Received: 30 November 2020 | Revised: 28 February 2021 | Accepted: 1 April 2021

DOI: 10.1002/asi.24482

RESEARCH ARTICLE

JASIST WILEY

Emulation practices for software preservation in libraries, archives, and museums

Amelia Acker

School of Information, University of Texas at Austin, Austin, Texas

Correspondence
Amelia Acker, School of Information, University of Texas at Austin, 1616 Guadalupe St, Suite #5.202, Austin, TX 78701-1213.
Email: aacker@ischool.utexas.edu

Funding information
Institute of Museum and Library Services, Grant/Award Number: RE- 95-17-0058-17

Abstract

Emulation practices are computational, technical processes that allow for one system to reproduce the functions and results of another. This article reports on findings from research following three small teams of information professionals as they implemented emulation practices into their digital preservation programs at a technology museum, a university research library, and a university research archive and technology lab. 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 access. 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. These findings suggest that different emulation practices for preservation, research access, and exhibition undertaken in libraries, archives, and museums result in different forms of access to preserved software—accessing information and experiential access. In examining particular types of access, this research calls into question software emulation as a single, static preservation strategy for information institutions and challenges researchers to examine new forms of access and descriptive representation emerging from these digital preservation strategies.

1 | INTRODUCTION

Libraries, archives, and museums (LAMs) are at the beginnings of a significant change in providing access to digital cultural memory. This change is driven by a freshet of software, software-dependent media, and the increasingly affordable storage for digital artifacts and data collections found in information institutions. With the ever expanding reach of software-driven technologies in our lives, the urgency to preserve software becomes more necessary in the provision of information services, ranging from research data access to software-dependent information objects. One approach to providing access is through emulating software. Emulation practices are computational, technical processes that allow for one system to reproduce the functions and results of another. An emulator is intended to faithfully reproduce information, experience, or form with hardware or software, or in combination. But accessing obsolete software with emulation approaches is time and resource intensive, so emulation as a service provision for long-term preservation and archiving is currently being experimented with (Rechert et al., 2010; von Suchodoletz et al., 2013).

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 needed for the library and information science community to identify capacity gaps and educational development opportunities. This research article reports on findings from research in 2018–2020 following three small teams of preservationists, librarians, archivists, and information professionals as they implemented emulation strategies into their day-to-day work at a university research library, a university research archive and technology lab, and a museum of technology in the United States.

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 descriptive metadata for access (Guttenbrunner & Rauber, 2012). Emulation as a preservation strategy for scientific knowledge and digital cultural memory was first proposed in 1995 (Rothenberg, 1995), and since then has been used alongside migration and other data transfer techniques to provide access to digital objects in LAMs. But over the past 25 years, 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 using obsolete software, there are few studies that examine practitioners and their working strategies to confront legacy software among large teams for building and maintaining infrastructure initiatives (Borgman et al., 2014; Cohn et al., 2009).

Software and software-dependent media are increasingly collected, accessioned, and curated by LAMs, but few researchers have considered how software emulation is becoming an information service provision in these institutions as part of their cultural heritage missions, access mandates, and services provided to stakeholders. As such, an analysis of preservationists' implementation of software emulation can illuminate the requirements and design of emulation programs for access in cultural heritage contexts in ways that can have lasting impact on best practices and educating future digital preservationists and digital collections managers, among other kinds of practitioners working in information institutions. Generalized findings may also guide the standardization of emulation practices for digital preservation. 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 challenges in order to provide access to software using emulation practices for preservation. Software emulation strategies for preservation and access in LAMs are complex, time consuming, and frequently present unknown technical problems of trial, analysis, and testing to achieve reliable access to preserved materials (Bearman, 1999; Thibodeau, 2002). 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 here as "emulation encounters" that allow for different forms or experiences of access to preserved software.

This article reports on findings from the following 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 by focusing on the development and standardization of emulation practices and software preservation workflows. In the next sections, I provide background, motivation, and the theoretical frame for this digital preservation research on software emulation. Then I discuss methods used in the design of the study and the data sources collected. These sections are followed by the major findings and a discussion related to hurdles, themes, and emulation encounters observed throughout the interviews and observational fieldwork.

2 | BACKGROUND: SOFTWARE EMULATION, DIGITAL PRESERVATION, AND STUDYING WORKFLOWS

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). Software emulation as a digital preservation strategy has been hotly debated in terms of costs, scalability, and importantly, the ontological and epistemological status of what is being preserved during the process (sometimes called "the performance") of emulation. Some digital preservation experts, such as Jeff Rothenberg and David Bearman, have been boosters for decades, arguing that emulation has the potential to be a magic bullet confronting the inevitability of software obsolescence. While museum curators and researchers concerned with the historiography of software, electronic literature and digital art have argued that emulation has no place in pure preservation efforts because in its mimicry it updates and fundamentally replaces data or other computational layers of original works to create new versions (Moulthrop & Grigar, 2017; Rinehart & Ippolito, 2014; Swade, 2003). A third strategy exists between these two positions, that of using emulation for rescue and recovery efforts in extracting data or code (Granger, 2000).

In 2013, the U.S. 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 emphasized 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 with emulation strategies—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 need to be accessed, software as a utility for 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 (Granger, 2000; Hagenmaier et al., 2019; von Suchodoletz & van der Hoeven, 2009). It is incredibly resource-intensive for information institutions with limited operating budgets, so no standardized approach to emulation currently exists. The breadth of technical expertise needed is high for working professionals, and user demand for accessing obsolete software from collections is still unproven. While some early studies evaluating emulators exist (Guttenbrunner & Rauber, 2012; Hedstrom et al., 2006), the challenges of representing emulation as an access point and information service to users remain.

There are many ways LAMs could approach providing access to obsolete software-driven experiences with emulation techniques. Consider for example an emulation service that resembles checking out a book or downloading an article from a catalog or scholarly database. Or perhaps emulation could resemble a special collections model where researchers are trained to witness a set of documents in the original order with which a creator originally assembled and used them over the course of their work. A number of initiatives have explored remote services, providing emulation environments for users to potentially access materials online (Rechert et al., 2010; von Suchodoletz et al., 2013; von Suchodoletz & van der Hoeven, 2009). But remote emulation access assumes broad digital rights over the software and knowledge of the original application or computing environment. 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 in LAMs (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 in modern information institutions. But how should we study these processes among preservationists where maintenance in support of long-term collections access is a primary work responsibility?

In following information professionals and preservationists as they implemented new digital preservation workflows with emulation, this research builds on other studies of archivists, data mangers, and information professionals providing long-term access to software-dependent information, digital objects, and databases (Borgman et al., 2016; Thomer et al., 2018; Waterton, 2010). 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). Software preservation, like other collaborative work, can often be seen when following "invisible technicians" who care for knowledge, maintain machines, and manage data archives or digital collections (Orr, 2016; Plantin, 2019). This research on preservationists extends a long history of STS and information studies work that has examined information professionals as invisible technicians, who are now sometimes called "maintainers" (Russell & Vinsel, 2018).

Since software emulation in information institutions is a collaborative system that supports complex social interactions between 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 as they coordinate amongst 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. In addition to the processes 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). 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 broader benefits of transparency and accountability of information institutions (Wolf 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.

3 | THEORETICAL FRAMEWORK

We can expect to see a number of changes in the digital work of preservation within LAMs as they begin to provide more and more access to not only digital objects and research data but also software experiences through emulation techniques. In addition to drawing on research of invisible technicians, maintenance work, and workflows among teams, this research draws upon theories of software as an object that is both *lived with* and *backgrounded* in social practices and coordination among teams.

In theorizing the impact of software, STS researchers, historians of computing, and information researchers have found that software is durable, modular, and becomes entrenched in its use as an epistemic tool for collaborating, communicating, and observing the world (Blanchette, 2011; Ensmenger, 2014; Kelty & Erickson, 2015). It also can easily break and is vulnerable to loss, decay, and rapid obsolescence. According to software studies scholar, Adrian Mackenzie, software is a "multidimensional and mutating object" (Mackenzie, 2006, p. 2). By highlighting its malleability, circulation, and fragility, we can also begin to see the significance of software's *backgrounding* qualities to challenges of digital preservation, description, and representation for those who seek to replicate it for long-term preservation and access.

In addition to Mackenzie's framing of software and its backgrounding qualities, this research draws on Marisa Leavitt Cohn's framing of software's temporalities and forms of duration. In attending to its durations and temporality, Cohn reminds us that software is "an object subject to continuous change and lived with over time as it evolves" (Cohn, 2019, p. 423). For Cohn and other STS scholars of high technology, such as Sharon Traweek, it is this "living with" software over durations that binds people to information infrastructures resulting in an assemblage of professional expertise, hardware and software, code and programming languages, standards and processes increasingly found in virtually all modern, software-driven organizations (Cohn et al., 2009; Sim et al., 2009; Traweek, 1992).

In order to prepare for these changes in preserving obsolete software with emulation practices and to theorize their significance to digital preservation and memory work in information institutions, we must understand how preservationists understand emulation in situ with grounded observations of software preservation workflows as they are lived with and retrieved from the background with emulation techniques. Studying software preservation 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). This theoretical framework also emphasizes understanding the articulation and coordination among 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.

4 | METHODS

This research was conducted as part of *Fostering Communities of Practice in Software Preservation and Emulation* (FCOP), a 2-year project aimed at broadening participation in software preservation through a cohort model (Software Preservation Network, 2017). In spring 2017, a call for proposals was circulated to advance digital preservation practice and explore the challenges of 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 were made up from institutions across the United States and comprised a large cohort of preservationists who volunteered to participate in the research. 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, the author 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. In spring of 2020, I reported my initial findings and themes to the larger group for a member check, receiving feedback, and verifying the results. Research participants included archivists, catalogers, public services staff, metadata librarians, systems administrators, software developers, and administrators, among other titles.

The 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 implementing emulation workflows. 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. (In reporting these results and findings, I have anonymized both the field sites and the interviewees.) I embedded for 3 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.

Field sites included a research library at a large public university, a research archive and technology lab at an engineering-focused 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 using grounded theory techniques (Strauss & Corbin, 1997). The codes were based on my initial research questions and prior relevant literature. Except in one case, all the preservationists I interviewed were full-time employees of these information institutions. In the one exception, the interviewee was a student intern who was employed as a 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 among 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, exhibition programming and designating preservation work of these information institutions. Parts of meeting notes were subject to analysis. For example, 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.

5 | RESULTS

Overall, each of the field sites had different cultures of teamwork, documentation, and user services but similar preservation and access mandates for the information services they provided to stakeholders. Here I present trends, challenges, and access points 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.

5.1 | Emulation practices for different access purposes

LAM 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 beginning with emulation for preservation, then archival access, and then exhibition or display.

5.1.1 | Emulation for preservation

The first kind of practice, "emulation for preservation" is in support of pure digital preservation—for example, copying disk images, accessing original bitstreams, or making preservation copies of source code. These practices are involved in existing digital preservation workflows and rely on 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 is 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 digital rights management (DRM) or software licensing that may prohibit particular access techniques. At the technology museum, background research on the software license would often lead an emulation project, and if a specific software 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, usually it was 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, four team leads described this work as a kind of "time traveling," discovery, or as "mysteries" to be solved. Much of this diagnostic and recovery work was approached as open-ended with anticipated hurdles and dead ends, ranging from hardware failure and corrupted floppy disks to missing software license keys. Preservation for emulation strategies involved several layers of diagnostics to arrive at the source of what is being preserved, ranging from a bitstream on a disk, to some software-dependent data for migration, or a software-dependent experience. For the preservationists I observed, it was not always clear from the outset with emulation for preservation if the goal was to preserve the software as a utility (for data migration) or if it would be preserving the software for its own sake and to support future access experiences. Interviews with four key informants involved adventurous, action language relating to the detective work needed to solve mysteries in order to successfully work through a path to a reliable emulation. I hypothesize that sleuthing metaphors used to explain unknown ends was 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 previously been created and documented. One overarching goal of the large cohort project was to generate case studies and document workflows to address gauging time, pacing efforts, and deploying resources. At all three sites, the common emulation workflow would follow a "stack" or path, beginning with a digital object (such as a file), a software application, an operating system, and computing hardware. The emulation process would begin by identifying an object or application and then working along the path through each layer of the stack in order to create a reliable emulation performance as well as documentation. Despite this common workflow, emulation for preservation, diagnosis, or recovery is unique for each layer of the stack. One interviewee described the lilting workflow of diagnosing across layers of the stack as having "voids" in the middle or at the end of the workflow: "I call it madness...torment. I'll multi-task and then I'll drift into another dimension."

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, encountering a corrupted blank floppy disk, or locating the necessary disk drive adapter. 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. These specific workflow processes would then be documented in project management documents such as lab notebooks, spreadsheets, and manuals for future team members managing emulation projects. Such initial diagnostics or "mysteries" are incredibly time consuming for preservationists and can often lead to discouraging outcomes or incremental successes. When asked about estimating a typical day for emulation for preservation, one interviewee responsible for software curation at the museum reported,

> "All day, six things [software titles] copied to work on the [exhibition] floor is a really good day for this preservation workflow. A bad day is spending the whole day on one thing and only getting a sense of whether the issue is media, format, copy, or an emulator issue."

The unpredictability of success, the amount of time spent in each layer of the stack, and high rates of failure for many new workflows yet to be tested creates a challenge for preservationists to justify emulation services to their managers, and as a result, many interviewees commented on the importance of documenting their processes as evidence of effort for managers and administration. However, such emulation failures that take "a whole day" of effort can result in valuable empirical data for best practices as well as experiential knowledge for the individual building expertise in diagnostics.

5.1.2 | Emulation for archival access

A second layer of emulation practices involves using emulation in support of archival access, usually with users who are researchers. 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 archival practices 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 intended archival users.

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 be standardized and 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 one digital archivist, this description hurdle was in the challenge of explaining, "How an emulator changes the lived experience of the [software]" if the archival user had never had experiential knowledge of living with that software, operating system, or hardware.

For the sites with public facing catalogs for searching their collections, teams would bring in description experts to consult on naming conventions or to draw upon professional descriptive standards such as the "UC Guidelines for Born-Digital Archival Description" (University of California Systemwide Libraries, 2017). Providing access to a software title might look much like any other publication described in a catalog; however, explaining that the title is available through an emulation environment, or in some cases, clarifying that the title could be used to access or extract data from software-dependent files was ad-hoc at each site. In such archival access workflows, teams often had discrete hand-off points between preservation and description. 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.

5.1.3 | Emulation for exhibition

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 in exhibition halls or at public displays, but these encounters are 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. For example, the technology museum is a "touch museum" and makes efforts for almost all its hardware in the collection to be used, touched, and interacted with. Exhibits at the technology museum and the archive technology lab usually had a rolling cart or a chair at an exhibition station (Figure 1), allowing users to sit and engage with the software and hardware exhibit as they would encounter a personal computer at a desk or an arcade console or booth.

Interviewees responsible for the design and curation of exhibitions expressed some dismay about the challenges of explaining software emulation to users in an interactive display, but most assumed that visitors would rarely have personal knowledge of the earlier computing environment on display and emphasized the physical experience of accessing historic software, hardware machines, and obsolete operating systems. Because of the touch mandate, a pristine preservation approach (as seen in archival contexts) is second to supporting physical experiences with machines running software. In observing field sites and emulation practices in place, it was emulators used in exhibitions where the backgrounding qualities of software was often most noticeable (if hardest to investigate) because emulators are technologies of imitation. For many exhibits, the emulator is effectively hidden from the user in the service of demonstrating the hardware or experiencing the software. Few display placards or exhibits that I studied explicitly mentioned that an emulator was being deployed and depending on the curatorial goals and learning objectives of the exhibit, a perfect imitation without a visitor detecting the emulator is evidence of an ideal encounter of experiencing access in emulated software.

5.1.4 | Emulation encounters

From axial coding the interviews, we find that each of these interconnected emulation practices accomplished with small teams support many different users in *emulation encounters* of software preservation performances. In addition to different types of users I observed, Table 1 describes the varying conditions of access and motivations for possible emulation encounters in LAMs that were witnessed during fieldwork.

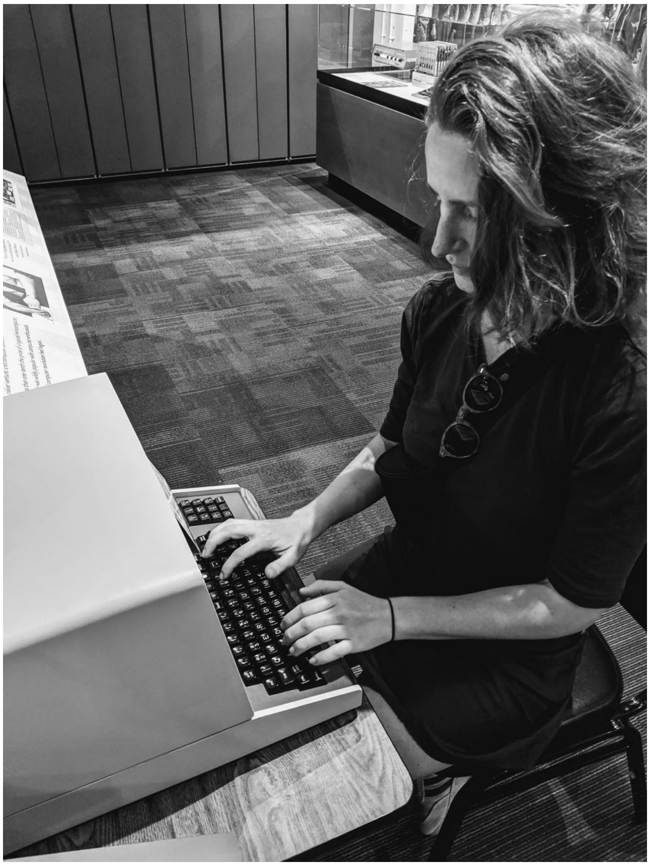

FIGURE 1 An exhibition user sitting at an Altair 8800 station

The first type of user that I observed and primarily interviewed was "preservationist users." Conceptually, preservationists who use emulation for archival preservation draw upon "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 information service provisions.

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

TABLE 1 Types of emulation users and encounters

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 who 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 video game design

researcher or hobbyist intending to access materials knows the software need based on format and needs an emulator 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. Throughout interviews nearly all subjects expressed concern for new generations of users who are accustomed to personal computing and cloud storage infrastructures and are therefore unaccustomed to how long it 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 remains open and further user studies evaluating emulation experiences are needed.

Preservationist users, their technical knowledge, service provision support, and digital 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. Exceptions to these use cases may be digital humanities or media archaeology researchers, where software reconstruction *is* the method of scholarly inquiry (Cardoso Llach & Donaldson, 2019).

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 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 exhibition 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 without having to set up the virtual machine, fire up and select the proper emulator, and initiate the intended software title.

Exhibitions for LAMs assume patrons from all backgrounds, all ages, and all sorts of personal computing experiences. 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. Exhibition users may have had personal experiences with the obsolete software on display or, just as likely, they may have never encountered the machine on display and indeed, that is why they are visiting the exhibit. 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 user's engagement with an immersive computing experience is more important than communicating what, how, and why emulation is occurring.

6 | DISCUSSION: THE WHEN OF EMULATION

For the preservationists I interviewed who were responsible for preservation and access workflows, emulation encounters are performances or events with software but communicating how emulation is achieved is a thorny problem of representation. Interviewees who were metadata specialists, catalogers, or processing archivists reported that these problems of representation primarily related to representing temporal expressions and layers of virtualization. Interviewees responsible for descriptive representation such as a managing archivist or a metadata librarian would regard the technological frames of the preservation practitioners and their intended users as incommensurable and as yet, unknown. While all encounters with emulation practices reveal multiple layers of time between the now, the past, and 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 using the software (as information and evidence) of a performance within a specific virtual environment.

For the preservationists I interviewed, this frame-within-a-frame-within-a-frame of temporal virtual perspectives (e.g., a contemporary hardware machine running a virtual machine of an historic operating system that is running a software emulator) presents a problem of reconciling layers of time as well as describing possible actions within such a virtualized environment. So, at the heart of emulation-driven access to software objects is what we might call "the when of emulation" for access experiences (Acker, 2020). Unlike accessing a book or providing access to a unique artifact in a collection, emulating software-dependent information involves many layers of abstraction, pacing, and moving between different layers of time. Moreover, these emulation techniques update a digital object in ways that materially transform its significant properties, creating questions about the meaning and precise representation of what an emulator is preserving when it is used. Becker has discussed the importance of this as a type of category error thusly: "if the data and the computation are replaceable, they cannot be significant. Instead, significance lies in the performance" (Becker, 2018, p. 23). If each emulation is a meaningfully different technological performance, and each emulator environment is nested within an assembled chain of virtual machines, operating systems, hardware and sometimes special peripherals, then how should preservationists represent the significance of each performance? Unlike other information service provisions found in libraries and archives, each user cannot reliably have the same access experience given the possibilities that emulation affords in the virtualization of software experiences that software provides. 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.

Interviewees who were responsible for describing and providing access to completed emulation projects speculated about how these dynamic "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 distinguished or described such as Apple's macOS beachball, or its predecessors the spinning pinwheel and the hourglass? If accessing an individual's software collection in a special collections library or a research archive, 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. The answers to many of these questions rely on making the hidden work of preservationists and emulators visible, because the user's encounter and unique performance changes as much as it informs. This is further complicated by the reality of software's backgrounding qualities and that emulators, too, are software.

Interviewees who were responsible for systems administration, managing the institution's integrated library system and software development, 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 informants responsible for preparing emulators, 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. This process could take as long as 10 minutes. Those interviewees who were responsible for instruction, public services, or research support expressed concern that potential users would be discouraged by the time consumed in "getting to go." This finding matches earlier user studies evaluating the usability of emulators. Hedstrom et al. (2006) found that users preferred ease of use and speed of interaction more than the original look and feel of software. On the other hand, waiting a few minutes for an operating system to start or for an application to load is an accurate experience of using earlier eras of software. For example, I observed that waiting for a CD-ROM 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, interviewees who were preservationist users all observed that the time taken to start up the emulation environment was still less time than earlier computing eras with microcomputers, or with software stored on magnetic tape, floppy disks, or even optical discs. Importantly, the timeliness to start the emulator reveals another limit between those that live with the emulation software as maintainers and those users who use it as a utility for research access or in a brief exhibition encounter.

Many of the descriptive speculations from interviewees about "getting to go" and imagining users' needs (such as anticipating preferences for faster access or desired paths) illustrated 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 what one digital archivist described as, "the way back to the when of the emulation." Most interviewees that I spoke to about "getting to go" assumed that contemporary users' current computing expectations would conflict with earlier eras of technology before the perceived instant access of cloud-based storage, high-speed internet, apps, and mobile computing devices. It is clear that a more generalized vocabulary that addresses "the when of emulation" as unique performances 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.

7 | CONCLUSION

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 U.S. 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 experience of accessing emulated software in representing preserved software both as a utility for access as well as information and evidence found in the experience of using software emulation for its own sake.

We are still in the early days of emulation as a service provision in LAMs. This exploratory research project was motivated by asking questions about emulation practices and if their access outcomes were specified by the unique environs of each place observed. Here I have reported on field observations, interviews, 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 for objects and experiences. Overall, findings from each site revealed that among 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 in exhibitions of computing experiences. Each of the different emulation practices identified in this research feature different goals (and challenges) in representing what is being preserved when encountering emulation. 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 found in information institutions.

ACKNOWLEDGMENTS

This research was made possible by the Software Preservation Network and a grant from the Institute for Museum and Library Services, grant RE-95-17-0058-17.

The author is very grateful to the memory workers who allowed her to interview, observe, and participate in their valuable software preservation work. The author would like to thank three site champions and anonymous referees who contributed greatly to the interpretation of these findings.

ORCID

Amelia Acker https://orcid.org/0000-0001-5332-0487

REFERENCES

Acker, A. (2020). Emulation encounters: Software preservation in libraries, archives, and museums. *Proceedings of the Association for Information Science and Technology*, *57*(1), e279. https://doi.org/10.1002/pra2.279

Bailey, D. E., & Leonardi, P. M. (2015). *Technology choices: Why occupations differ in their embrace of new technology*. MIT Press.

Bearman, D. (1999). Reality and chimeras in the preservation of electronic records. *D-Lib Magazine*, *5*(4). http://www.dlib.org/dlib/april99/bearman/04bearman.html

Becker, C. (2018). Metaphors we Work by: Reframing digital objects, significant properties, and the design of digital preservation systems. *Archiv*, *85*, 6–37.

Blanchette, J. (2011). A material history of bits. *Journal of the American Society for Information Science and Technology*, *62*(6), 1042–1057. https://doi.org/10.1002/asi.21542

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*, (pp. 257–266). IEE. https://doi.org/10.1109/JCDL.2014.6970177

Borgman, C. L., Golshan, M. S., Sands, A. E., Wallis, J. C., Cummings, R. L., Darch, P. T., & Randles, B. M. (2016). Data management in the long tail: Science, software, and service. *International Journal of Digital Curation*, *11*(1), 128–149. https://doi.org/10.2218/ijdc.v11i1.428

Bowker, G. C., & Star, S. L. (2000). *Sorting things out: Classification and its consequences*. MIT Press.

Cardoso Llach, D., & Donaldson, S. (2019). An experimental archaeology of CAD. In J.-H. Lee (Ed.), *Computer-aided architectural design. "Hello, culture"* (pp. 105–119). Springer. https://doi.org/10.1007/978-981-13-8410-3_8

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 and 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*, *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

Ensmenger, N. (2014). When Good Software Goes Bad: The Surprising Durability of an Ephemeral Technology. *MICE (Mistakes, Ignorance, Contingency, and Error) Conference*, 1–16. http://homes.sice.indiana.edu/nensmeng/files/ensmenger-mice.pdf

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.

Granger, S. (2000). Emulation as a digital preservation strategy. *D-Lib Magazine*, *6*(10), 101045. https://doi.org/10.1045/october2000-granger

Guttenbrunner, M., & Rauber, A. (2012). Evaluating emulation and migration: Birds of a feather? In H.-H. Chen & G. Chowdhury (Eds.), *The outreach of digital libraries: A globalized resource network* (pp. 158–167). Springer. https://doi.org/10.1007/978-3-642-34752-8_22

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

Hedstrom, M., Lee, C., Olson, J., & Lampe, C. (2006). "The old version flickers more": Digital preservation from the user's perspective. *The American Archivist*, *69*(1), 159–187. https://doi.org/10.17723/aarc.69.1.1765364485n41800

Kelty, C. M., & Erickson, S. (2015). The durability of software. In I. Kaldrack & M. Leeker (Eds.), *There is no software, there are just services* (pp. 39–55). Meson Press. https://doi.org/10.14619/008

Kling, R. (1991). Cooperation, coordination and control in computer-supported work. *Association for Computing Machinery*, *34*, 125396. 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*, 27. https://doi.org/10.5334/dsj-2017-027

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/

Moulthrop, S., & Grigar, D. (2017). *Traversals: The use of preservation for early electronic writing*. MIT Press.

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

Orr, J. E. (2016). *Talking about machines: An ethnography of a modern job*. Cornell University Press.

Plantin, J.-C. (2019). Data cleaners for pristine datasets: Visibility and invisibility of data processors in social science. *Science, Technology, and Human Values*, *44*(1), 52–73. https://doi.org/10.1177/0162243918781268

Rechert, K., von Suchodoletz, D., & Welte, R. (2010). Emulation based services in digital preservation. *Proceedings of the 10th Annual Joint Conference on Digital Libraries*, 365–368. https://doi.org/10.1145/1816123.1816182

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

Rinehart, R., & Ippolito, J. (2014). *Re-collection: Art, new media, and social memory*. MIT Press.

Rothenberg, J. (1995). Ensuring the longevity of digital documents. *Scientific American*, *272*(1), 42–47.

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

Russell, A. L., & Vinsel, L. (2018). After innovation, turn to maintenance. *Technology and Culture*, *59*(1), 1–25.

Sim, S. E., Cohn, M. L., & Philip, K. (2009). The work of software development as an assemblage of computational practice. *ICSE Workshop on Cooperative and Human Aspects on Software Engineering*, *2009*, 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*, *8*(1), 9–30. https://doi.org/10.1023/A:1008651105359

Strauss, A., & Corbin, J. M. (1997). *Grounded theory in practice*. SAGE.

Swade, D. (2003). Virtual objects: The end of the real? *Interdisciplinary Science Reviews*, *28*(4), 273–279. https://doi.org/10.1179/030801803225008686

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

Thibodeau, K. (2002). Overview of technological approaches to digital preservation and challenges in coming years. In *The state of digital preservation: An international perspective* (pp. 4–31). Council on Library and Information Resources. https://www.clir.org/pubs/reports/pub107/

Thomer, A. K., Twidale, M. B., & Yoder, M. J. (2018). Transforming taxonomic Interfaces: “Arm's length” cooperative work and the maintenance of a long-lived classification system. *Proceedings of the ACM on Human-Computer Interaction*, *2*(CSCW), *173*, 1–173. https://doi.org/10.1145/3274442

Traweek, S. (1992). *Beamtimes and lifetimes: The world of high energy physicists* (revised ed.). Harvard University Press.

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

von Suchodoletz, D., Rechert, K., & Valizada, I. (2013). *Towards emulation-as-a-service: Cloud services for versatile digital object access*. IJDC. https://doi.org/10.2218/ijdc.v8i1.250

von Suchodoletz, D., & van der Hoeven, J. (2009). Emulation: From digital artefact to remotely rendered environments. *International Journal of Digital Curation*, *4*(3), 146–155. https://doi.org/10.2218/ijdc.v4i3.118

Waterton, C. (2010). Experimenting with the archive: STS-ers as analysts and co-constructors of databases and other archival forms. *Science, Technology, and Human Values*, *35*(5), 645–676. https://doi.org/10.1177/0162243909340265

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

How to cite this article: Acker A. Emulation practices for software preservation in libraries, archives, and museums. *J Assoc Inf Sci Technol*. 2021;1–13. https://doi.org/10.1002/asi.24482