Reviving Antique Software: Curation Challenges and the Olive Archive

Daniel Ryan
Carnegie Mellon University
WEH 4418, 5000 Forbes Ave
Pittsburgh PA 15213
+1 (412) 268-5278
dfryan@andrew.cmu.edu

Gloriana St. Clair
Carnegie Mellon University
WEH 4418, 5000 Forbes Ave
Pittsburgh PA 15213
+1 (412) 268-5278
gstclair@andrew.cmu.edu

ABSTRACT
A growing percentage of the world's intellectual output is in the form of executable content, such as simulation models, tutoring systems, data visualization tools, and expert systems. To preserve this content over time, we need to freeze and precisely reproduce the execution that dynamically produces that content. Olive, a rough acronym for "Open Library of Images for Virtualized Execution," is a system built at Carnegie Mellon University. Olive preserves and provides access to this executable content. It relies on virtual machine (VM) technology to bundle software with all of its dependencies. These VMs are streamed over the internet in real time to ensure a smooth user experience while maintaining fidelity to the original execution environment[1].

This demonstration examines some of the challenges the Olive team has encountered in the process of preserving software over the last several years. Among these difficulties are technical challenges, problems of scale, legal limitations, and a lack of existing curation standards for executable content.

General Terms
infrastructure, preservation strategies and workflows, specialist content types, case studies and best practice.

Keywords
preservation, software, virtualization.

1. INTRODUCTION
Born-digital interactive content makes up an increasing proportion of creative and scholarly output around the world today. The global, instantaneous, and unrestrainable nature of software has made it a major part of our cultural heritage. Significantly, executable content draws its cultural impact from its interactivity: users have to participate and interact with software in order to understand what it does, how it works, and why it is useful.

Historically, libraries, museums, and other cultural memory organizations have been effective in preserving the developing record of civilization globally, and in assisting the users of that record to understand it and to use it to create new knowledge. In the arts and humanities, citizens and scholars can view cave paintings at Lascaux, the Bayeux tapestry, the Bill of Rights, the archival papers of U.S. Senator John Heinz, and over twenty million books. Published scholarly work is more widely disseminated than it has ever been. Those interested in their heritage can listen to traditional music, study ancient commercial records and texts, and attend plays written by Shakespeare. Currently, these seekers cannot use primary source materials from the growing realm of executable content, because the *execution environment* is not compatible with modern technology. Instead, scholars must rely on a variety of secondary sources, including screenshots, descriptions, and community commentary.

In *Preserving Digital Information, Report of the Task Force on Archiving of Digital Information,* Don Waters and John Garrett made a daunting prediction that if libraries did not seek to preserve digital information, the result would be difficult. "Failure to look for trusted means and methods of digital preservation will certainly exact a stiff, long-term cultural penalty[2]."

The pervasiveness of executable content is a worldwide phenomenon. When historians look back on the nature of society during the computer revolution, they will need working, perfectly faithful instances of the software in use and the experience of interacting with it. When sociologists seek to understand exactly which characteristics of Angry Birds drove many adults internationally to spend large portions of time flinging digital birds at digital houses, they will need to run it and play it themselves. No explanation or description could suffice.

2. PROJECT HISTORY
In 2012, Carnegie Mellon computer science professor Mahadev Satyanaranan (Satya) approached the Dean of Libraries, Gloriana St. Clair, to discuss a project for which he saw an application that might be suited to the University Libraries. Satya had been working with Vasanth Bala (Vas) at IBM Research to package and stream virtual machines for fast application deployment.. This project was known as Internet Suspend/Resume® (ISR). As the project evolved, Satya and Vas began to see ISR's potential for preserving software. The ISR team understood the technical and infrastructural challenges behind such a project, and thought it was worth investing the time and money to devise a solution. Neither IBM nor Satya was interested, however, in keeping old things around forever. They agreed to begin by reaching out to the Carnegie Mellon Libraries, where Gloriana had established a reputation as a digital pioneer and an extensive collaborator with the computer science department. Thus, the Olive project was born. St. Clair assured Satya that not only were she and the CMU Libraries interested in solving this problem, but also that the library community shared her sense of responsibility around executable content.

In 2010, IBM hosted a meeting to test the idea that libraries and campus computing might be interested in preserving executable

content. Participants were enthusiastic about the technology, anxious about the legal situation, and worried about both the economic and the organizational issues.

Both IMLS and the Sloan Foundation gave grants for a proof of concept phase of Olive development. Since October 2012, the Olive project has received $497,000 from the Institute for Museum and Library Services, and $400,000 from the Sloan Foundation, to support a proof of concept effort and development. Part of the funding sought from the Sloan Foundation was awarded to Ithaka S+R for a whitepaper on sustaining an entity like Olive after the core research and development has been completed. The report recommends an additional three years of funding for intensive R&D, followed by the formation of a sustaining coalition of interested parties sharing the financial burden of the operational costs of such an archive.

3. APPROACH

3.1 Execution Fidelity

Software reproduction is a complex problem, the solution of which requires the perfect alignment of many moving parts. Achieving execution fidelity has long evaded preservationists and has stymied the efforts of the digital library community to archive executable content[8][3][4]. Even minor changes can cause a breakdown in the stability of the execution environment. These changes can include dynamically linked libraries, preferences, configuration files, clock timings, hardware capabilities, and more. Simply constructing the appropriate environment in which legacy software will execute often requires expert knowledge of the original environment. We refer to the successful alignment of all of these variables as *execution fidelity*[5][6]. As legacy software falls further into deprecation, the level of knowledge required to achieve execution fidelity becomes increasingly rare.

3.2 Virtual Machines

In order to encapsulate an execution environment, Olive relies on virtual machine technology. Communicating with a *virtual machine monitor*, VM images are supplied with a virtualized representation of a computer architecture and instruction set see *Figure 1*). Virtual machine monitors leverage the actual hardware of a machine (the host machine) to ensure that the operating system and applications inside are unable to distinguish between the virtual environment and a real legacy system. This precise imitation of hardware is why Olive relies on virtualization as a preservation strategy.

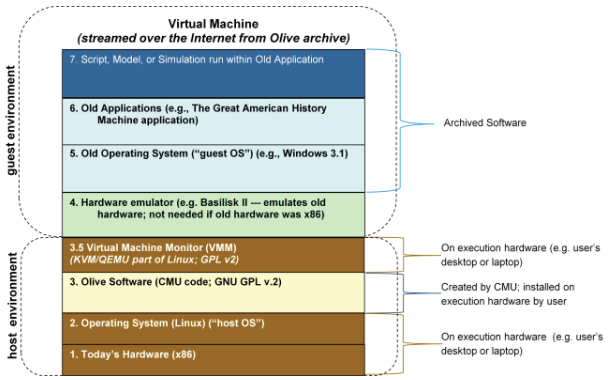

Figure 1: Olive Client Architecture

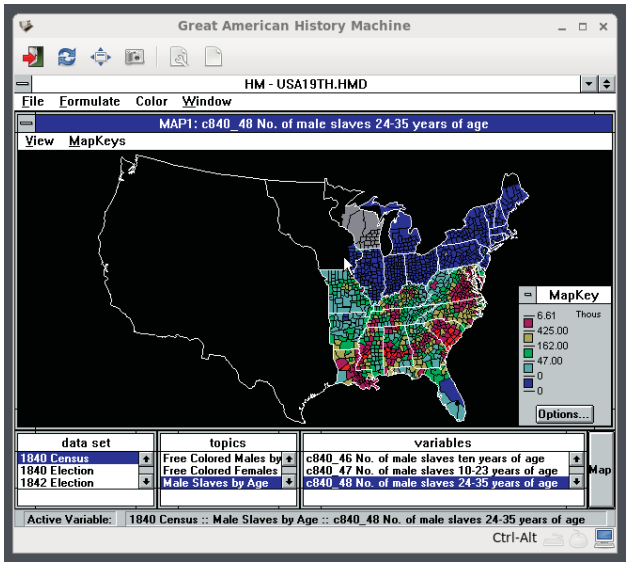

Figure 2: Great American History Machine (Windows 3.1)

Olive is built on standard, unmodified web technologies (standard web servers, HTTP for communication) and works to stream VM images in pieces as they are requested. Execution can happen either directly on a user's computer or on a compute node dedicated to VM execution.

3.3 Examples & Demo

There are several pieces of software archived in Olive, but here we will focus on only two brief examples:

1. The Great American History Machine (see Figure 2): A piece of educational software written in the late 1980's by Professor David Miller at Carnegie Mellon. This software was used to teach early American History at institutions across the United States. It offers unique tools for exploring census and election data. Professor Miller and his team did not have the technical resources to migrate this tool when Windows 3.1 became deprecated, so the software fell into disuse until we recovered it.
2. Mystery House (see Figure 3): Mystery House is the original graphically-rendered adventure game written for the Apple II. It brought graphical interaction to the mainstream just over 30 years ago, yet actually running that software today is a significant challenge; not only did we need the original disk image, but we also had to find an Apple II emulator and the accompanying ROM (read-only memory), which was originally built into the machine. Without archival, executable instances of software like Mystery House, we lose our ability to reflect on the history of computer games and human/computer interaction.

These examples highlight the potential for olive to preserve and provide access to software which might otherwise be lost.

4. CHALLENGES

4.1 Technical Challenges

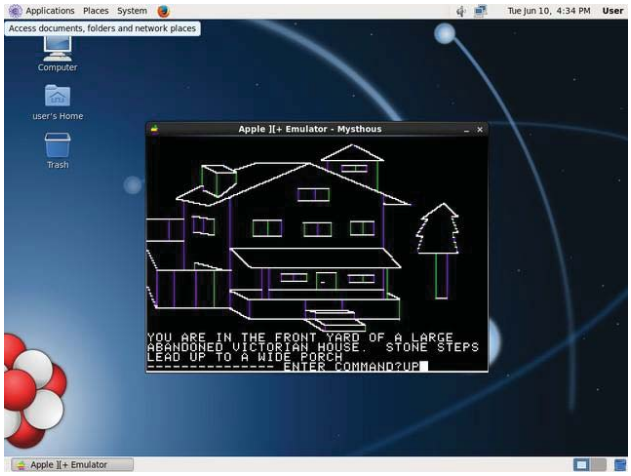

Figure 1: Mystery House on Apple II emulator

In the simplest terms, Olive will be like YouTube for executable content. Olive provides a tool for preserving and remotely accessing software. To preserve this content over time, we need to freeze and precisely reproduce the execution that dynamically produces the content. While this may sound simple, many have studied the problem over the last two decades, but only now are successful efforts underway.

Here are a few of the technical challenges we have encountered while trying to achieve a working implementation of Olive:

- Low latency streaming and caching of VM images[8];
- Lack of backward compatibility in updated releases of dependent software;
- Bugs which existed in old software/hardware but only present themselves in modern systems;
- Effective, secure, and flexible implementation of access controls, and
- Failure of modern VMMs to represent faithfully the extended memory space required to run certain systems.

For example, the version of qemu/kvm bundled with Ubuntu 12.04 was several releases out of date as compared with that packaged with Redhat Enterprise Linux. VMs built on RHEL 6.x would fail to boot when exported to an Ubuntu 12.04 machine with qemu/kvm installed from the normal Ubuntu repository. In order to overcome this difficulty, the software Olive provides for packaging VMs strips down and validates the XML. This XML is responsible for defining the configuration of a VM in order to ensure continuing compatibility, both forward and backward[7][8].

In another edge case, we discovered that Windows 3.1 mouse support suffered from a strange bug which caused the mouse pointer to jump randomly around the screen. Upon investigation, the Olive development team learned that the serial mouse drivers for Windows 3.1 contain an off-by-one error which is only exposed when mouse updates occur more than 40 times per second. On older mice, this did not cause problems because they did not send updates so frequently. However, modern laser mice send information much more often than 40 times per second. After tracking this issue down, we were forced to construct a binary patch for the driver.

4.2 Legal Challenges

The world's accrued wisdom is available to scholars and students globally. In general, the pre-1923 U.S. content can be benefited

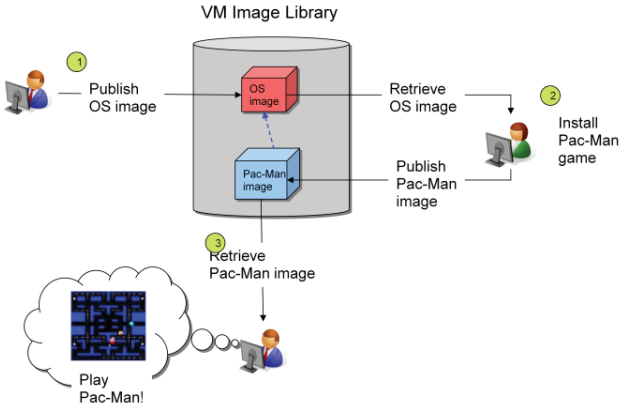

Figure 2: Crowdsourced Publication Workflow

from without much concern about being sued for reusing that work in the creation of newer work. For instance, Shakespeare's output can be performed in all kinds of redacted and enhanced formats and interpretations. Shakespeare's heirs may wince but they cannot and do not sue. In striking contrast, the family-profit-maximizing Tolkien Estate manages its assets by aggressively controlling all aspects in all formats. J. R. R. Tolkien himself sued Ace Books for publishing a pirated paperback edition of *The Lord of the Rings*. Ace paid damages and Tolkien's publisher moved to meet demand by bringing out an authorized paperback. The Tolkien Estate continues to be zealous in managing its property. For a less popular author, this approach might be detrimental.

Generally, most post-1923 content requires some kind of license in most countries. Presently, Olive is a closed research project, which affords it certain protections from infringement claims under fair use provisions of the copyright law. However, we recognize the need for an open, accessible archive for software, and CMU General Counsel Mary Jo Dively commissioned an extensive risk assessment of varying levels of public access to Olive. We are continuing to study this report.

4.3 Curation Challenges

Many collections of historical material are established, managed, and maintained by curators, who are responsible for selecting content, developing and applying an acquisition process, and keeping that content secure from degradation. Often this means protecting works of art from sunlight and flash photography, or protecting books from falling apart. When the object of curation is a piece of software in executable form, however, the process of curation is not particularly well defined. For a given piece of software, curation might involve identifying the hardware it requires to operate, locating an emulator for that hardware (if necessary), determining the platform and version required to run the software, configuring the emulator, installing and configuring the platform, locating and importing dependent drivers, installing the software, ensuring faithful behavior, generating metadata, and tracking down related rights information, and packaging and uploading the containing VM.

This set of tasks would be daunting enough given a modern, well documented technology stack. For old or deprecated software, the dependency stack will often require extensive expertise to configure and install successfully, if it is still possible to identify and acquire the full dependency stack at all. Documentation for these configuration and installation procedures is often lacking, and finding an expert will become increasingly difficult.

As part of a grant from the Institute for Museum and Library Services, the Olive team agreed to preserve Doom, the original first-person shooter game written for MS-DOS. Beginning with an image of the original MS-DOS 6.22 installation floppy disk, we soon learned that reliable instructions for achieving a successful system configuration were scarce, poorly documented, and largely dependent upon third party additions with similar challenges. Similar issues arose when we attempted to install Windows 95.

Because of the degree of expertise required and the sheer quantity of software which is in jeopardy of becoming extinct, we currently plan to investigate crowd-sourced curation models in the next phase of our work. As we move forward, our development team is implementing functionality to allow new VMs to be published as a changeset applied to an existing VM, which would eliminate the need to confront a complex dependency stack more than once. A sample curation workflow supported by this model can be seen in Figure 4.

5. CONCLUSION

Preserving software in its execution environment is critically important to our institutional goal of preserving the cultural record. The Olive Archive is an infrastructure designed to limit challenges to future curators, but will begin to rely more heavily on community involvement in the coming years. Many important questions must be addressed by curators and preservation experts as institutions take on the daunting challenge of capturing, describing, checking, cleaning, migrating, and maintaining collections of software in virtual machines.

6. ACKNOWLEDGMENTS

The Olive Archive is supported by grant funding from the Institute for Museum and Library Services and from the Sloan Foundation. We are grateful to Vas Bala and IBM for initially supporting this research, to Carnegie Mellon for financial, legal, technical and moral support, and to Deanna Marcum and Ithaka S+R for their ongoing insight and assistance. Final thanks to the advisory group and to our project team: Erika Linke, Mahadev Satyanarayanan, Keith Webster, Benjamin Gilbert, Jan Harkes, Jerome McDonough, and Anita de Waard.

7. REFERENCES

[1] Open source software at the Olive Archive can be found available at https://github.com/cmusatyalab.

[2] Donald Waters and John Garrett, "Preserving Digital Information, Report of the Task Force on Archiving of Digital Information," Council on Library and Information Resources, May 1996. Available: http://www.clir.org/pubs/abstract//reports/pub63.

[3] P. Conway. Preservation in the Digital World. http://www.clir.org/pubs/reports/conway2/, March 1996..

[4] P. Conway. Preservation in the Age of Google: Digitization, Digital Preservation, and Dilemmas. *Library Quarterly*, 80(1), 2010.

[5] B. Matthews, A. Shaon, J. Bicarreguil, and C. Jones. A Framework for Software Preservation. *The International Journal of Digital Curation*, 5(1), June 2010.

[6] Satyanarayanan, Mahadev ; Bala, Vasanth ; Clair, Gloriana St. ; Linke, Erika ; Georgakopoulos, Dimitrios (Bearb.) ; Joshi, James B. D. (Bearb.): Collaborating with executable content across space and time.. In:*CollaborateCom* : IEEE, 2011. - ISBN 978-1-4673-0683-6, S. 528-537.

[7] Gilbert, Benjamin. 2013. Building VMNetX with qemu and libvirt. Workshop. Carnegie Mellon University (Jun. 2013), https://olivearchive.org/static/documents/vmnetx-gilbert.pdf.

[8] Bowman, M., Debray, S. K., and Peterson, L. L. 1993. Reasoning about naming systems. *ACM Trans. Program. Lang. Syst.* 15, 5 (Nov. 1993), 795-825. DOI= http://doi.acm.org/10.1145/161468.16147.

[9] Yoshihisa Abe, Roxana Geambasu, Kaustubh Joshi, H. Andrés Lagar-Cavilla, and Mahadev Satyanarayanan. 2013. vTube: efficient streaming of virtual appliances over last-mile networks. In *Proceedings of the 4th annual Symposium on Cloud Computing* (SOCC '13). ACM, New York, NY, USA, , Article 16 , 16 pages. DOI=10.1145/2523616.2523636 http://doi.acm.org/10.1145/2523616.2523636