Curation as "Interoperability With the Future": Preserving Scholarly Research Software in Academic Libraries

Curation as "Interoperability With the Future": Preserving Scholarly Research Software in Academic Libraries

Alexandra Chassanoff*
Program on Information Science, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA, 02139. E-mail: achass@mit.edu

Micah Altman
Program on Information Science, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA, 02139. E-mail: escience@mit.edu

This article considers the problem of preserving research software within the wider realm of digital curation, academic research libraries, and the scholarly record. We conducted a pilot study to understand the ecosystem in which research software participates, and to identify significant characteristics that have high potential to support future scholarly practices. A set of topical curation dimensions were derived from the extant literature and applied to select cases of institutionally significant research software. This approach yields our main contribution, a curation model and decision framework for preserving research software as a scholarly object. The results of our study highlight the unique characteristics and challenges at play in building curation services in academic research libraries.

Introduction

Software plays an increasingly vital role in the scholarly record. A recent survey of software users at academic institutions found that a large majority of respondents (180 of 215, or 84%) were creating their own source code (Alnoamany & Borghi, 2018). Software is now a ubiquitous and critical, if often invisible, component of evidence-based research in most scientific disciplines—and current and emerging practices for validating and reproducing of the empirical results in scholarship generally require understanding of and/or access to the software used by the original researchers (NASEM, 2016). Moreover, newly funded initiatives like the Software Heritage Foundation, Software Sustainability Institute, and the Software Preservation Network suggest an alignment across communities of practice on the importance of treating software as a first-class research object to be collected, preserved, and made accessible.

Academic research libraries are well-positioned to help scholars with organization and management of born-digital research outputs. Thus far, existing work to articulate best practices, approaches, and innovations for working with digital research content have largely been driven by the scientific research community outside of the library space (Belhajjame et al., 2014; Stodden et al., 2016). Notably, these approaches do not incorporate digital preservation strategies for sustaining meaningful access to digital content over time. Frameworks are often modeled on simple, static, and self-contained conceptions of data rather than reflecting complex, dynamic, and networked systems such as software. Because computational work often depends on platforms or independent components to function, the change or failure of components can present significant risks to integrity, functionality, and usability of computational artifacts (Kaltman, Wardrip-Fruin, Lowood, & Caldwell, 2014; Laurenson, 2017). Despite consensus from the research science community that reproducible scholarship is a cornerstone of scientific progress, the extent to which related standards and policies are being implemented in practice has not been evaluated. How can research libraries support the use and reuse of research software, to ensure that modern research needs are met and persist over time?

*Corresponding author

Additional Supporting Information may be found in the online version of this article.

Received November 21, 2018; revised March 18, 2019; accepted April 14, 2019

 DOI: 10.1002/asi.24244

In this article, we aim to address this question by uniting and synthesizing ongoing work from the digital preservation sphere on complex digital objects (Dietrich & Adelstein, 2015; Thibodeau, 2002) with parallel research in research data management and scientific communities (National Academy of Sciences, National Academy of Engineering, and Institute of Medicine Committee on Ensuring the Utility and Integrity of Research Data in a Digital Age, 2009; Peng, Privette, Kearns, Ritchey, & Ansari, 2015). We introduce and describe a model and decision framework focused on six curation dimensions in support of preserving and providing access to software in research libraries. We use an ecosystem approach to provide the necessary holistic lens for analyzing and charting intersecting needs, uses, and, practices across different domains. Applying a cross-dimensional, cross-stakeholder perspective in our analysis allowed us to identify and model key curation criteria over the lifespan of research software, acknowledging both the technical competencies required to keep software "alive" and the lived, embedded social practices and needs of researchers and other stakeholders.

Background

Since the early 1990s, the persistence of scholarly digital information has been an area of concern for library and archives (Kenney & Personius, 1992; Rothenberg, 1999; Waters & Garrett, 1996). The maturation of scanning technologies led institutions to explore how digital technology could aid in preservation efforts. As an alternative to microfilm, digitally stored images could be easily shared across networks without compromising fidelity. Standardization of network protocols also helped institutions grow and share their collections of resources across domains, arguably laying the backbone for digital libraries (Arms, 2012).

Despite a shared interest in both interoperability and networked research environments, there were divergences in how scholars and practitioners came to understand and define the landscape. Writing in 1999, Borgman calls attention to the "competing visions" on digital libraries. She argues that researcher communities tend to view digital libraries as "providers of content," while libraries instead take a broader perspective as "services provided to the community" (Borgman, 1999, p. 231). As the growth in digital information continued to expand, there was an increasingly recognized lack of cohesion within university settings about departmental responsibilities and roles. In turn, it became difficult to prescribe collaborative and distributed activities for building shared infrastructure. Greenstein and Thorin (2002) describe the challenges of stewardship in scholarly research spaces with particular clarity:

> Data produced as a byproduct of research are a crucial part of the scientific record… Together, these information resources constitute invaluable university assets that are at risk of loss because it remains difficult to locate responsibility and capacity for their long-term maintenance in any one department or in a departmental collaboration. (p. 34)

Early digital library proponents recognized that digital information needed to be actively cared for, in order to remain accessible. The threat posed by technological obsolescence of original storage media was noted in the literature as a core challenge for future access (Cleveland, 1998; Hedstrom, 1997). However, a narrow focus on these threats has, at times, mired the development and implementation of preservation strategies. As such, the current state of the field is that there are no longitudinal data points that can be used to assess the effectiveness of different preservation actions.

The impetus to engage with digitally created artifacts at the time of their creation is not novel for practitioners. In fact, calls in the archival literature to consider the full "life history" of documents date back to 1940 (Brooks, 1940). More recently, attention to resource creation has emphasized the role that record creators can play, from actively participating in the construction of the archival record (Shilton & Srinivasan, 2007) to encouragement for data producers to create "archive-ready data" (Hedstrom, Niu, & Marz, 2008). Having principle investigators and project managers involved in good documentation practices at the onset of a project can greatly improve tertiary use and value (National Research Council, 1995). Curators are advised to communicate early on with record creators because creators may not know the value of their possessions and may not understand the degree to which these possessions face degradation and loss (Paradigm Project, 2006). At the same time, contrasting the practices of digital archiving with digital curation highlights the importance of archivists' interventions at the start of resource creation because "ignoring the front end of records creation is a recipe for submission information packages that are not worth ingesting" (Cunningham, 2008, p. 535).

The need for a comprehensive approach to the caretaking of digital research resources is recognized across a spectrum of scholarly communities (Boss & Broussard, 2017; Collberg & Proebsting, 2016; Lynch, 2017). Long-term use and value are most effectively supported through the application of quality metadata and documentation (Esanu, Davidson, Ross, & Anderson, 2004). However, the reuse of digital research resources in new contexts can still present difficulties. For example, the same data set can be used and interpreted by two different communities with entirely different purposes and outcomes (Borgman, Wallis, & Enyedy, 2007). Planning for these sorts of complexities is a critical aspect of developing sustainable research infrastructure that works across different domains, disciplines, and practices.

We conjecture that research libraries are well-positioned to lead or support curation actions across the entirety of the research software lifecycle. In parallel with changes to federal funding requirements, libraries have been at the forefront of emerging efforts to build capacity for research data management (RDM) services (Tenopir, Sandusky,

DOI: 10.1002/asi

Allard, & Birch, 2014; Cox & Corrall, 2013). Despite the relatively new phenomenon of capturing software (and related outputs) as part of the scholarly record, libraries have a long history of collecting, preserving, and providing access to resources for the pursuit of knowledge (Norton, 1854). They qualify as sites of social infrastructure, a necessary corollary for endurable and persistent information structures (Paskin, 2002). Library interventions at the stage of resource creation have been shown to result in positive outcomes, including: producing new models for scholarly publishing and open access (Crow, 2002; Bonn & Furlough, 2015), establishing new roles for librarians doing outreach (ARL, 2009; Maron & Smith, 2009), increasing recognition of value for inside-out collection approaches (Dempsey, 2016), and highlighting the importance of incorporating diverse perspectives directly into collection-building (Sadler & Bourg, 2015).

An Ecosystem Approach to Modeling Software Curation

Our research design was iterative-inductive, working across multiple research phases to characterize and refine independent dimensions of a curation model and decision framework. We employed an ecosystem approach to model research software as a scholarly object to be collected, preserved, and made accessible. An ecosystem approach emphasizes "the material interdependencies among the group of organisms which form a community and the relevant physical features of the setting in which they are found, and the scientific task becomes one of investigating the internal dynamics of such systems and the ways in which they develop and change" (Geertz, 1963, p. 3). For this analysis, we adopted an inclusive definition of research software that includes software used as an object of research (as a direct object, as in software engineering; or as an instrumental object that affects humans and society, as in anthropology); software used by researchers to collect, interpret, process, or analyze data; and software produced by researchers to embody theories, models, or methods (see Hong, Crouch, Hettrick, Parkinson, & Shreeve, 2010 for a review of common software uses in research).

The central question guiding our work was: What significant ecosystem characteristics support long-term access to and use of research software for different communities of practice? Our approach is grounded in Moore's theory of digital preservation, whereby effective preservation environments "validate communication from the past … while communicating with the future" (Moore, 2008, p. 64). From this vantage point, curation is an ongoing, continuous process of caretaking that maintains significance across time and space for different communities. To our knowledge, the practices of software curation are largely undefined in the literature. In this research, we chose to define software curation as follows: the active caretaking practices to support the meaningful creation, use, and reuse of software as a research object.

Identifying Curation Dimensions

To identify relevant characteristics for modeling software curation, we first conducted an environmental scan and literature review in the following areas: research data management, game preservation, software and code studies, digital and data curation, and digital preservation. Our review was scoped to focus on research related to academic libraries. Thus, topics like software engineering or rights management were considered out of scope. In our analysis, we reviewed both empirical studies and practitioner guidelines to surface common topical areas. We applied a sociomaterial lens to broaden our analysis and understanding of intersecting drivers that influence curation actions throughout the research software lifecycle, adopting different stakeholder perspectives across domains and disciplines. Sociomaterial perspectives are well-suited for research that seeks "to make visible the material dynamics in practice situations—the relationships among bodies, tools, technologies, and settings—as well as human intentions, expertise, and communication" (Fenwick & Nimmo, 2015, p. 67). Crucially, we wanted to disregard "established, competing perspectives that privilege either a technological or a human-centric understanding of social phenomena" (Harris & Abedin, 2016, p. 8).

Following this analysis, we identified six focal areas—which we label *curation dimensions*—for research libraries developing services targeting software preservation. Each dimension broadly characterizes an area of attention for institutions to focus curation actions to improve preservation-readiness. We list these dimensions in alphabetical order in Table 1, and include the driving questions from the literature that provided grounding for each dimension, along with examples of potential values. We go on to summarize ways that libraries can support and facilitate curation services for preserving and providing access to research software.

Activities. While forecasting future uses of software is beyond the scope of this research, curation actions and preservation strategies should be grounded in understanding the potential activities different communities of practice might want to do with research software. Examples of potential activities that involve software include: publishing software with appropriate rights and licensing (Morin, Urban, & Sliz, 2012); using software to assist in qualitative analysis (Johnston, 2006); and reusing software to validate or verify previous research results (Miranda & Bertolino, 2017). Envisioning this range of activities can help articulate important relationships between entities that should be preserved. For example, imagine researcher Alice publishes new analysis of a data set using research software developed and maintained locally at her institution. Depending on the institutional collecting policy, curation actions that

DOI: 10.1002/asi

TABLE 1. Modeling software curation.

Dimension	Driving question	Example values
Activities	*What potential activities might designated communities want to do with software?*	Aggregate, analyze, cite, create, deposit, migrate, transform, publish, version, reuse
Boundary conditions	*What characteristics of the software experience emphasize the transmission of information or assist with its comprehension?*	Renderability; file tree navigation; contextual metadata; installability; rights management
Carriers	*In what file and media formats are the resources of interest instantiated? What risks to future understanding are posed by these formats?*	(Media) Removable magnetic media; LTO-9 tape; Paper printout: dot-matrix/ASCII; cloud-storage: S3-bucket (File) Java bytecode v1.2; Fortran 90 source; PDF/A-3
Documentation	*What existing information documents design choices, intended uses, and methods of operations—and how can these be used to support choices made by curators or end-users?*	Readme files; metadata; codebooks; methodology; scripts; correspondence
Purpose(s)	*Was there a specific intended task gap that needed to be filled?*	to validate or test existing claims; to generate a research outcome; to document research process
Scenario(s)	*What potential future scenarios could support each desired activity? Who are the stakeholders in each scenario, and how/why do they interact?*	Cross-cultural heritage artifact; amateur hobbyist project; legal evidence; part of a virtual experience; as a research tool;

facilitate future usability of that software might become a significant driver for preservation strategies. There are also actions that can improve or facilitate the activity of software reuse. For example, a number of recent studies suggest that standardization of software description encourages reuse (Altman, Borgman, Crosas, & Matone, 2015; Li, Greenberg, & Lin, 2016).

Boundary conditions. Researchers have different needs related to how they use research software, and software itself requires specific conditions to be functional. In this research, we use the term "boundary conditions" to refer to the significant attributes that characterize the research software experience. This term builds on the concept of essence from the digital preservation literature, defined as "a way of providing a formal mechanism for determining the characteristics that must be preserved for the record to maintain its meaning over time" (Heslop, Davis, & Wilson, 2002, p. 13). Boundary conditions document and communicate baseline functional behavior for research software, encouraging greater transparency and accountability (Gebru et al., 2018). At the same time, libraries must plan for the instability of fixed meanings for interactive content. All user interactions will generate inherently variable experiences because "the form or narrative of the work may only develop through incremental actions by users" (Abbott, 2012, p 62).

Carriers. Overall, standards for acquiring, processing, and describing born-digital materials have been relatively slow to develop, resulting in a high backlog of legacy media at collecting institutions. Processing legacy materials on different carriers can be challenging, due to a variety of technical factors, including: format obsolescence (Singh, 2009), hardware and/or software component failure (Rosenthal, Robertson, Lipkis, Reich, & Morabito, 2005), or the risk of alteration to original materials (Woods, Lee, & Garfinkel, 2011). Legacy media can also present challenges to traditional modes of archival processing, in which parallel arrangement and description of materials is encouraged to ensure archival integrity and provenance. Original media sources are not always accessed at the time of accessioning, due to resource shortages. While preservation strategies and actions depend on the object(s) being preserved, efforts to record software components (and relationships among them) should be robust and thorough (Hedstrom & Lee, 2002; Matthews, Shaon, Bicarregui, & Jones, 2010). A promising development for digital preservation has been the modification of the PREMIS data model, which was recently adapted to record representation information about computing environments (Dappert & Farquhar, 2009).

Documentation. Collecting institutions have long recognized the important role that documentation plays in maintaining value for research objects. The US National Archives and Records Administration (NARA) first began recognizing machine-readable media as records in 1968. During the 1970s, NARA began to formally request that systems documentation accompany transfers of machine-readable records. Collecting software documentation provided essential context and could encompass "a wide range of evidential sources" including design specifications, tutorials, and even films of systems in use (Bearman, 1989). To accommodate the specifics of electronic records, the National Archives and Records Service developed two standardized inventory forms, GSA 7036 and GSA 7091, to record information about the media itself as well the use of content and potential restrictions.

More recently, Sköld (2018, p. 140) notes that video game preservation literature advocates collecting documentation as a means of capturing "the structure of social life and meaning-making" for communities of practice. Indeed, computer hobbyists and computational media scholars frequently use such documentation to reconstruct or remix software (Cover, 2013; Monroy-Hernandez, 2012). Because managing documentation can be time-consuming, research libraries should develop collection policies that

DOI: 10.1002/asi

include appraisal decisions about the kinds of materials the institution wants to retain.

In many cases, the software being curated is "legacy" software—software that is no longer currently maintained or developed. Such software may be still operational (although outdated), may no longer be executable on legacy systems, or may be inaccessible. In cases where legacy software itself is not accessible, documentation that describes software architecture or functionality can be sufficient context for scholars. As an example, consider the multiple implementations of the software program Eliza, built originally by MIT scientist Joseph Weizenbaum in the 1960s and widely considered to be the first "chatterbot." The original source code has never been available so Eliza implementations are necessarily based on software descriptions from a paper Weizenbaum published (Chassanoff, 2018). For software stored on inaccessible or decayed media, accompanying documentation in the form of source code printouts or even previously videotaped interactions, can arguably provide scholars with sufficient information about expected functionality.

Purpose(s). Determining why research software was created and/or used can provide valuable context for future use. (Bearman, 1987). The range of purposes for software creation and use include: to validate research results (Hwang, Fish, Soito, Smith, & Kellogg, 2017), to investigate and fix source code bugs (Abdalkareem, Shihab, & Rilling, 2017), and to increase efficiency (Banker & Kauffman, 1991). In ideal cases, conversations can be conducted with resource creators to gather crucial information about the intended purpose and functionality of the software. Curation actions that capture the original purpose of creation can map easily to emerging standards for provenance metadata (for a review, see Cheney, Chong, Foster, Seltzer, & Vansummeren, 2009), defined as "a record that describes entities and processes involved in producing and delivering or otherwise influencing that resource" (Gil et al., 2010, section 2.1).

Scenario(s). In a typical research library setting, multiple stakeholders may be involved in the creation, acquisition, use, description, preservation, or reuse of software, each having distinct goals and motivations. Considering potential scenarios for software creation, use and reuse from the perspective of multiple stakeholders is an effective mechanism for mapping critical practices in the ecosystem. From this vantage point, one can easily imagine that different communities of practice have vastly different curation needs since "significance is in the eye of the stakeholder" (Dappert & Farquhar, 2009, p. 1). For example, researchers studying the history of software might be interested in viewing original documentation about a specific type of software; specifically, how it was developed or created, how it was intended to be used, who developed it and why. On the other hand, digital archivists acquiring software into their digital preservation system might be primarily interested in ensuring the integrity and trustworthiness of the object(s). Linking explicit needs to communities of practice through establishment of common scenarios can help guide and prioritize curation actions.

Case Study Design and Application

A case study approach was chosen to iteratively refine the model. First, we investigated an individual case, worked to describe the case in terms of the currently proposed dimensions, and then evaluated how well the applied dimensions served to guide preservation actions for future use. We then applied the final set of dimensions to the entire group of cases to identify remaining gaps and to understand and characterize patterns across all of the cases.

The case setting for this study was the Massachusetts Institute of Technology (MIT), a research institution rich with both technological development and technological histories. Since the 1940s, the Institute has excelled in the creation and production of software and software-based artifacts. Project Whirlwind, Sketchpad, and Project MAC are just a few of the monumental research computing projects with origins at MIT.

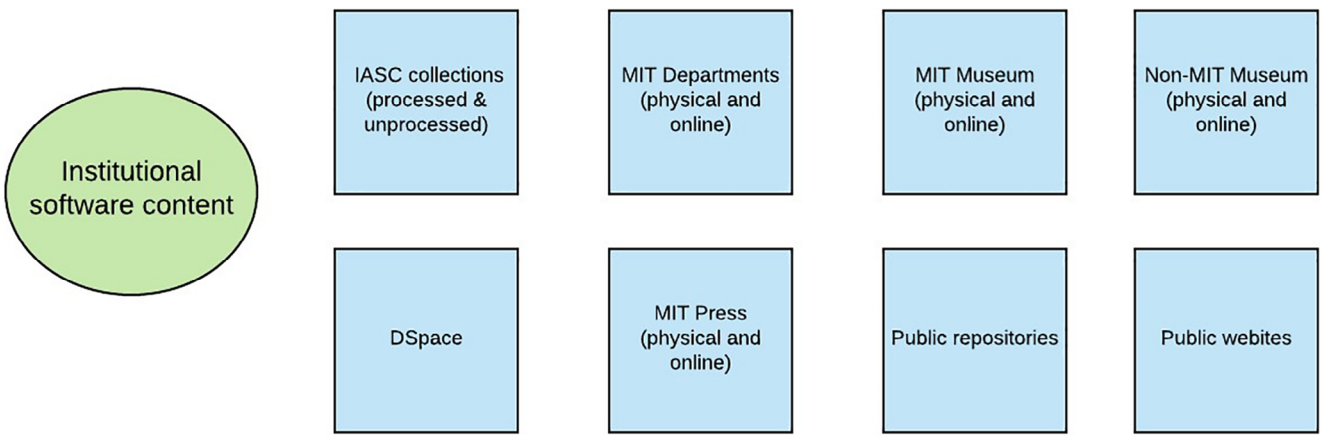

FIG. 1. Where does software live at MIT? [Color figure can be viewed at wileyonlinelibrary.com]

DOI: 10.1002/asi

As an initial step, we conducted an informal survey of different types of research software and their corresponding locations (see Figure 1) across MIT's campus. The purpose of this activity was to broadly identify the kinds of content that a research library might be interested in preserving. In our survey, we located research software content spread across different institutional and physical domains—faculty offices, department closets, webpages, and cloud servers were just a few of the many locations where we identified research software. For example, the Institute Archives & Special Collections (IASC) has over time acquired significant software and related content (for instance, project notebooks, printouts of source code) stored in their manuscript collections.

Following completion of the software survey, we began to formulate prototypical cases that described software curation scenarios encountered in research libraries. To develop these scenarios, we collected and analyzed three sources of data:

- a disciplinary literature review and environmental scan;
- compilation of software-related inquiries captured by the MIT Libraries' Data Management Services (DMS) group;
- compilation of software-related projects at MIT with contacts and locations, where possible.

We began to iteratively develop a baseline instrument for conducting case study research, using the previous analysis to inform the scope and range of research questions. We also began to develop guidance documents and templates for gathering critical information about software creation, use, and reuse practices. For example, we drafted a *Software Intake Form* for content curators to record acquisition, documentation, and transfer information related to software (see Appendix A1). We then mapped representative scenarios (for instance, "Faculty member creates software-driven artwork/publication") to identified examples at MIT, resulting in 12 potential cases. We reviewed and evaluated each case in terms of four criteria, which we assessed to be capable indicators of "information-rich" cases (Patton, 1990). Below is the evaluation matrix we used for case selection (see Table 2).

Both researchers evaluated the full list according to the criteria and rationale specified in the evaluation matrix, ultimately selecting five cases for further analysis.

TABLE 2. Evaluation matrix.

Evaluation criteria	Rationale
Representativeness	Case is likely found at other institutions
Institutional importance	Case is likely of interest to MIT Libraries; would want to acquire, preserve, and make this content accessible over the long term
Diversity	Case provides ample representation of different types and locations of software
Participation	Department/faculty/grad student is willing to participate in this research

TABLE 3. Linking representative scenarios with selected cases.

Prototypical scenario	Case (department/software)
Absent creator developed software now housed on legacy media in archival collections	MIT Institute Archives & Special Collections (IASC), *GRAPPLE* software
Faculty/grad students/department generates software-driven scholarly work	MIT Comparative Media Studies, *Autofolio Babel*
Faculty/grad students/department develops software iteratively with updates	MIT Computer Science and Artificial Intelligence Laboratory (CSAIL), *Genesis* software
Faculty/grad students/affiliate maintains widely used open source software components for preservation	MIT Crystallography Department, *SHELX*
Library staff is retiring locally-created, community-supported software	MIT Libraries, *DSpace codebase*

To test the applicability of the model and the identified curation dimensions, we conducted semistructured interviews with participants from five cases (see Table 3).

Each interview was recorded and transcribed. We opted to keep memos as a means of documenting decision-making or to note inconsistencies and areas for follow up. We used interviews as feedback mechanisms to continually refine the instruments and guidance templates. Following the completion of interviews, we created software curation profiles (SCPs) for each case describing significant software characteristics and outlining pathways for improving preservation-readiness (see Appendix A2 for an example of a completed SCP). We also developed a template that research libraries can use to assess institutional needs, evaluate capacity, and identify potential pain points in implementing services for research software curation and preservation (see Appendix A3).

Together, this corpus of cases includes exemplars of common software curation scenarios; provides considerable variation across the curation dimensions; and spans a broad range of potential software uses. Although not intended to be statistically representative, the corpus presents a broad spectra of curation challenges. We conjecture that if a model is able to identify the critical curation decisions across this corpus, it is likely to be successful when applied to other cases.

Case Analysis and Discussion

The GRAPPLE Software

Our first case analysis focused on materials in the J.C.R. Licklider collection, initially acquired in 1996 and currently housed at the IASC (J.C.R. Licklider Papers, 1938-1995). Licklider, widely hailed as an influential figure for his visionary ideas around personal computing and human–computer interaction, first came to MIT in 1950. The Licklider collection was initially suggested by an IASC archivist familiar with legacy software in different

DOI: 10.1002/asi

archival collections. We focused attention on GRAPPLE, a dynamic graphical programming system developed while Licklider was at the MIT Laboratory for Computer Science. The GRAPPLE software offered a rich example of legacy software of significant institutional and historical interest.

In our case analysis, we first documented all of the materials in the collection in an Excel spreadsheet, noting format, box location, and folder where applicable. Materials included paper printouts of source code, interim user manuals, technical reports, project correspondence, and multiple undated, unidentified computer tapes. Many documents had multiple versions, typically distinguished by date and filename. The printouts of source code printouts usually totaled around 40 pages. The computer tapes have not yet been formatted for access.

Applying curation dimensions to GRAPPLE. While the software itself had not been formatted for access, the **documentation** contained in the collection provided substantial information about the software. Analysis of the user manual, technical reports, and source code printouts provides sufficient context for a scholarly understanding of the functionality and purpose of GRAPPLE. Other collection materials offer interesting conceptions of personal computing while also providing clear evidence that computer scientists such as Licklider regarded abstraction as an essential part of successful computer design. A pamphlet entitled "User Friendliness—And All That" notes the "problem" of mediating between "immediate end users" and "professional computer people" to successfully aid in a "reductionist understanding of computers." This exemplifies the inherent paradox of curating and preserving born-digital materials like software: digital objects require active caretaking and maintenance to persist, but are functionally designed to make technical logic invisible to users (Chun, 2005). Enabling access and discovery points for the collection can begin with generating digitized, machine-readable output of documentation using optical character recognition (OCR), making it easily indexed and available for discovery.

According to the project's final technical report, the **purpose** of the GRAPPLE project was "to explore the use of computer graphics in preparing programs and in monitoring the interpretation or execution of programs" (Licklider, 1988, p. 3) The report goes on to describe GRAPPLE's intended functionality: providing a historical perspective into how computing documentation and instructions for programs operated at the time. For example, there is a verbose description of the different software functions and how a user might "move" through a typical setup. A short user manual was built into the software itself.

Considering the range of different **scenarios** for accessing the collection materials helps to illuminate the unique cultural and historical value of the software. Historians of programming languages might be interested in studying the evolution of the coding syntax contained in the GRAPPLE collection. The team used the now-defunct programming language MDL (which stands for "More Datatypes than Lisp"); numerous examples of MDL in action can be found through printouts of code packages. Hobbyists and vintage software enthusiasts could potentially reconstruct aspects of the software.

Much of the documentation included in the collection describes what the ideal user and environment for GRAPPLE are—specifying these **boundary conditions** provides important future benchmarks for evaluating a preserved system. For example, excerpts from an original project technical report included in the GRAPPLE collection describe how an interaction with GRAPPLE should take place:

> It exists as source language files and as what in MDL is called a file of partly executable, partly interpretable code. We assume that a programmer will carry out a multi-session programming project with a save file and its descendants. At the beginning of the first session, the programmer loads the initial save file, say grapple.save, into a Digital Equipment Corporation VAX computer (with BBN Computer Co. BitGraph terminal) operating under the Berkeley Unix 4.2 or 4.3, by typing mudsub grapple.save and then a line-feed. (Although both programs can deal with both cases, Unix mainly uses lowercase letters and MDL uses mainly uppercase letters.) Then, when the character string "RESTORED" appears on the screen, the MDL interpreter is running. The programmer starts GRAPPLE by typing <G> and a character that we shall call the *DO-IT* character. For the VAX computer with BitGraph terminal, the *DO-IT* character is the *line-feed* character (Licklider, 1988, p.19).

Such a vivid description of the interaction, along with describing intended functionality, provides important details about what the software should achieve in ideal operating conditions, and the conditions for its use.

Part of the GRAPPLE collection exists on magnetic media, a storage **carrier** that popularized in the 1960s. Despite nearly 60 years of existing in archival collections, an accurate count of legacy media, formats, and their conditions in archival institutions is unknown.

The different **activities** that one might use GRAPPLE for would be interesting to revisit today, considering the implementation challenges discussed by the GRAPPLE team at the time of coding and development. One obstacle to successful implementation noted by the team at the time of development was the limited graphical display environments. In their final project technical report from 1984, the GRAPPLE team described the potential of desktop icons for identifying objects and their representational qualities.

> Our conclusion is that icons have very significant potential advantages over symbols but that a large investment in learning is required of each person who would try to exploit the advantages fully. As a practical matter, symbols that people already know are going to win out in the short term over

DOI: 10.1002/asi

> icons that people have to learn in applications that require more than a few hundred identifiers. Eventually, new generations of users will come along and learn iconic languages instead of or in addition to symbolic languages, and the intrinsic advantages of icons as identifiers (including even dynamic or kinematic icons) will be exploited. (Licklider, 1988, p.17)

Despite advances in technology, fundamental dynamics in the study of human–computer interaction remain relatively unchanged. Conducting a historical analysis of GRAPPLE using the assembled documentation shows evidence of a longstanding powerful relationship between representational symbols and the production of knowledge. What might it look like to bring to life today software that was conceived in the early days of personal computing, and what can that tell us about the relationship between humans, computers, and the interactions between and among them?

Devising Curation Strategies for Legacy Software

Below we use our case description and analysis of GRAPPLE to make recommendations for curation strategies that research libraries can adopt in order to support preservation and access goals for legacy software.

Identify appraisal criteria. Establishing appraisal criteria is an important first step that can be used to guide decisions about the selection of relevant materials for long-term access and retention. For hybrid collections that contain legacy software, determining appraisal criteria will require decision-making about the desired level of access and preservation to materials. What components of the collection should be made accessible? Does the software itself need to be executable? Making these decisions at the institutional level can guide the identification of appropriate preservation strategies (for instance, emulation, migration) based on the desired outcomes.

Assemble relevant materials. A significant challenge with legacy software lies in the gathering, identification, and overall assembling of relevant materials to provide necessary context for meaningful access and use. Locating and inventorying related materials (for instance, memos, technical requirements, user manuals) is an initial starting point. In some cases, meaningful materials may be spread across the web at different locations. While it remains a controversial method in archival practice, a documentation strategy may provide useful framing and guidance. MIT archivist Helen Samuels first introduced the idea, which treats archival practice as collaborative work among record creators, archivists, and users. Documentation strategies are an "analysis of the universe to be documented … and the formulation of a plan to ensure the adequate documentation of an ongoing issue or activity or geographic area" (Samuels, 1991, p. 126).

Identify stakeholders. Identifying the various stakeholders, either inside or outside an institution, can help to ensure proper transfer and long-term care of legacy software, along with managing potential rights issues or confirming expected functionality. Here we draw on Carlson's (2010) work developing the Data Curation Profile Toolkit and define stakeholders as any group, organizations, individuals, or others having an investment in the software that you would feel the need to consult regarding access, care, use, and reuse of the software.

Describe and catalog materials. Preservation-readiness can be increased by thoroughly describing and cataloging selected materials, with an emphasis on capturing relationships among entities. In some cases, this may consist of describing aspects of the computing environment including hardware, software, file systems, libraries, and versioning. Depending on the media format, automated tools that extract and document dependencies can prove essential. In some cases where the software itself may not be accessible, describing related materials (that is, printouts of source code, technical requirements documentation) adequately can provide important points of access and enhance discoverability for collection materials.

Digitize and OCR paper materials. Paper printouts of source code and related documentation can be digitized according to established best practice workflows (Digital Library Federation, 2005). The use of OCR programs produces machine-readable output, enabling easy indexing of content to enhance discoverability and/or textual transcriptions. The latter option can make historical source code more portable for use in simulations or reconstructions of software.

Migrate media. Legacy software often resides on unstable media such as floppy disks or magnetic tape. The threats posed by unstable media are substantial, and have been long recognized in digital preservation (Hedstrom, 1997). Although there is a large body of research in the area of long-lived storage, economic and technical drivers remain barriers to the widespread adoption of durable media (Rosenthal et al., 2012). And despite a large body of good practice for conducting migrations, the migration of collections at scale and across technical platforms is a substantial challenge (Altman et al., 2014, section 4). In cases where access to the software itself is desirable, migrating and/or extracting media contents (where possible) to a more stable medium is recommended.

Other Cases

Previously, we described and analyzed the GRAPPLE legacy software by applying identified curation dimensions. In this section we summarize both the patterns and limitations we discovered in applying a parallel analysis to the selected cases in Table 3.

As described in our methodology section, the overall goal of our cross-case comparison was to explore,

DOI: 10.1002/asi

articulate, and characterize curation pathways for improving preservation-readiness of software in research library settings. By applying a qualitative case study analysis to each of the cases, we were able to identify the set of features that were most important to making curatorial decisions across the collection.

In earlier stages of this research, we generated and considered a larger candidate set of curation dimensions. We used this set of features to probe the extent to which the candidate set of curation dimensions were individually necessary to characterize each of the individual features of the collection. We also analyzed whether these dimensions were collectively sufficient to characterize all of the critical features of the entire collection of cases.

The analysis of other cases suggested that the initial candidate curation dimensions could be simplified. Although a number of dimensions were present in the literature, they were not needed to support curation decisions on the collection examined. For example, we ended up removing the "stakeholders" and "motivations" dimensions from our list because they did not significantly factor into curation and/or preservation criteria for three of our five cases. We also removed "functions" because of overall redundancy with other dimensions: In our five cases, characterizing software's function was not as significant as knowing the purpose of software or anticipating the different kinds of activities it could be used for.

At the same time, the relevancy and importance of the remaining curation dimensions was clarified through this examination of other cases. For example, envisioning motivating scenarios for software reuse helped to articulate potential curation actions for desired outcomes. In GRAPPLE's case as a legacy historical artifact, providing digitized access to software documentation might be more useful than provisioning access to the stored media. However, in the case of *Autofolio Babel*, the original software creator describes how the lived experience of the software should factor into preservation planning for future use.

> It's not really a software concern at this point, but rather a concern for a system that includes software. And having *Babel* as the software component work — that's more or less a subset. I wouldn't want someone to take video of this and put that video out as a 'preservation method. This needs to be a functioning computing machine for this to work, so the software preservation would be part of it from my standpoint (Chassanoff, 2018).

Finally, we used this examination of other cases to refine the terminology and definitions of the remaining dimensions. For example, "boundaries" was altered to "boundary conditions" to reflect the importance of characterizing software not just in terms of its constituent parts but also how it ideally functions in a given environment.

Recommendations

According to a recent content analysis of academic library websites, a large majority 185 (185 of ~282) research-intensive academics libraries are now offering RDM services (Yoon & Schultz, 2017). In contrast, only a few libraries provide explicit services for software curation and preservation. In part, this can be attributed to the conceptual, technical, and social challenges bound up in devising caretaking strategies in this sphere. Complexities run the gamut, from simply defining software as a preservation object to established methods optimizing the potential for reuse (Chassanoff, Borghi, AlNoamany, & Thornton, 2018).

We advocate for more collaborative approaches in research library settings among librarians, data managers, archivists, and technologists to design and architect scholarly infrastructure services that explicitly support the creation, use, reuse, and preservation of scholarship across the research lifecycle. For example, research libraries are well-positioned (but not always well-resourced) to provide services that encourage best practices for workflow planning and management of scholarly research. Hosting monthly workshops for researchers can help communities of practice stay abreast of emerging practices, projects, and developments. Developing and instituting training is another method for instituting active engagement practices with communities toward sustainable digital resource creation and preservation.

Software and software-driven artifacts are increasingly a critical aspect for the research community that libraries serve. Addressing software strategically serves an unmet institutional need and provides opportunities for collaborations with researchers early on in the process. Although libraries will need to invest in further expertise in software curation and preservation, the cost is incremental. The results will broaden the set of curation skills and infrastructure, which in turn makes librarian services more immediately useful and more likely to remain useful as researchers' engagement with digital objects evolve.

A Decision Framework for Improving Preservation-Readiness of Software

We now focus attention on implementation pathways for research libraries and other stewardship organizations to improve preservation-readiness of research software. Drawing from key digital preservation models and theories reviewed during the initial phase of research, we map curation dimensions onto specific phases in the stewardship of research software (see Table 4). Our analysis draws on previous work by Nancy McGovern (2012), who conceptualized a template-based approach to aid organizations in managing digital content types over their lifetime. In our adaptation, we focus on relevant decision points for research libraries that can facilitate and guide the implementation of services for software curation.

Software creation. Research software can be created and used for a variety of different purposes, from validating results in previous studies to generating new research outcomes. Customized training and engagement with software

DOI: 10.1002/asi

TABLE 4. Key decisions in software curation.

Research software stewardship phase	Dimensions	Decision points for stewardship organizations
Software creation	Boundary conditions; Carriers; Documentation; Purposes	Who are the key stakeholders who create, use, provide access to and preserve research software at your institution? What best practice training and workshops can libraries offer to aid in creating digital resources that are preservation-ready?
Software selection and appraisal	Activities; Boundary conditions; Purposes; Scenarios	What kinds of software does your institution want to collect, preserve, and provide access to? What copyright/intellectual property issues are associated with collecting this software? Is there specific language that should be included in deposit agreements?
Software acquisition and ingest	Activities; Boundary Conditions; Carriers; Documentation	What changes need to be made to existing workflow to acquire software? Where will acquired materials be stored pre-ingest? What kinds of quality assurance should be done after acquisition? What precautions need to be made for handling obsolete/at-risk media?
Software description and access	Activities; Documentation; Purposes; Scenarios	How should existing software in hybrid collections be described? What metadata schemas and standards are best suited for describing software? How will sensitive/copyright materials be flagged?
Software preservation and storage	Boundary conditions; Carriers; Purposes; Scenarios	What preservation strategies best fit institutional commitment to software (e.g., migration, emulation, normalization, archival storage). What components of software do we want to preserve? Should original media carriers be preserved? Are there different priority levels that can be assigned based on media risk?

creators at this stage of research can be a useful intervention point for teaching research data management literacies. On the curation side, knowing the purpose of the original software can be very helpful in understanding how it can be used. In ideal cases, conversations can be conducted with software creators and/or primary user groups to gather this information. Broadly characterizing the different activities that research software can be used for helps curators determine priorities for what components should be preserved and made accessible. For example, if a primary purpose for the creation of software is to generate new research outcomes, then curation strategies should prioritize enabling software functionality over time. In terms of training, data management workshops can include the importance of creating well-documented "readme" files at the time of software creation. Workshops can also offer training in software unit testing and best practices for data management workflows.

Software selection and appraisal. Bearman (1987) notes, "Framing a software collecting policy begins with the definition of a schema which adequately depicts the universe of software in which the collection is to be a subset" (p. 16). Inventorying all potential materials of interest with some relationship to the software is one strategy. Does the binary code need to be retained? Is the research software linked to significant publications? Does software need to be executable in order to be useful? Answers to these questions can in turn be used to frame decision-making around the types of components to be collected, preserved, and made accessible. Collecting decisions can also be guided by foregrounding particular scholarly needs from a purposeful curation perspective, which emphasizes preservation outcomes in "combinations and states other scholars would find them most useful" (Palmer, Weber, Renear, & Muñoz, 2013, sec. Foundations…, unpaginated). In other words, what kinds of research software are institutionally-significant? What characterizes meaningful access and use/reuse for institutional research needs?

Software acquisition and ingest. Institutions acquiring software will need to be aware of the specific challenges associated with transfer and extraction of content from different types of storage carriers. Working with software that exists in binary format on a website will require a different set of skills and processes than reformatting or simply trying to access legacy software on magnetic media. What changes need to be made to the existing acquisition and ingest workflows in order to accommodate the different storage options for software? What kinds of quality assurance should be done to ensure that software has been fully acquired? What precautions need to be made for handling obsolete or at-risk media? Institutions should develop procedures for evaluating media carriers at the time of acquisition, to perform necessary transformations for unstable content.

Software description and access. Research librarians can also provide assistance by sharing emerging best practices aimed at addressing common researcher activities. For example, librarians can instruct faculty and staff on particular methods for improving source code discoverability, such as applying digital object identifiers (DOIs; GitHub, 2016) or submitting their software to trustworthy repositories that assign permanent DOIs, like Zenodo. To describe software in research collections, guidance can be drawn from Force 11's Software Citation principles (Smith, Katz, Niemeyer, & Force11 Software Citation Working Group, 2016). Considering multiple points of access through rich description will improve software discoverability. Addressing legacy software in hybrid collections or backlogs may also require the development of new protocols

DOI: 10.1002/asi

for description and arrangement; depending on existing workflows and systems, integrating previously unprocessed media into existing collections can introduce synchronization challenges (Prael, 2018).

Software preservation and storage. Successfully collecting, preserving, and providing access to software as a research object will likely require significant policy and procedural development for research libraries. Beyond copyright and building capacity for technical challenges, institutions will need to understand and plan for the range of media carriers and resulting risk factors that are introduced. Maintenance activities like scheduled refreshments or migrations will be essential to ensure software persistence and integrity. At the same time, assessing the significant aspects of a collection can be an important directive for establishing priorities about preservation. End users desiring authentic experiences with particular research software may request access to emulation environments, prioritizing services for interacting with original hardware and software. Other preservation requirements may consist of storing legacy media off-site, while still retaining important documentation that provides historical information about software creation and use.

Summary

Born-digital materials are inherently complex and represent challenges for collecting institutions. Changes in form and format have vast implications for the preservation of scholarship, requiring different tools, technologies, and workflows. Our research modeled software curation as the active caretaking practices to support the meaningful creation, use, and reuse of software. Using an ecosystems approach, we frame research software stewardship within a larger context of intersecting sociomaterial needs, uses, and practices. While preservation criteria may change according to different stakeholder needs, we identified six topical areas of curation that can improve preservation-readiness across different phases of research software stewardship.

This study identifies a host of future areas of research with the potential for high-impact outcomes, including methods for extracting legacy content, development of digital preservation metadata and associated workflows, and understanding what different communities of practice require when reusing born-digital content. Further research is needed to test the proposed curation model we have developed, although we have begun pilot testing templates with recruited practitioners working on software preservation. At the same time, what defines the material boundaries of research software? How do institutions want to represent digital research objects for meaningful access and use? How should research libraries and archives approach and handle legacy media that has been stored separately from its original collection, sometimes untouched for years (or decades)? What caretaking strategies can support preservation goals? These open questions can serve as areas of future research.

Communities of practice have only recently begun to address these complexities and new approaches have been slow to emerge. The decision framework we have introduced provides a mechanism for research library staff to responsibly intervene at various points across the research software stewardship lifecycle, and provide guidance on curation actions that can be implemented. It also introduces a vision for shared collaboration and utilization of skill sets across domains in service to performing the "hard work of software history" (Lowood, 2013, p. 1). Effectively characterizing software in its dual role as both artifact and active producer of artifacts remains an essential piece of understanding and ensuring its complex value over time.

Acknowledgments

We describe contributions to the article using a standard taxonomy (see Allen L., Scott J., Brand A., Hlava M. Altman M. Publishing: Credit where credit is due. Nature, 2014;508[7496]:312–313). A.C. and M.A. provided the core formulation of the paper's goals and aims, and A.C. led the writing of the original article and revisions. Both authors contributed to conceptualization through additional ideas and through commentary, review, editing, and revision. This material is based upon work supported by the CLIR and the Sloan Foundation.

References

Abbott, D. (2012). Preserving interaction. In L. Konstantelos, J. Delve, D. Anderson, C. Billenness, D. Baker, & D. Dobreva (Eds.), The preservation of complex objects volume 2: Software art (pp. 61–70). Bristol, UK: JISC. Retrieved from http://radar.gsa.ac.uk/2806/1/pocos_vol_2_final_release[1].pdf.

Abdalkareem, R., Shihab, E., & Rilling, J. (2017). On code reuse from StackOverflow: An exploratory study on android apps. Information and Software Technology, 88, 148–158.

Alnoamany, Y., & Borghi, J. A. (2018). Towards computational reproducibility: Researcher perspectives on the use and sharing of software. PeerJ Computer Science, 6, e26727v1.

Altman, M., Bailey, J., Cariani, K., Corridan, J., Crabtree, J., Gallinger, M., & Lazorshak, B. (2014). National agenda for digital stewardship. Retrieved from National Digital Stewardship Alliance website: https://ndsa.org//national-agenda

Altman, M., Borgman, C., Crosas, M., & Matone, M. (2015). An introduction to the joint principles for data citation. Bulletin of the Association for Information Science and Technology, 41(3), 43–45.

ARL Digital Repository Issues Task Force (2009). The research library's role in digital repository services. Retrieved from Association of Research Libraries website: https://www.arl.org/storage/documents/publications/repository-services-report-jan09.pdf

Arms, W. Y. (2012). The 1990s: The formative years of digital libraries. Library Hi Tech, 30(4), 579–591.

Banker, R. D., & Kauffman, R. J. (1991). Reuse and productivity in integrated computer-aided software engineering: An empirical study. MIS Quarterly, 15(3), 375–401.

Bearman, D. (1987). Collecting software: A new challenge for archives & museums (Vol. 1). Archives & Museum Informatics. Technical Report. Retrieved from http://www.archimuse.com/publishing/bearman_col_soft.html

Bearman, D. (1989). The case for software as documentation. IASSIST Quarterly, 13(1), 18–23. Retrieved from https://iassistdata.org/sites/default/files/iqvol13n1.pdf.

Belhajjame, K., Zhao, J., Garijo, D., Hettne, K., Palma, R., Corcho, Ó., … & Goble, C. (2014). The research object suite of ontologies: Sharing and exchanging research data and methods on the open web. arXiv preprint arXiv:1401.4307.

Boon, M., & Furlough, M. (Eds.). (2015). Getting the word out: Academic libraries as scholarly publishers. Chicago: Association of College and Research Libraries, a division of the American Library Association, 2015.

Borgman, C. L. (1999). What are digital libraries? Competing visions. Information Processing and Management, 35, 227–243.

Borgman, C. L., Wallis, J. C., & Enyedy, N. (2007). Little science confronts the data deluge: Habitat ecology, embedded sensor networks, and digital libraries. International Journal on Digital Libraries, 7(1–2), 17–30.

Boss, K., & Broussard, M. (2017). Challenges of archiving and preserving born-digital news applications. IFLA Journal, 43(2), 150–157.

Brooks, P. (1940). The selection of records for preservation. The American Archivist, 3(4), 221–234.

Carlson, J. (2010). The data curation profiles toolkit: Interviewer's manual. Retrieved from https://doi.org/10.5703/1288284315651.

Chassanoff, A. (2018). Scholar profile of Nick Montfort [Blog post]. Retrieved from https://informatics.mit.edu/blog/guest-post-scholar-profile-nick-montfort

Chassanoff, A., Borghi, J., AlNoamany, Y., & Thornton, K. (2018). Software curation in research libraries: Practice and promise. Journal of Librarianship and Scholarly Communication, 1(eP2239).

Cheney, J., Chong, S., Foster, N., Seltzer, M., & Vansummeren, S. (2009). Provenance: A future history. In Proceedings of the 24th ACM SIGPLAN Conference Companion on Object Oriented Programming Systems Languages and Applications (pp. 957–964). New York: ACM.

Chun, W. H. K. (2005). On software, or the persistence of visual knowledge. Grey Room, 18, 26–51.

Cleveland, G. (1998). Digital libraries: definitions, issues and challenges. IFLA, Universal dataflow and telecommunications core programme. Retrieved from https://archive.ifla.org/udt/op/udtop8/udt-op8.pdf

Collberg, C., & Proebsting, T. A. (2016). Repeatability in computer systems research. Communications of the ACM, 59(3), 62–69.

Cover, R. (2013). Reading the remix: Methods for researching and Analysing the interactive textuality of remix. M/C Journal, (4), 16. Retrieved from http://journal.media-culture.org.au/index.php/mcjournal/article/view/686.

Cox, A. M., & Corrall, S. (2013). Evolving academic library specialties. Journal of the American Society for Information Science and Technology, 64(8), 1526–1542.

Crow, R. (2002). The case for institutional repositories: A SPARC position paper. Retrieved from Association of Research Libraries website: https://rc.library.uta.edu/uta-ir/bitstream/handle/10106/24350/Case%20for%20IRs_SPARC.pdf

Cunningham, A. (2008). Digital curation/digital archiving: A view from the National Archives of Australia. The American Archivist, 71(2), 530–543.

Dappert, A., & Farquhar, A. (2009). Significance is in the eye of the stakeholder. In International Conference on Theory and Practice of Digital Libraries (pp. 297–308). Berlin, Heidelberg: Springer.

Dempsey, L. (2016). Library collections in the life of the user: two directions. LIBER Quarterly, 26(4), 338–359. http://doi.org/10.18352/lq.10170.

Dietrich, D., & Adelstein, F. (2015). Archival science, digital forensics, and new media art. Digital Investigation, 14, S137–S145.

Digital Library Federation. (2005). Technical guidelines for digitizing archival materials for electronic access: Creation of production master files–raster images. Washington, D.C.: Digital Library Federation. Retrieved from https://lccn.loc.gov/2005015382/.

Esanu, J., Davidson, J., Ross, S., & Anderson, W. (2004). Selection, appraisal, and retention of digital scientific data: Highlights of an ERPANET/CODATA workshop. Data Science Journal, 3, 227–232.

Fenwick, T., & Nimmo, G. R. (2015). Making visible what matters: Sociomaterial approaches for research and practice in healthcare education. In J. Cleveland & S. J. Durning (Eds.), Researching medical education (pp. 67–80). Hoboken, NJ: John Wiley & Sons.

Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J.W., Wallach, H., Daumeé III, H., & Crawford, K. (2018). Datasheets for datasets. arXiv preprint arXiv:1803.09010. Retrieved from https://arxiv.org/abs/1803.09010

Geertz, C. (1963). Agricultural involution: The processes of ecological change in Indonesia. Oakland, CA: University of California Press.

Gil, Y., Cheney, J. Groth, P., Hartig, O., Miles, S. Moreau, L., & Pinheiro da Silva, P. (2010). Provenance xg final report. W3C. Retrieved from http://www.w3.org/2005/Incubator/prov/XGR-prov-20101214/

GitHub Staff (2016). "Making your code citeable." GitHub Guides. Retrieved from https://guides.github.com/activities/citable-code

Greenstein, G. & Thorin E. (2002). The digital library: a biography. Digital Library Federation, Council on Library and Information Resources. Retrieved from https://www.clir.org/wp-content/uploads/sites/6/pub109.pdf

Harris, G. & Abedin, B. (2016). Participating or not participating? A sociomaterial perspective of the embeddedness of online communities in everyday life. Retrieved from https://arxiv.org/abs/1605.04714

Hedstrom, M. (1997). Digital preservation: a time bomb for digital libraries. Computers and the Humanities, 31(3), 189–202.

Hedstrom, M., & Lee, C. A. (2002). Significant properties of digital objects: Definitions, applications, implications. In Proceedings of the DLM-Forum (Vol. 200, pp. 218–227). Retrieved from https://ils.unc.edu/callee/sigprops_dlm2002.pdf.

Hedstrom, M., Niu, J., & Marz, K. (2008). Incentives for data producers to create "archive/ready" data: Implications for archives and records management. In Proceedings from the Society of American Archivists Research Forum. San Francisco, CA: Society of American Archivists.

Heslop, H., Davis, S., & Wilson, A. (2002). An approach to the preservation of digital records. Canberra: National Archives of Australia. Retrieved from https://trove.nla.gov.au/work/14930496

Hong, N.C., Crouch, S., Hettrick, S., Parkinson, T., & Shreeve, M. (2010). Software preservation benefits framework. Software Sustainability Institute Technical Report. Retrieved from https://www.research.ed.ac.uk/portal/files/1219870/SoftwarePreservationBenefitsFramework.pdf

Hwang, L., Fish, A., Soito, L., Smith, M., & Kellogg, L. H. (2017). Software and the scientist: Coding and citation practices in geodynamics. Earth and Space Science, 4(11), 670–680.

J.C.R. Licklider Papers (MC 499). (1938–1995). MIT Libraries, Cambridge, MA.

Johnston, L. (2006). Software and method: Reflections on teaching and using QSR NVivo in doctoral research. International Journal of Social Research Methodology, 9(5), 379–391.

Kaltman, E., Wardrip-Fruin, N., Lowood, H., & Caldwell, C. (2014). A unified approach to preserving cultural software objects and their development histories. Retrieved from https://escholarship.org/uc/item/0wg4w6b9

Kenney, A. & Personius. L.K. (1992). The Cornell/Xerox commission on preservation and access joint study in digital preservation report: Phase 1. Commission on Preservation and Access. Retrieved from https://eric.ed.gov/?id=ED352040

Laurenson, P. (2017). The lives of digital objects: A community of practice dialogue, Pericles Project. Retrieved from https://www.tate.org.uk/download/file/fid/108046

Li, K., Greenberg, J., & Lin, X. (2016). Software citation, reuse and metadata considerations: An exploratory study examining LAMMPS. In Proceedings of the 79th ASIS&T Annual Meeting: Creating knowledge, enhancing lives through Information & Technology (p. 72). American Society for Information Science.

Licklider, J. C. R. (1988). Graphical programming and monitoring final technical report. Retrieved from http://www.dtic.mil/dtic/tr/fulltext/u2/a197342.pdf.

Lowood, H. (2013). The lures of software preservation. Preserving.exe: Toward a national strategy for preserving software. Retrieved from http://lcweb2.loc.gov/master/gdc/lcpubs/2013655114.pdf

Lynch, C. (2017). Stewardship in the "age of algorithms." First Monday, 22(12). Retrieved from https://ojphi.org/ojs/index.php/fm/article/view/8097/6583.

Maron, N. L., & Smith, K. K. (2009). Current models of digital scholarly communication: Results of an investigation conducted by Ithaka strategic services for the association of research libraries. Journal of Electronic Publishing, 12(1). Retrieved from https://quod.lib.umich.edu/j/jep/3336451.0012.105.

Matthews, B., Shaon, A., Bicarregui, J., & Jones, C. (2010). A framework for software preservation. International Journal of Digital Curation, 5 (1), 91–105. Retrieved from http://www.ijdc.net/index.php/ijdc/article/view/148.

McGovern, N. (2012). Digital content review template. Digital preservation management workshops. Retrieved from https://dpworkshop.org/workshops/management-tools/process-results/template.

Miranda, B., & Bertolino, A. (2017). Scope-aided test prioritization, selection and minimization for software reuse. Journal of Systems and Software, 131, 528–549.

Monroy-Hernandez, A. (2012). Designing for remixing: Supporting an online community of amateur creators (Doctoral dissertation). Cambridge, MA: Massachusetts Institute of Technology.

Moore, R. (2008). Towards a theory of digital preservation. International Journal of Digital Curation, 3(1). Retrieved from http://www.ijdc.net/index.php/ijdc/article/view/63.

Morin, A., Urban, J., & Sliz, P. (2012). A quick guide to software licensing for the scientist-programmer. PLoS Computational Biology, 8(7), e1002598.

National Academy of Sciences, National Academy of Engineering, and Institute of Medicine Committee on Ensuring the Utility and Integrity of Research Data in a Digital Age. (2009). Ensuring the integrity, accessibility, and stewardship of research datain the digital age. Retrieved from: https://www.ncbi.nlm.nih.gov/books/NBK215264/; https://doi.org/10.17226/12615

National Academies of Sciences, Engineering, and Medicine (NASEM). (2016). Statistical challenges in assessing and fostering the reproducibility of scientific results: Summary of a workshop. Washington, DC: National Academies Press.

National Research Council. (1995). Preserving scientific data on our physical universe: A new strategy for archiving the nation's scientific information resources. Washington, DC: National Academies Press.

Norton, C. (1854). English Laws for women in the nineteenth century. Private Circulation, Ohio State University.

Palmer, C., Weber, N. M., Renear, A., & Muñoz, T. (2013). Foundations of data curation: The pedagogy and practice of purposeful work. Archives Journal, 3.

Paradigm Project. (2006). Workbook on digital private papers draft lifecycle for the long-term preservation of digital archives. Retrieved from http://www.paradigm.ac.uk/workbook/introduction/paradigm-lifecycle.html

Paskin, N. (2002). Digital object identifiers. Information Services & Use, 22(2), 97–112.

Patton, M. (2002). Qualitative research and evaluation methods. 3rd ed. Thousand Oaks, CA: Sage Publications.

Peng, G., Privette, J. L., Kearns, E. J., Ritchey, N. A., & Ansari, S. (2015). A unified framework for measuring stewardship practices applied to digital environmental datasets. Data Science Journal, 13, 231–253.

Prael, A. (2018). Centralized accessioning support for born-digital archives. Code4Lib Journal, 40. Retrieved from http://journal.code4lib.org/articles/13494.

Rosenthal, D. S., Robertson, T. S., Lipkis, T., Reich, V., & Morabito, S. (2005). Requirements for digital preservation systems: A bottom-up approach. D-Lib Magazine, 11(11). Retrieved from http://www.dlib.org/dlib/november05/rosenthal/11rosenthal.html.

Rosenthal, D. S., Rosenthal, D. C., Miller, E. L., Adams, I. F., Storer, M. W., & Zadok, E. (2012). The economics of long term storage. In L. Duranti & E. Shaffer (Eds.), Proceedings of Memory of the World in the Digital Age: Digitization and Preservation: An International Conference on Permanent Access to Digital Documentary Heritage (pp. 513–528). Vancouver, British Columbia, Canada: UNESCO.

Rothenberg, J. (1999). Avoiding Technological Quicksand: Finding a Viable Technical Foundation for Digital Preservation. A report to the council on library and information resources. Washington, DC: Council on Library and Information Resources. Retrieved from https://files.eric.ed.gov/fulltext/ED426715.pdf

Sadler, B., & Bourg, C. (2015). Feminism and the future of library discovery. Code4Lib Journal, 28. Retrieved from https://journal.code4lib.org/articles/10425.

Samuels, H. (1991). Improving our disposition: Documentation strategy. Archiv, 33, 125–140.

Shilton, K., & Srinivasan, R. (2007). Participatory appraisal and arrangement for multicultural archival collections. Archiv, 63, 87–101.

Singh, R. (2009). Digital preservation of mass media artifacts: Technologies and challenges. Journal of Digital Asset Management, 5(4), 185–195.

Sköld, O. (2018). Understanding the "expanded notion" of videogames as archival objects: A review of priorities, methods, and conceptions. Journal of the Association for Information Science and Technology, 69(1), 134–145.

Smith, A., Katz, D., Niemeyer, K. FORCE11 Software Citation Working Group. (2016). Software Citation Principles. PeerJ Computer Science, 2, e86. https://doi.org/10.7717/peerj-cs.86

Stodden, V., McNutt, M., Bailey, D. H., Deelman, E., Gil, Y., Hanson, B., … Taufer, M. (2016). Enhancing reproducibility for computational methods. Science, 354(6317), 1240–1241.

Tenopir, C., Sandusky, R. J., Allard, S., & Birch, B. (2014). Research data management services in academic research libraries and perceptions of librarians. Library & Information Science Research, 36(2), 84–90.

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). Washington, DC: Council on Library and Information Research. Retrieved from http://www.clir.org/wp-content/uploads/sites/6/pub107.pdf#page=10.

Waters, D. & Garrett, J. (1996). Preserving digital information. Report of the task force on archiving of digital information. The Commission on Preservation and Access, Research Libraries Group. Retrieved from https://www.clir.org/pubs/reports/pub63/

Woods, K., Lee, C. A., & Garfinkel, S. (2011, June). Extending digital repository architectures to support disk image preservation and access. In Proceedings of the 11th Annual International ACM/IEEE Joint Conference on Digital Libraries (pp. 57–66). New York: ACM.

Yoon, A., & Schultz, T. (2017). Research data management services in academic libraries in the US: A content analysis of libraries' websites. College & Research Libraries, 78(7), 920–933.

Appendix A: Software Intake Form

The Software Intake Form is a lightweight tool for content curators/stewards acquiring research software for long-term access and preservation.

Depositor information	
Acquisition ID	
Title of software	
Software creator(s)	
Affiliate department/institution	
Software custodian(s)	
Date of deposit	
Acquisition method	
Date of software creation	
Related project(s)	
Project funding/contributor(s)	
Project funding dates	
For what purpose was this software created?	
Are there sensitive materials in this deposit?	
If yes - what levels of access should be provided for software files?	□ World: Unrestricted access. □ Research use only: Research access only; no permission to redistribute or copy. □ Restricted research use: Research access, with restrictions TBD.

Access and Use		
Describe/ extract the computing environment(s) required for software to run:		
Type	**Name**	**Version**
Application		
Operating system(s)		
Software libraries		
Software plug-ins		
Software license(s)		
Other:		
What supplemental materials can you provide as part of your deposit?		
Type	**Description**	**Location**
User manuals		
Technical reports		
Project reports		
Bug logs		
Source code		
Correspondence		
Publications		
Other:		

On a scale of 1-5, please rate your level of agreement with the following statements:

1 – Strongly disagree; 2 – Disagree; 3 – Neither agree or disagree; 4 – Agree; 5 – Strongly agree;

____ It is important to me that software provenance has been fully documented.

____ It is important to me that I will be able to access this software in the future.

____ It is important to me that others can easily discover this software in the future.

____ It is important to me that I can replicate my previous software experience in the future.

____ It is important to me that others can use this software in the future.

____ This software offers a unique experience.

____ I want research libraries to steward this software.

____ I am comfortable with the idea that this software may be updated or enhanced.

Appendix B: Completed Software Curation Profile for GRAPPLE software

Brief Description: GRAPPLE is a dynamic graphical programming system developed by JCR Licklider while at the MIT Laboratory for Computer Science. The software project was funded by DARPA and ran from September 1982-September 1986.

Acquisition Information: Materials in The JCR Licklider Papers were first acquired by the Institute for Special Archives and Collections in 1996. Licklider was a psychologist and renowned computer scientist who came to MIT in 1950. He is widely hailed as an influential figure for his visionary ideas around personal computing and human-computer interaction.

Contents List (selected):

Type	Name	Format	URI/Location
Documentation	"User Friendliness - and All That" Pamphlet	Paper	Box 13, MC 499, JCR Licklider Papers, MIT Institute Archives & Special Collections
Documentation	GRAPPLE Interim User Manual	Paper	Box 14, MC 499, JCR Licklider Papers, MIT Institute Archives & Special Collections
Documentation	GRAPPLE Program Description	Paper	Box 14, MC 499, JCR Licklider Papers, MIT Institute Archives & Special Collections
Software	GRAPPLE software print out	Paper	Box 14, MC 499, JCR Licklider Papers, MIT Institute Archives & Special Collections
Software	GRAPPLE software magnetic tape	Tar?	Box 14, MC 499, JCR Licklider Papers, MIT Institute Archives & Special Collections
Publication	"Graphical Programming and Monitoring Final Technical Report"	PDF	http://www.dtic.mil/dtic/tr/fulltext/u2/a197342.pdf

Purpose of software: According to the user manual, the purpose of GRAPPLE was “the development of a graphical form of a language that already exists as a symbolic programming language.”

Potential use and users by material types:

Material type	Potential research/scholarly use
Computer Tape Reel	Historians of magnetic tape technology; Reconstruction of GRAPPLE for pedagogical purposes
Source Code Print Outs	Study evolution of coding language; Evidence of defunct coding syntax
User Manuals; Technical Reports	Envision new approaches to old HCI problems; Study early concepts of emojis

Defining preservation-readiness for GRAPPLE:

- Legacy media has been stabilized
- Significant materials in the collection have been cataloged and digitized (where possible).

Curation pathways for GRAPPLE:

- ***Appraise software as a collection object.*** As noted above, the GRAPPLE software is housed within the JCR Licklider manuscript collection which is at the pre-processing stage. Paper materials had been inventoried and magnetic media have been initially assessed for format migration. At this stage, the Institute Archives has recommended GRAPPLE be rehoused into archival storage. While the software itself remains inaccessible, the collection contains substantial amounts of documentation including source code print outs, project reports, informational pamphlets, and user guides). Considering the documentation related to GRAPPLE in different social contexts helps to illuminate the value of the collection in relationship to the history of early personal computing.

- ***Describe, catalog, and digitize GRAPPLE-related collection materials.*** During archival processing, digitization of paper materials like the original GRAPPLE source code and user manuals will broaden discovery and access points for the collection. The finding aid for the manuscript collection should also be updated to indicate separation of legacy media. Descriptive information for the GRAPPLE software (and other kinds of artifacts) should include: date of creation, publisher, extent, format, description, category, series title, and collection title.

Note: portions of the above have been published with references here.

Appendix C: Institutional Scenarios for Curating Research Software

An information-gathering exercise for research libraries interested in implementing software curation services. This template provides prompts for articulating scenarios to understand current institutional needs, capacity, and potential pain points. Co-developed with Jessica Meyerson of the Software Preservation Network.

I**nstructions:**

1. Articulate 4 scenarios related to research software at your institution.
2. Identify stakeholders, goals, resources, and challenges for each scenario.
3. Conduct interview with one stakeholder per scenario using questionnaire below.

Part 1: Creating Institutional Scenarios

Scenarios	**Stakeholders**	**Goals**	**Resources**	**Challenges**
	Who are the stakeholders in this scenario?	What are the goals of this scenario?	What resources are available to achieve the goals of this scenario?	What challenges are likely for this scenario?
Scenario 1: Software use				
Scenario 2: Software creation				
Scenario 3: Software reuse				
Scenario 4: Software publisher				

Part 2: Complete questionnaire with one stakeholder

1. For what purpose(s) do you [create/use/reuse/publish] software for?
   - To validate or test existing claims
   - To generate a new research outcome
   - To document or assist in the research process
   - As an historical artifact
   - To provide or recreate an experience
   - Other_____

2. What documentation is important for how you [create/use/reuse/publish] software?

- User manuals
- Technical specs/requirements
- Bugs/Testing Protocols
- Correspondence
- Promotional material
- Publications
- Other_____

3. What components are important to retain for software that you [create/use/reuse/publish]?

- Hardware / peripherals
- Libraries
- Dependencies
- Programming languages
- Algorithms
- Environments
- Documentation

4. Where is the current storage location for the software that you [create/use/reuse/publish]?

- Removable media (diskettes; CDs; USB drives)
- Computer hard drive
- Hosted on website (github; research group homepage; cloud storage)

5. What **three characteristics** are essential to preserve about this research software?

- Functionality
- Discoverability
- Reliability
- Authenticity
- Trustworthiness
- Reproducibility
- Citability
- Provenance
- Context
- Stack architecture
- Renderability
- Other_____