TACIT PROCESSES

Qualitative Analysis Toward Bottom-Up Emulation Workflows

Eric Kaltman

California State University Channel Islands
USA
eric.kaltman@csuci.edu
0000-0002-7406-3827

Adam Larson

California State University Channel Islands
USA
adam.larson535@myci.csuci.edu

Abstract – This paper describes the use of a modification of qualitative grounded theory to analyze in-situ preservation workflows involving emulation techniques. The goal of this in-process work is to identify and delineate common tasks across the emulation of different classes of software objects through a unique approach based in bottom-up qualitative observation.

Keywords – Emulation, Digital Preservation, Qualitative Analysis, Grounded Theory

Conference Topics – From Theory to Practice; We're All in This Together.

I. INTRODUCTION

Software preservation workflows are becoming necessary within the greater orbit of digital preservation. Many legacy files, programs, and other born-digital materials in collections resist or would, in fact, be damaged by migration efforts. The use of virtualization methods, like emulation, to access, view, and manipulate legacy data in its original computing contexts is, therefore, necessary to preserve both the technical context of software's use and that of users' visual, tactile, and other embodied properties. Emulation, specifically, is becoming a common, catch-all term in digital preservation for any process that involves one computing context interpreting the data of another. While there is much work on emulation for software preservation, including many large, consortia helping to support emulation efforts, much of the discussion is not focused on how to proceed with emulation work but more on what that work, at a higher level, portends for the future of digital preservation. Working with virtualized environments to correctly configure and articulate legacy software dependencies and installations is (admittedly, according to many sources) an ad-hoc or bespoke affair. The technical nuances of different historical systems are highly varied and the network of dependencies for a given piece of software (and its dependent data) can grow daunting even for experienced users. Finding points of commonality across different classes of software, and different contexts of software study, would help to create a general set of procedures to build better workflows (and better-automated solutions) for emulation in preservationist contexts.

The purpose of this short paper is to lay out a methodology based in qualitative grounded theory for examining granular records of digital preservation activities involving emulated solutions and evaluating their common processes, including the mistakes and successes along the way. Although the use of emulation and virtualization is frequently advocated, it is rarely described (due to a lack of time or space) in enough detail for novices in the area to get started. The goal for this work is to take a closer look at the in-situ, tacit, and often overlooked processes that constitute digital preservation activities. The following sections provide some needed background on both emulation in

iPRES 2023: The 19th International Conference on Digital Preservation, Champaign-Urbana, IL, US.

https://hdl.handle.net/2142/121130

preservation and qualitative methods. After that, the work proceeds with the organization of the initial study, explores early results, and then concludes with discussion and planned future work.

II. Background

This section addresses, briefly, certain technical definitions that provide context to this work, the general desire in the community for these efforts, and notes on related emulation studies.

A. Emulation in Preservation

The use of emulation in libraries and other memory institutions has grown steadily since Rothenberg first posited the need for virtualization preservation solutions for born-digital software [1]. Generally, approaches to emulation make use of off-the-shelf (OTS) emulators or virtual machine managers (i.e. QEmu or Oracle's VirtualBox) that run on a host machine and allow the installation of guest operating systems or programs [2], [3]. The configuration and management of these OTS applications can become complex in many instances, with the practitioner needing experience with both guest and host OS installation procedures, networking configuration, data formatting and imaging, and general contemporary knowledge of the target data to be emulated [4]. As articulated by Acker, emulation in preservation work is conflated with general virtualization techniques to include any approach that allows one system to imitate the functionality of another [5]. Additionally, an emerging set of projects aims to make emulation workflows easier by abstracting the complex system configuration into the cloud. Systems like EaaSI and Olive allow expert practitioners to preconfigure environments on cloud-based servers and then view them through standard web-browsers [6], [7]. This study made use of both native OTS and cloud-based solutions.

B. The Need to Articulate Preservation Process

Although there is literature on the use of emulation in preservation, including in-depth analysis of emulation use case studies, emulation workflow design, and even qualitative studies of emulation workflows, there is also a consistent call within that same literature for better articulation of the requirements needed for emulation and software preservation activities [5, 8]. In many cases, institutions lack the technical capacity and staff necessary for comprehensive software preservation activities. Hagenmaier et al. explicitly call for more work on the finer details of software preservation workflows and the determination of commonalities across practitioner practice [8]. As noted above, many software preservation efforts are ad-hoc and institution-specific. The time and attention needed to disseminate explicit descriptions of highly varied workflows (each system has its own constraints and challenges) make most accounts that of individual trees instead of the forest. This work is positioned to begin the laborious process of recording, tabulating, and organizing disparate emulation use cases into a larger, generalized framework of practice that can inform future practitioners through the creation of training resources and computational support applications.

C. Related Work in Emulation

There are a few examples of emulation and software preservation workflows that inform this work. Acker investigated the workflows and management of the FCoP project, in which numerous GLAM institutions engaged with targeted emulation case studies. Acker used a modified grounded theory approach to qualify the larger domain of emulation practice [5], [9]. This present work seeks to look at similar processes but with a more granular focus. The goal is not to divine the larger categories of emulation use in GLAMs (Acker defined "preservation", "scholarly use" and "exhibition" as top-level concepts), but to model the day-to-day, minute-to-minute investigations and processes needed to recover to-be-emulated materials.

D. Grounded Theory and Diary Studies

The methodology used below is based on grounded theory (GT) with a data collection process akin to diary studies. GT is a qualitative analysis methodology that retrieves models and theories from raw data through a bottom-up, generative, and expandable process. The purpose of GT is to avoid a priori assumptions about a domain, and instead use observational data to derive concepts about it. There are many approaches to GT and this work most aligns with Corbin and Strauss due to their allowance for directed research questions and less restrictive methodology (for instance Glaser et al. prescribe specific analysis instruments that would not be

applicable to this study's approach) [10]–[12]. In general, GT proceeds through distinct phases of initial conceptual coding, aggregating "selective" coding, and then theory "integration". Initial codes are derived from raw data and then compared and developed through "memoing", a process used to elaborate on connections between concepts and their relationship to both the contexts of the described actions and their interrelationships. Another important aspect of GT is "theoretical sampling", in which insights from an initial data analysis identify further avenues for data sampling. This allows for the analysis to find new insights and then seek out new data to reinforce or contradict an emerging theory. The analysis ends with "saturation" when the researcher divines no new concepts or connections from the sampled data. Complimentarily, diary studies approaches collect longitudinal data from participants about a repeated set of activities through a self-reported diary [13]. In this study, the researchers recorded their daily efforts at software recovery through emulation.

II. Methodology

To generate the initial observation data for this project, three Computer Science undergraduate research associates (RAs) at California State University Channel Islands (CI) recorded their attempts to transfer and emulate software data from two sources: local materials stored on legacy media formats from the CI library, and a collection of interactive project backups donated by a well-known media arts program. The local data was completely unanalyzed, so its contents and requirements were determined during the study. The interactive arts projects had previously been studied in a different context related to file format profiles of game and entertainment development records [14].

The RAs had technical experience with emulators and virtual machine managers but not much experience with digital preservation workflows. This was a benefit in that many novice issues related to information gathering and configuration were cataloged. A potential negative is that some of their challenges might not occur in actual preservation practice, however, given that many institutions do not have well-developed digital preservation programs the RAs' technical backgrounds might be more developed than some library staff. Additionally, the veritable "clean slate" of the RAs' preservation knowledge caused them to find solutions and resources that had not occurred to the preservation expert that organized the study. Regardless, the recorded sessions do indicate numerous avenues for potential training topics and resources.

The RAs worked to recover any data they were interested in among the case set items. Specifically, RAs made use of a local EaaSI node, VirtualBox, and the MacOS SheepShaver and Basilisk II emulators [15], [16]. Observations were recorded daily for two months resulting in around 700 pages of notes. RAs were instructed to be as granular as possible and to identify all information sources consulted. The goal was to make target data objects available through emulation, however, there were no direct criteria for when an emulation task was considered complete.

After data collection, the notes were loaded into Altas.ti, a standard qualitative data analysis (QDA) tool [17]. The QDA allowed for simplified comparison between notes, automatic organization of codes, and aligning codes with analytic memos. Initial coding involved a reading pass through the notes followed by assigning conceptual and identifying codes to various quoted subsets. This will allow for future search and correlation analysis. Currently, the notes feature 1447 codes across 3588 quotations tied to 41 memos, however, the analysis is far from complete. The next step is to look through the assigned codes more deeply to find patterns and conceptual duplication. Many codes cover similar concepts (as noted below), and the goal is to arrive at a set of larger categories of preservation processes derived from codes that point toward conceptual unity across use cases.

III. From Practice To Theory: An Example With Host-Guest Data Sharing

This section will briefly detail the GT process as it is applied to the sharing of data between a host and a guest operating system. Typically, systems running in virtualization are sandboxed from the host environment. This means that data and file transfer into the guest environment needs to be mediated through some interface or connection between systems. While the analysis was not explicitly looking for this phenomenon, it arose from the initial coding with 5 related codes covering 75 quotations. These rough codes ("file transfer between host and guest", "host guest shared folder", "Guest Additions", "guest additions issue", and "VirtualBox guest additions")

were then grouped under a "Host Guest Data Transfer" concept (seen in Figure 1). The "guest additions" refer specifically to a feature of the VirtualBox hypervisor that allows for modifications to be installed inside a virtual machine (VM) to implement features that were not provided by the initial guest system. In this case, the additions allow for higher screen resolution than might have existed at the time, and for certain systems to access shared memory locations to enable shared folder access.

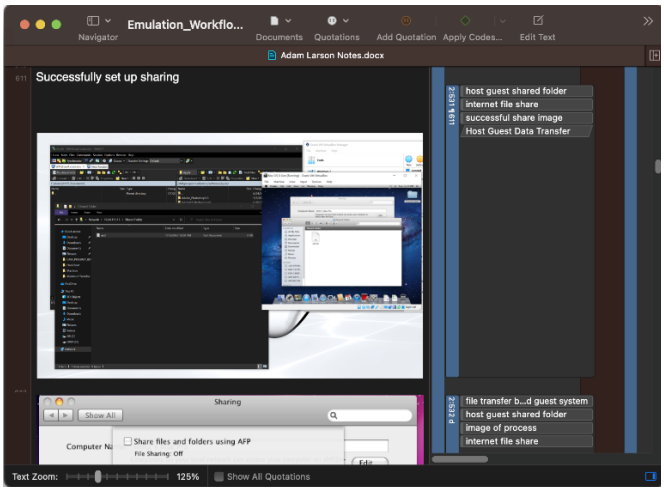

Figure 1. Altas.ti interface with codes applied to screenshot of successful data transfer

In looking at the details of the coded quotations, it is possible to cross-reference these conceptual codes with identifying codes that describe the operating system and tools being used. In this case, the codes correlated with the use of VirtualBox to virtualize Microsoft Windows (specifically ME, XP, Vista, Server 2008, 7, and 10) and MacOS X (Lion, Snow Leopard) environments, along with Sheepshaver and Basilisk II (for System 7, MacOS 8 and 9). To proceed further, GT methods then inquire into the specific dimensions and properties of observed actions and interactions. These are then placed in a larger context to hopefully intuit some emerging theory of process. One potential property of the "Host Guest Data Transfer" was the specific interface needed to allow it. Based on the coded quotations, there appeared to be four primary methods of importing data into a guest environment:

1. Shared folders that allowed for a storage location on the host to be mounted inside the guest.
2. Shared network folders connecting the host and guest machines through a virtual network controller.
3. Allowing the guest to connect to the Internet and remotely download files.
4. Loading the data into a disk image and mounting it in a virtualized media drive.

While these approaches were not decided in advance, they emerged as a result of the interaction between the practical needs of the RAs for recovering specific objects and the available features provided by the virtualization technologies. The use of GT allowed for the organic detection of specific patterns of preservation actions and interactions that corresponded with the larger "process" of host to guest data transfer. Here, the analysis highlighted data transfer as an area of contention among the RAs (in that they repeatedly noted difficulties with consistent data sharing) and what the general solutions appeared to be, given the case items.

Further, it is possible to view the quotations linked to these methods and divine potential dimensions of the data transfer concept, like the symmetry of the methods used. In the case of methods 1 and 2, there was a symmetric link established between the host and guest that allowed for transfer into and out of the guest environment. However, methods 3 and 4 are unidirectional, in that they allow for data to go into the guest environment without a complementary retrieval mechanism. In fact, method 4 was the primary means used by the EaaSI system for inserting data into environments highlighting a potential difficulty with cloud-based emulation solutions vis-à-vis locally executed ones. This dimension of "symmetry" in data transfer processes is then a potential new site of analysis as the concept can be compared with the literature for further elaboration and validation.

Additionally, it is also possible to look at the knowledge context within which these data transfer methods are embedded. Since the RAs also recorded where they researched the data sharing methods, a network of online and textual documentation, individual experimentation, and online tutorials and videos prefigures the combined knowledge necessary to engage, as a preservation practitioner, with the "Host Guest Data Transfer" concept. Continued work on related preservation tasks would likely position this concept relative to other processes needed for the emulation of target data objects. From this, a general theory of emulated preservation techniques could then emerge.

Finally, there is serendipity and surprise in the malleability of the GT approach that finds meaning in "mundane" minutia. When working through sharing method 1, one RA realized that they needed to move files from the guest shared folder into a local one to avoid permission and access issues. Another RA discovered that method 2 necessitated removing significant network security features from the *host system* for guest access to be possible. These new notions related to "permission" and "security" might now be potential vectors for dimensional analysis.

IV. Promise, Limitations And Future Work

The preceding example highlights how close attention to practitioner activities can reveal deeper relationships between seemingly disparate preservation targets and points to the potential for subject agnostic knowledge sharing. In the example above, the RAs were working with data from disparate sources but they all still needed to find some way to get that data, once acquired, into the emulated environment. The GT approach required that the raw details of the process be reconsidered in comparative and generalized contexts, and it was through this consideration that patterns started to emerge. However, this work is developing as there are hundreds of codes to aggregate and process.

Contrarily, some limitations of the current study must be noted, including items that will change for future studies. The sample size, while extensive in activities, was limited in participants. The RAs worked for a combined 960 hours, and the resulting notes are rich in specific details relating to a variety of emulated environments. However, since the RAs were students and not trained preservationists, it is unclear if some of the specific procedures or issues encountered might simply not occur with more experienced practitioners. A caveat here is that GT methods are designed to address sampling issues by allowing for "theoretical sampling" based on progressive findings. It would be feasible to add the subject position of the individual as a dimension of the analysis and compare practitioner experience with the execution of preservation tasks. Additionally, embedding more self-reflection into the process would be beneficial. The researchers did not proceed with GT analysis until after the initial data collection interval ended. It would have helped the study to begin coding and analysis during data collection to steer the RAs toward fruitful pathways. The next steps for this research are to proceed with constructing a model of both the dependencies and related processes incumbent on the emulation of software and software-dependent data objects. Current progress is promising and there are likely to be more unlikely commonalities discovered across the documented use cases, effectively creating theory from practice.

V. Acknowledgements

We wish to thank the additional RAs, Desireé Caldera and Morgan McMurray, and the CI Summer Undergraduate Research Fellowship (SURF) for providing support to this work.

VI. References

[1] J. Rothenberg, "An experiment in using emulation to preserve digital publications," 2000, Accessed: Sep. 16, 2016. [Online]. Available: http://www.kb.nl/sites/default/files/docs/emulationpreservationreport.pdf

[2] "Oracle VM VirtualBox." https://www.virtualbox.org/ (accessed Sep. 19, 2021).

[3] "QEMU." https://www.qemu.org/ (accessed Sep. 19, 2021).

[4] D. S. Rosenthal, "Emulation & Virtualization as Preservation Strategies," 2015.

[5] A. Acker, "Emulation practices for software preservation in libraries, archives, and museums," *J Assoc Inf Sci Technol*, p. asi.24482, May 2021, doi: 10.1002/asi.24482.

[6] "EaaSI," *Emulation-as-a-Service Infrastructure*. https://www.softwarepreservationnetwork.org/emulation-as-a-service-infrastructure/

[7] M. Satyanarayanan *et al.*, "Olive: Sustaining executable content over decades." XSEDE, 2014.

[8] W. Hagenmaier, C. Williford, L. Work, J. G. Benner, S. Erickson, and M. Lassere, "Supporting Software Preservation Services in Research and Memory Organizations," Software Preservation Network, White Paper, 2022.

[9] A. Acker, "Accessing Software: Emulation in Information Institutions." Rochester, NY, Apr. 06, 2023. Accessed: Jun. 08, 2023. [Online]. Available: https://papers.ssrn.com/abstract=4450195

[10] K. Charmaz, *Constructing Grounded Theory*, 2nd Edition. London ; Thousand Oaks, Calif: SAGE Publications Ltd, 2014.

[11] B. G. Glaser, A. L. Strauss, and E. Strutzel, "The discovery of grounded theory; strategies for qualitative research," *Nursing research*, vol. 17, no. 4, p. 364, 1968.

[12] J. Corbin and A. Strauss, *Basics of qualitative research: Techniques and procedures for developing grounded theory*. Sage publications, 2014.

[13] K. Salazar, "Diary Studies: Understanding Long-Term User Behavior and Experiences," *Diary Studies: Understanding Long-Term User Behavior and Experiences*, Jun. 05, 2016. https://www.nngroup.com/articles/diary-studies/ (accessed Nov. 30, 2022).

[14] E. Kaltman, R. Lorelli, A. Larson, and E. Wolfe, "Organizing a Content Profile for a Large, Heterogeneous Collection of Interactive Projects," in *2021 IEEE International Conference on Big*

Data (Big Data), Dec. 2021, pp. 2231–2239. doi: 10.1109/BigData52589.2021.9671904.

[15] C. Bauer, "SheepShaver," *SheepShaver: An Open Source PowerMac Emulator*. https://sheepshaver.cebix.net/ (accessed Mar. 10, 2023).

[16] C. Bauer, "Basilisk II," *Basilisk II: An Open Source 68k Macintosh Emulator*. https://basilisk.cebix.net/ (accessed Mar. 10, 2023).

[17] "ATLAS.ti," *ATLAS.ti*. https://atlasti.com (accessed Mar. 10, 2023).