consent protocol that would assure confidentiality and privacy of participants. Field sites were all in the U.S. and included a research library at a large public university, a research archive and technology lab at large public university, and a technology museum. The questions I asked the participants focused on their experiences with emulation practices, in locating, accessing, understanding, and evaluating software and software-dependent objects for emulation projects. I developed an initial coding scheme to analyze the interview data, refined the codes after applying them to a subset of key interviews. The codes were based on my initial research questions and prior relevant literature. After open and axial coding the interview data, the core variable themes for selective coding were "access," "emulation experience," and knowledge from "processes and workflows". Except in one case, all the preservationists I interviewed were employees of these cultural heritage institutions. In the one exception, the interviewee was a student intern who was employed as a graduate research assistant designing and implementing online software emulation exhibits as a software developer.

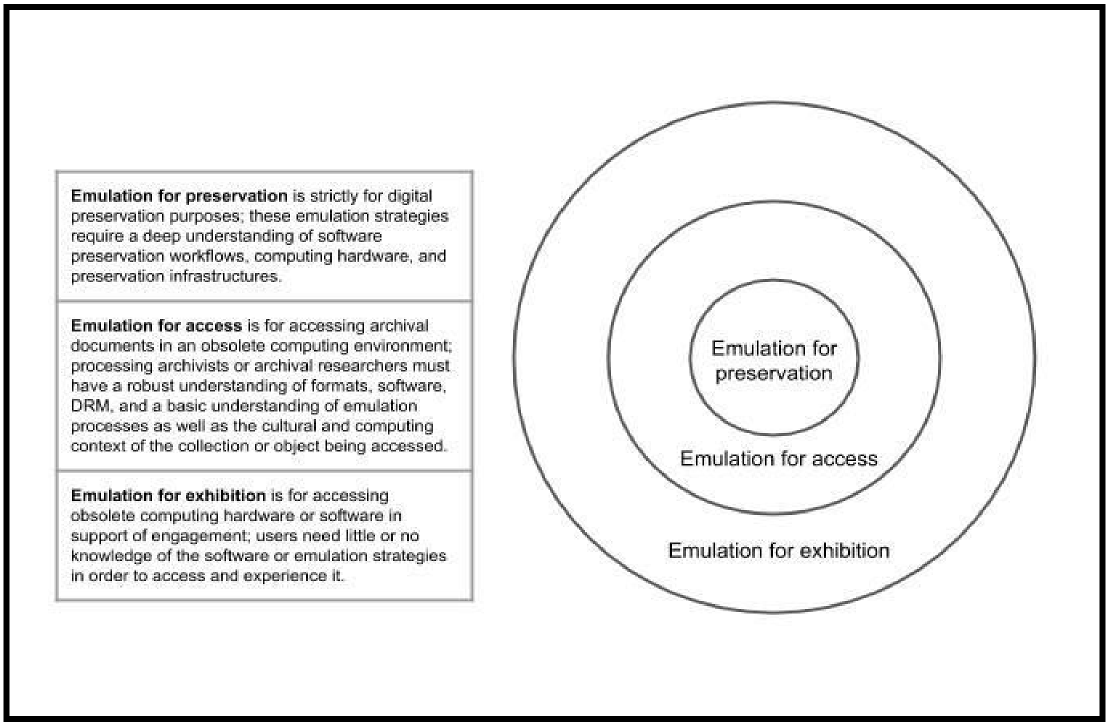
For interviews, I used the participatory workflow analysis method of inquiry. In this method, the interviewer asks participants to map out or draw their workflows, or describe them step-by-step by decomposing a specific task in their daily workflows individually and amongst teams (Chin et al., 2002). By asking participants about each step in their workflow and developing diagrams to structure our conversations, I was able to observe parts of these flow diagrams at each site. After interviews, I followed participants around their work sites and observed their work with emulation, metadata development for description and access, and software preservation processes. In addition to interviews, observations, and field notes, I also observed the conduct of everyday team meetings coordinating services, programming and preservation work of these information institutions. Parts of meeting notes were subject to analysis, I took note of actors' accounts of their work when it was strictly relevant to the implementation of emulation projects or in coordinating participation in the FCOP cohort initiatives.

RESULTS AND DISCUSSION

Overall, each of the field sites had different cultures of teamwork, documentation, and user services. Here I present three trends from major observations found across all these sites, providing some illustrations for how they were enacted, as well as contexts for definitions that could be applied in future software preservation research and user studies of emulation.

Emulation Practices for Different Access Purposes

Libraries, archives, and museum institutions preserve, curate, and provide access to their collections in many different ways (Corrado & Sandy, 2017). Overall, when observing emulation practices at different sites, I found preservationists' knowledge about emulation playing out in different, but interconnected ways as emulation practices were planned and initiated. These emulation practices may be understood as nested practices that begin at the center and radiate out from preservation, to access, to exhibition (Figure 1).



Emulation for preservation is strictly for digital preservation purposes; these emulation strategies require a deep understanding of software preservation workflows, computing hardware, and preservation infrastructures.

Emulation for access is for accessing archival documents in an obsolete computing environment; processing archivists or archival researchers must have a robust understanding of formats, software, DRM, and a basic understanding of emulation processes as well as the cultural and computing context of the collection or object being accessed.

Emulation for exhibition is for accessing obsolete computing hardware or software in support of engagement; users need little or no knowledge of the software or emulation strategies in order to access and experience it.

Figure 1. Types of emulation practices.

The first kind of practice, “emulation for preservation” is in support of pure digital preservation—for example copying disk images, accessing bitstreams or making preservation copies of source code. These practices are involved in existing preservation workflows and involve a deep understanding of emulation as a conceptual process, techniques, and a series of problem-solving tactics. At the research library, emulation for preservation was part of a larger preservation workflow processing archival collections, and was accomplished in a case-by-case basis that was often driven by the needs of researchers. Emulation with preservation methods are sometimes referred to as software recovery or software reconstruction. Recovery efforts begin with verification tests or brief iterative experiments involving an assemblage of hardware, software, and software objects to demonstrate that such interventions are initially even feasible for preservationists to undertake. Emulation for preservation also involves a broad knowledge of computing hardware, software, computing cultures in the past, storage formats, even legal policies like DRM or software licensing. At the technology museum, background research on the software license would often lead an emulation project, and if a title was included in the collection, a preservation copy for each listing in their collections inventory was their ultimate goal.

I observed preservation emulation practices at all three sites, but it was often delegated as a core responsibility for one team expert as part of their work accessioning, describing, or cataloging software materials into the collections. When witnessing emulation for preservation in practice, experts often described this work as a kind of “time traveling,” discovery, and “mysteries” to be solved, and most of this work was approached as open-ended with anticipated hurdles. This adventurous, action language is often used because emulation for preservation is often done on a case-by-case basis, which then results in experiential (and usually singular) knowledge gained by the sole expert carrying out the task. All participants reported that such tasks would take unknown amounts of time that could not be easily predicted if a previous workflow with similar software or hardware had not been created and documented. Emulation with the goal of preserving a disk image for example, involves planning for unknown and unpredictable results—such as a missing manual with a software license key or a corrupted blank floppy disk. One preservationist I followed, Lara described unknown these hurdles as a “void” at the end of most emulation workflows.

Emulation for preservation workflows often involved looping back or beginning again (with recovery and reconstruction tests), repeating serialized trial and error work in order to refine the proper tool chain and process steps to successfully accomplish the digital preservation goal in a repeatable, reliable emulation process. One day Lara and I tried to confirm and verify an emulated CD-ROM game and diagnosing a glitch that had been reported. We sat and waited for about 5 minutes. “I call it madness...torment. I’ll try and multi-task and then I’ll drift off to another dimension,” she said. After rebooting the machine and restarting the emulation environment, we were able to diagnose the problem. Specific workflow processes, such as diagnosing a glitch or verifying a preservation copy, would be documented in project management documents, ready-reference sheets, and manuals for future team members. However, the workarounds, loops, and repeated processes would often take long periods of time. While tips and tricks like rebooting a machine and restarting an emulation environment were documented in manuals, the estimated time to overcome hurdles by repeating steps were not often captured in process documentation.

A second layer of emulation practices involves using emulation in support of archival access, usually by users who are researchers accessing collections materials. By “archival access” I mean the processes that are related to traditional archival collections and contemporary experiences of accessing digital archives, primary sources, and special collections (Gilliland, 2017). These practices typically are concerned with issues of description, providing access (virtually or physically in a reading room), public services, and the actual delivery process of providing information to a user, as well as the policies and information sources that are used to communicate how to access materials to the intended users. When asked about how they represent materials like an emulated video game, Julian prioritized the experience of access for users as an archival experience, “I’m thinking how an emulator changes the lived experience of [playing] the game.” But Julian also spent time weighing the description of software emulation as a utility or a thing because the research library’s organizational remit of service provisions are divided up by teams who staff, resource, and manage services in distinctive ways. Different operations teams within the institution are responsible for providing access to artifacts or managing technology services for example, so for Julian, “it becomes more about communicating how to provide access to output versus an artifact, which may include but is not driven by IT services.”

Emulation for archival access assumes two layers of expertise from the preservationists in order for an access experience to successfully be achieved by the archival researcher or patron. First, those who support representation and access systems—archivists, catalogers, and metadata librarians must describe what has been emulated and what can be meaningfully rendered in providing access to these collections. These descriptions must then be made available in the public-facing institutional catalog system used by researchers. Second, emulation for access presumes an archival user who engages with software or software-dependent information in order to do their research so that once the information has been successfully emulated they can carry out their typical research practices of confirmation and discovery. For the sites with public facing catalogs to their collections, teams would often bring in description experts to consult on naming conventions or to advise with professional descriptive standards such as the "UC Guidelines for Born-Digital Archival Description" (University of California Systemwide Libraries, 2017). In such archival access workflows, teams often had discrete hand-off points once the preservationist and achieved a preservation copy for the permanent collection, they would then hand-off the project to team members responsible for metadata cataloging, public services, access, and exhibition who would then be responsible for describing and representing the emulation to the patrons that these different LAMs primarily serve.

A third type of emulation practice can be seen in those strategies used for exhibition and public engagement, typically found in museums or in display exhibits curated by archives and special collections. Emulation in support of exhibition may serve users with brief encounters. These tend to be the most abstracted with the most restricted access windows of time, but these encounters and underwritten by the proceeding emulation practices of preservation and archival access in order for the exhibition to succeed in rich sense-making of the software or hardware reception process. When asked about the purpose of emulation in exhibitions, a preservation engineer at the museum told me that "[t]he emulation is in support of experiences with the machines." In other words, the software emulation is in the service of touching and experiencing the hardware in the exhibit. At the tech museum and the archive technology lab exhibits usually had a rolling cart or a chair at an exhibition station, allowing users to sit and engage with the software and hardware exhibit.

Emulation Encounters

Each of these interconnected emulation practices accomplished with small teams support many different users in *emulation encounters* of software preservation performances. In addition to the types of users I observed, Table 1 describes the varying conditions of access and motivations for possible emulation encounters in libraries, archives, and museums that were witnessed during fieldwork.

Types of Users	Emulation Encounters
***Preservationist users* are experts that use emulators in their day-to-day work**	● Using emulation to extract a bit stream of software from a floppy disk to maintain a preservation copy of the software for the collection ● Creating a hardware emulator of a peripheral device for a computing machine in the collection for intended archival users ● Using a disk emulator to make a copy of a video game for exhibition users
***Archival users* are those that use emulation for their research accessing software-dependent materials**	● Using a virtual machine to run an historic operating system, examine file directory structures and open a proprietary file format ● Using an emulator to explore a virtual environment for designing and running game simulations ● Listening to a “software story” or oral history featuring a creator describing the motivations for developing a digital experience with software
***Exhibition users* are those that encounter software emulation in artificial exhibition experiences or computing displays**	● Playing Oregon Trail using a floppy disk emulator ● Using an Apollo II flight simulator at a space center’s visitor center ● Playing a cloned arcade game in an exhibit on the video game design

Table 1. Types of Emulation Users and Encounters.

The first type of user that I observed (and interviewed throughout the fieldwork) were “preservationist users”. Conceptually, preservationists who use emulation for archival preservation draw upon a lot of “time travel” techniques described earlier, because there are many layers of hardware, software, and optical or tape storage media that need to work together in order for the preservation outcome to be an authentic, reliable, and trustworthy preservation copy. Often this work involves weighing out the costs, time constraints, and labor resources that would be necessary in support of possible users and the institution’s current collections reach and scope. The typical preservation user for this type of emulation is an expert digital preservationist, who has advanced training as an information professional or staff member whose institutional role is to preserve software-dependent content as part of their role in service provision at the LAM.

Preservationist users engage in software emulation practices most often with archival access points and future users of those archives in mind. As mentioned above, emulation for archival access relies on a team of preservationists to preserve, describe and catalog information resources for a user and assumes a user with a more robust understanding of emulation, so the researcher or hobbyist intending to access materials knows the software need based on the format and needs an access point that the archive, university or museum is providing. This user is usually a historian, a scholar, a researcher, a hobbyist, or a fan with a particular software need and they anticipate what the emulation will provide access to (e.g. “I want to play Ghostbusters 1989 PC game in my browser” or “I want to look at legal briefs created in the Word Star format”). These users may also be archivists themselves, who are responsible for describing, cataloging, and providing access to collections dependent on software and need to gather contextual information for describing the software for inventory or for user documentation. In addition to having a good sense of how emulation enables one to access software or software-dependent objects, archival users must also have expertise or even personal knowledge of the contexts that the collection was created in as well as the computing culture of the era, for example office technology or personal computing practices of the creators. Throughout interviews subjects expressed concern for new generations of users accustomed to personal computing and cloud storage infrastructures and because how long may take to select the proper emulator, open the software and access a file. Still, how much preservationists can assume users will know about earlier computing contexts still remains to be unseen.

Preservationist users, their knowledge, service provision support, and preservation practices are essential for emulation in support of archival users to be successful in their pursuits. Indeed, most archival users themselves are subject specialists and researchers concerned primarily with the evidence stored in the software-dependent object as an archival document, more than the technical processes necessary for them to access it in its recovery from software.

Finally, this fieldwork allowed me to witness and hear professional visions of intended “exhibition users”. For those people who encounter exhibitions featuring software emulation, the emulation is typically in support of broad, public engagement. In a sense, emulation in exhibition is artifactual and so it is not necessarily important for the individual patron, visitor, or user to know how emulation works or that it’s even undergirding the experience of engaging with obsolete software or hardware. Consider a third grade class visiting the museum of technology—it is not a priority to confront the abstraction of a virtualized software environment for an emulated Oregon Trail game on an Apple IIe or to know how to eject a floppy disk to from a disk drive. Instead, emulation for exhibition emphasizes an experiential encounter with an earlier or historic computing experience that may be unknown and unfamiliar to the user. In most cases, the original look and feel is prioritized to support sense-making over the knowledge that the encounter involves emulation or a virtual layer that “gets to go” immediately. Exhibitions for LAMs assume patrons from all backgrounds and all ages. So, exhibition users are laypersons, who may be involved in a sense-making encounter—such as touching, feeling, figuring out the interface as they encounter the artifact. Many interviewees observed that such exhibits may be the first (and last) time they engage with the historic artifact or software object, so prioritizing a users’ engagement with an immersive computing experience is more important than communicating what, how, and why emulation is occurring.

The When of Emulation

For most preservationists I interviewed, emulation encounters are performances or events with software but communicating how emulation is achieved is a thorny problem of representation. Many scholars who observe software maintenance have discussed issues of representation in layers of abstraction, in virtualizing software as well as capturing its temporality as a “continuously evolving object” (Cohn, 2019, p. 423). While all encounters with emulation practices reveal multiple layers of time between the now, and the past, of software being made present—this layered temporality is difficult to describe because it captures an access experience but not just a single entry point between the legacy software object, the emulator (as a software utility) and the access experience of the software performance within a specific virtual environment.

For preservationists, this frame-within-a-frame-within-a-frame of temporal virtual perspectives, presents the problem of reconciling layers of time but describing possible actions within the virtualized environment as well. So, at the heart of emulation-driven access to software objects is what we might call ‘the when of emulation’ for access. Unlike accessing a book or providing access to a unique artifact in a collection, emulating software-dependent information involves many layers of abstraction, pacing, moving between different layers of time. Each emulation is a meaningfully different technological performance, and each user cannot reliably have the same access experience given the possibilities that emulation affords in the virtualization of software experiences. And so, in keeping obsolete software present, emulation becomes about providing *experiences of access* for the sake of software as information in addition to providing access to software as a utility.

Many interviewees, and team leads in particular, speculated about how these “access experiences” should be described and captured as evidence and as information artifacts themselves: How do we convey to a user who is having a “user session” where the actions of the original creators’ end and the preservation decisions of the archivist begin? How should generic operating features from the operating system or software environment be described? If accessing an individual’s software collection, to what degree should the individual creator’s decisions such as file naming conventions be explained to the emulation user? Should users be able to return, capture, or cite processes they encountered in such user sessions? If yes, how? Or should they have access to code, or logs of commands of actions taken, or possibly snapshots of the beginning, middle, and end of emulation sessions? Such snapshots would give users the ability to consider paths taken and not taken during user sessions, or provide comparisons to the historical subject’s experience of the software at the time of its use. One software developer, David mused that “option mitigated access through the reading room’s emulator” would one day be possible in research archives. Other interviewees suggested that preservationists should anticipate, or even curate, the most desired paths of those encountering emulation given how time consuming the initial orientation to an emulated experience can be. This was described by some informants as the long process of “getting to go” which began with accessing the emulation platform, selecting the proper operating system for emulation, mounting the virtual environment, opening the software and accessing files or media which could take as long as 10 minutes. Many interviewees expressed concern that potential users would be discouraged by the time consumed in “getting to go.” But often waiting for the software or file to load was the emulator working just as long as the original software or media would have operated when it was originally in circulation and use. Coincidentally, many interviewees observed that the time taken to start up the emulation environment was still less than earlier computing eras with microcomputers, or with software stored on magnetic tape, floppy disks, or even optical discs.

Many of these descriptive speculations from interviewees about “getting to go” and imagining users’ needs (and preferences about faster access) reveal a tension in the differences between providing access to objects such as books or digitized documents and providing access to emulated experiences when the software itself is the information being accessed. These visions of emulation tend to emphasize access with potential users—anticipating their needs and predicting their willingness to wait while “getting to go”. Most assumed that contemporary users’ current computing expectations would conflict with earlier eras of technology before the “instant access” of cloud based storage, high speed internet, and mobile computing devices. It is clear that a more generalized vocabulary that addresses ‘the when of emulation’ and pace layering of emulation practices is necessary in order for LAMs to provide access experiences, document ongoing workflows, and generate robust description to these access points as they do now.

CONCLUSION

We are still in the early days of emulation as a service provision in libraries, archives, and museums. A recent survey from the Software Preservation Network found gaps in services and predicts a number of opportunities for more service provision with software preservation programs. In surveying 124 information professionals regarding software preservation initiatives at a variety of institutions, the survey found that most respondents (93.5%) did not yet provide access to software despite having software in the majority of institutional collections (Hagenmaier et al., 2019).

This exploratory research project was motivated by asking questions about emulation processes and if their access outcomes were specified by the unique environs of each place observed. Here I have reported on field observations, interview and participatory workflow analysis of distributed software emulation projects and discovered that emulation practices are shaped by organizational structures, institutions’ access mandates, and preservationists’ visions of access. Overall, findings from each site revealed that amongst three ongoing software preservation programs, emulation can be motivated by different preservation and access mandates, but ultimately these practices are interconnected, radiating from expert knowledge about emulation as a process to passing encounters with emulators ‘hidden’ behind exhibitions of computing experiences. Each of the different emulation practices identified in this research supported access with emulation encounters featuring intended access goals. As such, different emulation encounters will prefigure possible use cases so more user studies are needed to enfranchise diverse user populations while communicating the methods and goals of software emulation.

There is an urgent need to develop comprehensive resources that describe existing approaches and known preservation standards specific to software technology and software emulation in US cultural heritage organizations. This includes a conception of long-term access to digital cultural memory and a general vocabulary for understanding digital preservation from software development to software emulation perspectives with many types of users. Further, we need a theory of digital preservation that accounts for ‘the when of emulation’ in representing software both as a utility for access as well as information itself.

Acknowledgements

Thank you to the Software Preservation Network for facilitating and supporting this research. I am deeply grateful to the memory workers who invited me into their institutions and allowed me to interview, observe, and participate in their valuable preservation work. This project was made possible by a grant from the Institute of Museum and Library Services (IMLS), grant RE-95-17-0058-17.

REFERENCES

Bailey, D. E., & Leonardi, P. M. (2015). *Technology Choices: Why Occupations Differ in Their Embrace of New Technology*. MIT Press.

Borgman, C. L., Darch, P. T., Sands, A. E., Wallis, J. C., & Traweek, S. (2014). The ups and downs of knowledge infrastructures in science: Implications for data management. *IEEE/ACM Joint Conference on Digital Libraries*, 257–266. https://doi.org/10.1109/JCDL.2014.6970177

Bowker, G. C., & Star, S. L. (2000). *Sorting Things Out: Classification and Its Consequences*. MIT Press.

Chassanoff, A., & Altman, M. (2020). Curation as “Interoperability With the Future”: Preserving Scholarly Research Software in Academic Libraries. *Journal of the Association for Information Science and Technology*, *71*(3), 325–337. https://doi.org/10.1002/asi.24244

Chin, G., Schuchardt, K. L., Myers, J. D., & Gracio, D. K. (2002). *Participatory Workflow Analysis: Unveling Scientific Research Processes with Physical Scientists* (PNNL-SA-33676). Pacific Northwest National Lab. https://www.osti.gov/biblio/15001571

Cohn, M. L. (2019). Keeping Software Present. In J. Vertesi & D. Ribes (Eds.), *DigitalSTS: A Field Guide for Science & Technology Studies* (pp. 423–446). Princeton University Press. https://digitalsts.net/essays/keeping-software-present/

Cohn, M. L., Sim, S. E., & Lee, C. P. (2009). What Counts as Software Process? Negotiating the Boundary of Software Work Through Artifacts and Conversation. *Computer Supported Cooperative Work (CSCW)*, *18*(5), 401. https://doi.org/10.1007/s10606-009-9100-4

Corrado, E. M. (2019). Software Preservation: An Introduction to Issues and Challenges. *Technical Services Quarterly*, *36*(2), 177–189. https://doi.org/10.1080/07317131.2019.1584983

Corrado, E. M., & Sandy, H. M. (2017). *Digital Preservation for Libraries, Archives, and Museums*. Rowman & Littlefield.

Donaldson, D. R. (2019). *Scientists, institutional repositories and data management practices*. https://www.ideals.illinois.edu/handle/2142/105361

Dourish, P. (2001). Process descriptions as organisational accounting devices: The dual use of workflow technologies. *Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work*, 52–60. https://doi.org/10.1145/500286.500297

Garrett, D., & Waters, J. (1996). *Preserving Digital Information, Report of the Task Force on Archiving of Digital Information* (pp. 1–71). CLIR. https://www.clir.org/pubs/reports/pub63/

Gilliland, A. J. (2017). *Conceptualizing 21st-Century Archives*. Society of American Archivists.

Hagenmaier, W., Williford, C., Lassere, M., Work, L., Benner, J. G., & Erickson, S. (2019). *Software Preservation Services in Cultural Herigate Organizations: Mapping the Landscape*. 16th International Conference on Digital Preservation iPRES 2019.

Hedstrom, M. (1995). Electronic Archives: Integrity and Access in the Network Environment. *The American Archivist*, *58*(3), 312–324. https://doi.org/10.17723/aarc.58.3.n7353726666u41v5

Kelty, C. M., & Erickson, S. (2015). The Durability of Software. In I. Kaldrack & Leeker, Martina (Eds.), *There is no Software, there are just Services* (pp. 39–55). meson press. http://dx.doi.org/10.14619/008

Kling, R. (1991). *Cooperation, coordination and control in computer-supported work*. Association for Computing Machinery. https://doi.org/10.1145/125319.125396

Kriesberg, A., Huller, K., Punzalan, R., & Parr, C. (2017). An Analysis of Federal Policy on Public Access to Scientific Research Data. *Data Science Journal*, *16*(0), 27. https://doi.org/10.5334/dsj-2017-027

Lee, C. P., Dourish, P., & Mark, G. (2006). The human infrastructure of cyberinfrastructure. *Proceedings of the 2006 20th Anniversary Conference on Computer Supported Cooperative Work*, 483–492. https://doi.org/10.1145/1180875.1180950

Mackenzie, A. (2006). *Cutting Code: Software and Sociality*. Peter Lang.

Meyerson, J. (2014). Software Preservation Literature Review: The Co-Determinacy of User Needs and Advances in Preservation Methods. *Saving Software Together*. https://www.softwarepreservationnetwork.org/blog/software-preservation-literature-review-2014/

National Digital Information Infrastructure and Preservation Program. (2013). *Preserving.exe: Toward a National Strategy for Preserving Software*. http://www.digitalpreservation.gov/multimedia/documents/PreservingEXE_report_final101813.pdf

Rieger, O. Y. (2018). *The State of Digital Preservation in 2018: A Snapshot of Challenges and Gaps*. Ithaka S+R. https://doi.org/10.18665/sr.310626

Rothenberg, J. (1999). *Avoiding Technological Quicksand: Finding a Viable Technical Foundation for Digital Preservation. A Report to the Council on Library and Information Resources*. Council on Library and Information Resources, 1755 Massachusetts Ave. https://eric.ed.gov/?id=ED426715

Sim, S. E., Cohn, M. L., & Philip, K. (2009). The work of software development as an assemblage of computational practice. *2009 ICSE Workshop on Cooperative and Human Aspects on Software Engineering*, 92–95. https://doi.org/10.1109/CHASE.2009.5071419

Software Preservation Network. (2017). Fostering Communities of Practice. *Saving Software Together*. https://www.softwarepreservationnetwork.org/fcop/

Star, S. L., & Strauss, A. (1999). Layers of Silence, Arenas of Voice: The Ecology of Visible and Invisible Work. *Computer Supported Cooperative Work (CSCW)*, *8*(1), 9–30. https://doi.org/10.1023/A:1008651105359

The Information Maintainers, Olson, D., Meyerson, J., Parsons, M. A., Castro, J., Lassere, M., Wright, D. J., Arnold, H., Galvan, A. S., Hswe, P., Nowviskie, B., Russell, A., Vinsel, L., & Acker, A. (2019). *Information Maintenance as a Practice of Care*. https://doi.org/10.5281/zenodo.3251131

University of California Systemwide Libraries. (2017). *UC Guidelines for Born-Digital Archival Description*. https://escholarship.org/uc/item/9cg222jc

Wolf, C. T. (2019). Explainability Scenarios: Towards Scenario-based XAI Design. *Proceedings of the 24th International Conference on Intelligent User Interfaces*, 252–257. https://doi.org/10.1145/3301275.3302317

Wolf, C. T., Bullard, J., Wood, S., Acker, A., Paine, D., & Lee, C. P. (2019). Mapping the "How" of Collaborative Action: Research Methods for Studying Contemporary Sociotechnical Processes. *Conference Companion Publication of the 2019 on Computer Supported Cooperative Work and Social Computing*, 528–532. https://doi.org/10.1145/3311957.3359441