



A Guide to FCoP Documentation

Table of Contents

Introduction	2
Themes	3
Terminology and definitions	3
Limits of scale and resources	4
Emulation entangles us in another layer of software	7
Upstream software preservation (pre-accessioning)	7
Software use expectations	9
Use cases	10
When to test?	12
User interactions with EaaS	13
Snapshots of software preserved	14
Software is physical too	14
Advocating the value	15
Software preservation operates in socio-technical infrastructures	16
The law and software preservation	17
The wonderful and frightening world of metadata	18



Introduction

The Fostering Communities of Practice (FCoP) project sought to document the activities of the cohort it formed in a variety of ways, and for a variety of reasons. It is suffice to say that the project has met the goal of its initial proposal to contribute to a “field-level” understanding of software preservation and emulation experiences. But what form does that understanding take? How applicable is it outside of the six organizations that participated in the cohort? And how much of that understanding directly relates to the mechanics of software preservation and emulation, and how much to external factors influencing those mechanics?

This guide to FCoP documentation will focus on the documentation produced by the cohort participants, and will highlight some of the commonalities that can be found across the different pieces of documentation. More straightforward tours of the documentation can be found on the FCoP website, as well as within some of the cohort participants’ final reports. Here, we hope to begin proposing themes or preliminary project findings that could be extended to other organizations and, potentially, organization types.

Before digging into the thematic concepts found in the documentation, it can be useful to distinguish between two high-level formal types of documentation that came from this project. On the one hand, we have documentation that was driven by FCoP project goals, and took a form that was often dictated, sometimes a priori, by the project staff. This type of documentation can be viewed as Tools and Templates. The other type of documentation was more specific to local cohort organizational settings, and may discuss opportunities and challenges that may not be as relevant outside of the cohort organization. Documentation of this kind might fall into a category of Emergent Documents.

The themes presented here have not been derived according to methodological processes and tools typically used by researchers for rigorous scholarship. Rather, the FCoP project staff identified categories during the review of the documentation, and often isolated specific passages or parts of the documentation relevant to the category or categories. Indeed, some documents have aspects that relate to more than one category. This process may have introduced certain biases carried by the FCoP project staff, who as professionals rank software preservation high on their priority lists. For example, at least 5 of the categories approach the idea of use from different angles; some categories are broad and expansive and some are narrow and may reflect interests of the staff; and finally, a certain bias towards archival thinking is noticeable. However, while control for bias has not been strictly observed, the FCoP project staff feel nonetheless that the guide can provide a valuable alternative index into the FCoP documentation. (Note all text, unless quoted, is contributed by FCoP project staff, and not by the cohort participants themselves).



Themes

Terminology and definitions

Software preservation has been a difficult term to define precisely. Both 'software' and 'preservation' as individual terms have multiple meanings for different communities, and when they are combined the situation becomes that much more complex. The FCoP cohort participants, through their experiences, have come to their own understandings which come out in their documentation, and they have also demonstrated that our definitions in this emerging field are evolving. Other terms have been defined along the way.

References:

- [UVA "Definitions for Today" \(slide 7\) of Intro deck](#): helpful as a starting point, also could be helpful as a set of interpretations that could be discussed and modified.
- [UA Software Preservation 101](#) Module:
 - Terms to Know slide (#7)
- [UA IDCC paper](#), summary of Communities of Practice section: "software preservation is necessarily a transdisciplinary challenge and thus must be approached by multi-disciplinary, collaborative, and highly communicative teams."
- From [UI 2014_07_HembroughSoftwareProject_Plan](#):
 - "Highest quality/most complete software" (in the context of de-duplicating software media):
 - Professionally manufactured software
 - Complete sets of software (no missing install disks)
 - Packages that include manuals/packaging/instructions
 - Software with serial numbers, licensing information.
- From UI 2019_11_20_DataInterestGroup folder:
 - [5.d of the script](#): "Emulation strategies focus on hardware and software environment recreation rather than transforming the digital object. The original file remains unmodified; it is the computing environment that changes." However, does the "original file" really "remain unmodified" if it displays differently or the user interacts with it differently through an emulated environment?
- [UA IDCC paper](#): PtU (preservation through use) is an attempt at memory-making
- [UA IDCC paper](#), Page 3: "A game does not exist until a player powers up the preserved and articulated software and hardware. Preserving video games thus not only benefits from use—it requires it." Is not the interactivity between object and rendering software just as essential to all software-dependent objects as it is between video game and player? The paper starts investigating this question of change over time later in the paper.



- [UA IDCC paper](#), first paragraph of Page 5: “We also learned in this first meeting that the participants had no shared software preservation vocabulary.”
 - “Challenges” section (starts on pg. 6) could be summarized as a lack of clarity, shared understanding, or shared vocabulary about software preservation, across disciplines.
-

Limits of scale and resources

One goal of the FCoP project was to move software preservation and emulation towards the mainstream of digital preservation practice. To accomplish this, however, the problem of integrating this new kind of work into existing workflows and procedures, as well as the problem of securing the resources necessary to convert software preservation from project-based experimentation to programmatic responsibilities, have emerged as an obstacle. The FCoP cohort has recognized these obstacles, as well as identified opportunities for making software preservation scalable and distributing the responsibilities to their colleagues (within their organizations, and among their professional peers at other institutions), and gave themselves permission to say that with limited resources, it is okay to be perform only the vital, critical operations.

References:

- [UA “Workflow Instructions”](#) section as training tool (begins on pg. 6)
 - [UA ‘LGIRA Workflow’ diagram](#) references the Workflow doc
 - What does EaaS do for you? (slide 22 of [UVA Technical Overview presentation](#)): stark statement of the current need for human interpretation. The slide also makes clear that basic set up of software (if not already configured, perhaps in EaaS) that much human intervention is required to set up.
- Internal vs. External description needs ([UVA Metadata and Description presentation](#)) illustrates how description in this area proceeds from necessity (internal) first and branches out from there in a rather vague, ambiguous way.
- “Goal: Minimal unique effort, and greatest possible sustainability” (pg. 15 in the [UVA Workshop Shared Notes](#))
- Pg. 18 of UVA [Workshop Shared Notes](#): “I could imagine copy cataloguing information for software - rather than everybody having to do it separately.” This is the approach that could counter the “item-level description is too time-consuming” concern stated earlier in the notes. But can there be a single source for the copy cataloguers? The presentation gives several places to go.
- Goal of the [UVA Software Questionnaire](#): emphasis on decreasing the burden for all involved. Dispelling the confusion, bringing clarity.



- Context and Boundaries section of the UVA Software Questionnaire implies that software should only be considered in relation to the collection it comes in with. Relates to the overall goal of the questionnaire (reducing burden).
- From the [UVA Final Report](#), about the Software Questionnaire: "It can be used and integrated into the work of curators, accessioning archivists, processing archivists, university archivists, and preservationists as the information drawn from these questions and answers can support appraisal, processing, description, preservation, and access workflows in the years to come for the collection (or can aid in the recommendation of the collection to other collecting institutions)."
- [UI Data Interest Group presentation script](#) (2019-11-20):
 - 2.f in script: affirmation that programmatic services limiting themselves to bit-level preservation is acceptable. Followed on by idea in 2.g. But what if the bit-level preservation never ends and you can never step back and strategize, plan, and develop other programmatic services?
 - Section 3 of script: Recap of the FCoP goals and opportunities, especially the UI background that the presentation gives, where it talks about "forays" into emulation but finding nothing scalable until FCoP..."
 - 5.f-h in script: scale, scale, scale... as well as "steep resource barriers to entry"
 - 5.i in script: we want to make this [scale/capacity] visible/explicit with this project
 - 9.ii: "drafting guidance documents related to workflow, resources and scaling efforts (not as fun nor as sexy as demo'ing the emulator)"
 - 9.vi: "Moving into a scalable and service-level implementation require significant curation work which must be done with engagement from content curators, making decisions based on preservation priorities, documenting what we've done to continue to build digital preservation capabilities, and making the work visible in order to share with others interested in undertaking software preservation and emulation efforts and to illustrate the workflows."
 - 11: "scaling the work beyond myself"
 - 11.a.i: "The resources required to curate legacy content in order to render the files is considerable. Curating this content to full functionality requires software, knowledge on how to run the software and associated operating system environment, it requires patience to discover and resolve errors which arise from software and file dependencies (such as specific fonts linked to a document) or making the decisions about what errors are acceptable (or desirable to maintain) and documenting choices made."
 - 13.a: "emulation is not a magic box" --> "levels of rendering success" ---> degrees of emulation; rendering success "depends" on or influenced by environment complexity and advance analytic work
 - 13.b: outreach and inreach important for visibility and distributing responsibilities



- 14.c: collecting software should require the same amount (if not more) forethought and strategy as any other collecting effort.
- From [UI 2019 Preservation Week script](#):
 - 6.a: required: patience to discover and resolve errors. Errors will happen, don't shy away from them. What errors are acceptable, and how can we document them?
 - 18.b: the need to balance this work with all the other responsibilities
 - 18.d: some criteria for keeping the balance (and not using the full emulation solution in all situations, which is not sustainable).
- From [UI 2019 SAA Popup script](#):
 - Slide 5: "The FCoP project has facilitated prioritizing emulation and software preservation, particularly within the broad landscape of institutional digital preservation activities where there are often competing priorities for time and attention." [bolding added, also not sure if this had been explicitly stated before].
 - Under slide 8's "Creating workflows for testing software" heading: "But the responsibility for this work must be shared in order to make efforts scalable."
- [GT FCoP Software Emulation Inclusion Criteria/Checklist](#) is a good example of local documentation scoping the kinds of software it wants to emulate. You don't have to try to emulate everything.
- [UA Workflow Documentation \(2020-03-10\)](#):
 - Pg. 4 (During the Processing stage): "Because the Archive takes in more material than there is available labor to process it, at this stage, video games and systems are not tested to see if they function properly."
 - Pg. 4 (Preservation stage): "There is even a current offer from a LGIRA user to help repair a multi-ton hydraulic arcade machine. In a way, the community that uses the materials is also contributing to their preservation."
 - Pg. 6 (Access stage): "Due to the technical and legal complexity of the issues related to emulation, user procedures for accessing emulated resources are still to be determined."
 - Selection for Emulation section: The list of prompts could be a useful framework for others. The Risks section seems to be largely addressed by ARL Best Practices, if the emulated environment is not made available online for the public.
 - "The answers to these prompts help LGIRA's leadership team make informed decisions about when and how to pursue an emulation project." Emphasis on the word "project." It continues: "Practically speaking, however, the organization's limited resources—and its desire to steer clear of legal and/or bureaucratic impediments—determines, to a great degree, which emulation projects we are reasonably able to pursue."
- [UI Use of Emulation-as-a-Service Infrastructure \(EaaSI\) for Preservation and Access at U of I Libraries](#):



- Pg. 1: “Emulation is an oft touted digital preservation strategy. However, emulator use in the professional arena is often limited to research projects or to institutions that have a great deal of resources dedicated to digital preservation and digital curation. Widespread and scalable implementation is limited as there are steep resource barriers to entry.”
-

Emulation entangles us in another layer of software

It is tempting to think that software emulation is an end in itself. But at least two FCoP cohort participants expressed in their documentation that not only is emulation more of a means to an end, but that it also might actually add yet another layer of software associated with a digital object’s long-term access (the real end). It might even introduce the need to emulate the emulator. Emulation does not reduce software dependency, it may just strengthen it.

References:

- [GATech RetroTech Online System Diagram](#) (in retroTechOnline github README.md)
 - [UI Data Interest Group presentation script \(2019-11-20\)](#):
 - 6.b: Anxiety of the emulation within a emulation within a emulation.... [unless the emulator developers somehow keep on top of technology changes]
 - 14.b: EaaS as a tool for migration
 - 14.b.ii: using emulation to mitigate the need to further use emulation (see also 14.b.iii)
 - [UI 2019 Preservation Week script](#):
 - 9.g.i: related to DataInterestGroup presentation’s discussion of emulation as a means for a migration end. Here, though, the presentation uses an interesting term: emulation as a “bridge technology”.
-

Upstream software preservation (pre-accessioning)

The FCoP cohort documentation reinforces the notion that the more information cultural heritage organizations can collect from record creators, and the earlier they can collect it, the better off downstream processes will be. Who is more downstream than preservationists? Beyond reinforcing this idea, the documentation offers solutions and tools for information professionals across the spectrum of the digital curation lifecycle that can facilitate, improve, and contribute to software preservation and emulation strategies.



References:

- [UVA Digital Donor Checklist \(V0.4\)](#)
 - “Types: What are the file types and formats that comprise the collection (TIFF, doc, etc.)?”
 - Email section
 - Software and software-dependent materials section
 - Decision tree distinguishing between
 - commercial/ubiquitous
 - commercial/scarce
 - non-commercial/homegrown
 - “What is operating system/computing environment?” (“Contextual - internal” section)
 - “Are there unusual or rare carriers or file format types that have been identified as part of the collection (we may not be able to accept/may need to refer certain rare formats we don’t have the capacity to properly preserve)”
- [UVA Software Questionnaire \(V1.0\)](#)
 - Use of the questionnaire triggered when accession is identified (through the checklist) as being in need of “early collection and preservation attention”. Checklist and Questionnaire are linked together.
 - “What operating system did you use with this software?”
 - “Is there other software that can also render the files you may have produced with this software?” Contributes to the donor’s perspective on how important it is to use the original software to interact with the digital object.
 - From the [UVA Final Report](#), about the Software Questionnaire: “The questionnaire was then reviewed by legal experts and experts in the field of archives and software preservation. It provides a way to begin a conversation and investigate important preservation and access considerations prior to or during the acquisition process with donors and researchers.”
- [GT oral history interview sample questions](#) (on pg. 3) could easily be adapted as a software questionnaire similar to UVA’s.
- [UI Guidelines for Donating Digital Materials](#)
 - “Preserving digital materials is dependent upon accessing and rendering the computer files which are also dependent upon layers of hardware and software. If you have original hardware and software that you used to create the files and no longer have a use for it the repository may be interested in retaining it.”
 - “The repository will take appropriate action to preserve your files. In some cases complete preservation – where the functionality, original look and feel of the document, and other features are retained - may not be possible. Our various



levels of digital preservation depend upon available resources and appraisal decisions made by the repository staff.”

- [UI Pre-Accession Preservation Appraisal Report](#):
 - Long-term preservation challenges integral to the appraisal process:
 - 1.4.3. Content locked in proprietary or unusual file formats (file format will weigh heavily in the assessment; see 1.4.4)
 - 1.4.4. Unusual file systems/operating system environments/data encoding
 - 1.4.5. Little to no documentation about the content creation process including
 - 1.4.5.1. computing environment in which content was created
 - 1.4.5.2. applications and versions of applications used
 - 2.2 - 2.5 give concrete criteria for identifying and triaging files based on their format. The File Format Analysis section identifies quick ways to at least categorize files based on their apparent format (or file extension). From there, what will be necessary to access the content in unusual or unrecognizable file formats?
 - From [UI 2019 Preservation Week script](#):
 - 9.d: The content appraisal argument for emulation. How can we reliably know how to appraise if we can't interact with the content in its native environment?
 - 9.g.ii: again, more of an appraisal tool, where we are re-appraising (at some point in the future) and attempting to assess the extent of information loss. This can only be done reliably with emulation as a baseline (not a perfect baseline, but as Tracy notes, it's "closer to the native environment").
 - The [UI Hembrough Software Collection Project Plan](#) applies the concept of enduring value to software
 - Also: "Manufactured software vs. Copied software"
 - [UI Data Interest Group presentation script \(2019-11-20\)](#):
 - 12.a.2: "What information they [curators] gather can influence preservation outcomes." Preservation outcomes depend on the information curators gather. Could be extended to appraising archivists as well.
-

Software use expectations

At a general level, a few FCoP cohort participants articulated how users might use legacy software to access software-dependent objects. Some posed questions about how using software might work, especially in emulated environments. These high-level statements and inquiries are a good introduction to the other categories of use that emerged in the FCoP documentation.



References:

- Slide 23 of the [UVA Shared Notes](#): “How will troubleshooting info be presented, for the user who is viewing an object?” Also: “many users will never have seen the software before, won’t know how to interact with the objects”.
 - Slide 24 (UVA Shared Notes), Technical Responsibilities, section reiterates the above sentiments about unfamiliarity with legacy software: compares to foreign language competencies.
 - Slide 24 of the UVA Shared Notes: “researchers value speed, ease of navigating the materials, and comfort with using the software - more than working with emulated environments.”
 - [UI Data Interest Group presentation script \(2019-11-20\)](#):
 - 7.c: Scott Schwartz, curator of the Sousa Archives at the University of Illinois: “as close...as possible” ... “as close as we can get” to the original native environment.
-

Use cases

Several documents produced by the FCoP cohort provide clear use cases for software preservation and emulation, articulated within the cohort participants’ local context, but often relevant for others as well. Some are specific walkthroughs of a use, while others are more general. Some follow the user of the software itself, while others concern the professional working with the software so that it can be more effectively employed by an end user in the future. Taken together, the FCoP use cases represent a significant contribution to the empirical understanding of software preservation.

References:

- The images on slides 13-14 of the [UVA workshop Intro deck](#) summarize the archives use case for software preservation and emulation..
- UVA Archival Description Strategies for Emulated Software (<https://docs.google.com/document/d/1YI6sMZ2lIrC4aiSMJIWEbSv8MzVT4JB9cfNfxRv3zc/edit?usp=sharing>):
 - “description strategies related to born digital content dependent on software for access..... as well as the description of the software itself.”
 - Different description options demonstrate flexibility traditionally accommodated in archival description, now applied to software and groups of software-dependent digital objects, along with descriptions of how to access them.
- [UI Data Interest Group presentation script \(2019-11-20\)](#):
 - 10.b: highlights the amount of legwork (especially on the internet) that must take place to get an emulated environment going. For instance, after finding an



appropriate Firefox version, the presentation recounts “I was also able to download a standalone version of the Adobe Flash Player to install in the Windows 7 environment.”

- From [UI 2019 Preservation Week script](#):
 - 12.c: “Scott [the curator] equates having born-digital production files to having access to a composer’s notebook where a researcher may gain additional information about what creative choices were made when composing or producing audio works.”
- 2 use cases related to preserving access to student work at GT:
 - <https://apps.library.gatech.edu/retrotech/cs2110>
 - <https://apps.library.gatech.edu/retrotech/cs2261>
- [UA IDCC paper](#), the operational mode (fourth paragraph of pg. 5): “people talked about the need for preservation in terms of “I need to do X with software Y and I don’t care how that happens”, contrasting with more conceptual and methodical approaches in the digital preservation community (“software Y needs to be carefully preserved to enable all of W, X, Z”).”
- [UA Software Preservation 101 Module](#): Research Software Perspective (#10-11)
- UA Preservation and Emulation Best Practices: A Seminar on Born-Digital Media Preservation slide deck (<https://osf.io/zf3wb/>) provides a solid walkthrough of the process from disk to emulation. Specific to the use case where software is stored on a disk..
- [UVA Final Report](#), pgs. 3-8
 - Pg. 3: “This use case not only reflects many of the more recent hybrid collections in our archives and special collections...” and, can be seen as an example of a larger set of collections, not entirely unique.
 - Same paragraph as above: “Even though this unique collection consists of digital materials that are only 10-20 years old, many unique files were already incompatible or “too old”, as the error message above says, to even open in available modern viewers in 2020.” We have entered an era where archival materials that are only 10 years may be “too old” to access. The software itself is giving us this message.
- [UI High-Level Workflow/Digital Curation Triage](#)
- [UI Use of Emulation-as-a-Service-Infrastructure \(EaaS\) for Preservation and Access at U of I Libraries](#): Provides an account of micro-level use of emulation at UI, as well as how UI sees emulation fitting into the UI Libraries programming in the future.
 - Local supplement ([Using the Emulation-as-a-Service-Infrastructure \(EaaS\) - Local Supplement](#)) provides a justification as well as a walkthrough of the process of emulating. Also includes documentation of errors. Not specific to EaaS.



- UI [illinois-fcop-software-collection-inventory-worksheet.gdoc](#): starts with inventorying the software-dependency in a collection, and then building out documentation about the software itself from there. Anticipating needs based on the collection material.
 - UI InstallReport.docx files (with screenshots and detailed steps, using DOSBox to emulate the appropriate environment):
 - [WordPerfect 6.0](#)
 - [Cakewalk Apprentice](#)
 - [Cakewalk Pro 5](#)
 - [CorelDRAW 4 Demo](#)
-

When to test?

While collecting software, and preserving the software's bitstream over time, are important first steps for software preservation, the information professional attempting to integrate software preservation into existing workflows will need at some point to try to run the software to ensure that it can be used. But where in the workflow should these test runs take place? The FCoP cohort documented varying approaches.

References:

- [UA Workflow Documentation \(2020-03-10\)](#): Pg. 6 (Access stage): "In both cases, prior to making the items available, they are tested to ensure they function. Testing is done at the Access stage and not the Processing stage due to the volume of material processed by the Archive. If the item does not function, a repair may be attempted by a staff member at this time."
 - UI: during transfer from current media to the digital repository ([Hembrough Software Collection Project Plan](#), bullet point #4); test to make sure the software is "readable"
 - [UI 2019 SAA Popup script](#):
 - Slide 6: "For example, running ProTools requires that we use a hardware authentication method. Also, the ProTools files found within the Wyatt collection can be rendered in the current version of ProTools. We've thus decided to, at present, provide unemulated reading room access to the Wyatt collection and emulate at a later time should it make sense to do so. We've documented important information about the ProTools files to better facilitate emulation or other modes of access." Essentially, the answer to when here is: if we can test it with contemporary software, test then, and document.
-



User interactions with EaaS

The browser-based Emulation-as-a-Service (EaaS) platform was the primary emulation tool used by the FCoP cohort, and the documentation produced by the cohort contains several accounts and illustrations of different user groups interacting with EaaS. There are even live demos that are currently (as of June 2020) are available for people to gain their own experience with EaaS. One thing that emerges from these interactions is that (this very particular type of) software emulation is not just about running an old software title. Rather, EaaS and the emulators it deploys make it possible to emulate an entire software environment, the whole stack of hardware, operating system, peripherals, disk storage, files, and, yes, software applications.

References:

- Emulation diagram on slide 5 of the Intro slide deck (<https://mfr.osf.io/render?url=https://osf.io/pj4ba/?direct%26mode=render%26action=download%26mode=render>) provides a visual aid of how an archivist interacts with EaaS.
- VectorWorks screenshot (slide 8 of [UVA workshop Intro deck](#))
- Slide 12's (from [UVA workshop Shared Notes](#)) demo of interacting with an object in EaaS includes Durbin's discussion of saving environments (how is that emulation functionality considered in the framework of software preservation?), Durbin thinking forward to use cases, and the question from the audience noticing EaaS performance lag.
- Slides from [UI Data Interest Group presentation slide deck \(2019-10-23\)](#):
 - 7
 - 9 (EaaS screenshot with error)
 - 10
- Slides from [UI 2019 Preservation Week slide deck](#):
 - 12 (EaaS screenshot of composer's file directory)
- Slides from [UI 2019 SAA Popup slide deck](#):
 - 30-31
- Live demo of ArchiCAD 4.5 at <https://apps.library.gatech.edu/retrotech/pyburn/>! (circa 1990s)
 - Complete with example of how to integrate EaaS within an organization's web design framework.
- Other GT live demos:
 - <https://apps.library.gatech.edu/retrotech/cs2110>
 - <https://apps.library.gatech.edu/retrotech/cs2261>
 - <https://apps.library.gatech.edu/retrotech/ribbit>
- GT live demo "landing pages" (such as <https://georgiatech.softwarepreservationnetwork.org/landing-page/?id=5309aee4-ae96-4>)



[bea-9c18-6e2cc3ef87f8](#)) provide the context that is almost always necessary to navigate legacy software environments. How can this context be expanded in future work?

- GT [Cohort4Lib slide deck](#) (Slides 11 and 12) have good illustrations of Georgia Tech's emulation efforts.
 - [UA EaaS Sandbox Workflow \(2020-03-05\)](#): operating systems turned into environments, its software turned into objects, and adding files to those environments
-

Snapshots of software preserved

Here and there the FCoP documentation provides brief glimpses into concrete examples of software preservation and emulation, without much interpretation from the cohort participants. It is nevertheless worth highlighting these glimpses, because there have been so few of these examples to date.

References:

- Slide 10 of [UVA workshop Intro deck](#) includes a file system screenshot and photographs of software media boxes.
 - Slide 2 of [Julia Kim's slide deck](#) (UVA workshop): Sheepshaver in action, emulated art
 - Slide 6 of Julia Kim's slide deck (UVA workshop): file formats and file types
 - Photoshop 6.0 screenshot on slide 8 of Julia Kim's slide deck (reminder of the difference between then and now)
 - Slide 9 of [UVA workshop Access and Use slide deck](#): VectorWorks screenshot
 - All of Q&A section in pages 18-19 of [UVA workshop Shared Notes](#)
 - [UI Data Interest Group script](#) (8.b.vi): had achieved a laptop emulation of Manion (a composer)
 - From [UI 2019 Preservation Week script](#):
 - 5.b and 5.c: info about Adobe CC and ProTools and Tracy's experience with their backwards compatibility issues
 - 12.e.ii.2 and 12.e.ii.3: provide updated use case information that was not included in the DataInterestGroup presentation. The composers are Peter Micahlove and Scott Wyatt.
-

Software is physical too

While software, by definition, has little tangible existence, there is always a medium by which it is carried, stored, and executed. And there are other physical components that accumulate alongside software that can have substantial significance for how software is used over time.



Some of the FCoP documentation identifies the places where software assumes a physical presence.

References:

- [UA Workflow Documentation](#):
 - Pg. 5 (Preservation stage): “To ensure the material remains in an as-received condition (until a determination can be made if the condition warrants repair), the Processing stage of the workflow includes explicit instructions (especially for interns) to not attempt to repair object or remove stickers, packaging, etc. The only exception is the removal of cellophane wrapping which can damage certain kinds of media over time.”
 - Also in UA workflow doc: what “further peripherals/equipment” necessary for preservation?
 - [UVA workshop Technical Overview deck](#), slide 5: Software vs. Hardware graphic poses a continuum between analog and digital.
-

Advocating the value

Digital preservation in general has long struggled to articulate an immediate value proposition. What does preservation do for us now (nevermind what it will do for us later)? Software preservation is no different. The FCoP cohort engaged in a variety of advocacy efforts, both within their organizations, and out in the pre-pandemic field of professional conferences. They considered different audiences: an archivist might care about software preservation for different reasons than a commuting salesperson listening to a podcast. Along the way, the University of Arizona cohort participants explored the possibilities of community-building in particular. If software preservation and emulation is to become a mainstream activity, stakeholders will need to be persuaded that it is valuable.

References:

- [UA IDCC paper](#):
 - Related to the 2nd paragraph of the “Linking...” section: establishes that they were able to move from video games to “software preservation writ large”
 - 3rd paragraph: “... most participants seemed to find it easy to jump from games to topics further afield.”
 - Last paragraph of “Linking...” section: “it appeared that software preservation resonated with individuals when it was encapsulated within the idea of active use.”
 - Solution in “Sustaining UA-SPIG” section argues for growing software preservation communities within existing community frameworks where “latent” interest in software preservation may exist.



- “An informal poll of regular attendees at one Coffee and Code session revealed that there is interest in software preservation-related conversations, especially as they relate to tackling challenges related to the problem of preservation for research reproducibility.” (page 7)
 - [UA Software Preservation 101 Module](#): Slide #8, Why?
 - [UA Workflow Documentation](#): Outreach section could be applied broadly. What if we opened up a testing sandbox where we could crowdsource the rendering of software in EaaS? Also contributing to metadata and technical documentation? Video walkthroughs?
-

Software preservation operates in socio-technical infrastructures

As technology increasingly determines the work of information professionals, any attempt to integrate software preservation into existing practice will necessitate integration into technological infrastructures. The array of systems that are used to provide access to information, to describe the information, and to preserve the information, can be daunting. And social infrastructures must be considered as well (after people use the systems, and they maintain them too). The FCoP cohort ambitiously ventured into this integration work, and documented their work with diagrams, README files, and statements of collaboration.

References:

- Diagram on slide 17 of [UVA workshop Technical Overview](#) presentation incorporates several components of UVA’s infrastructure, and demonstrates how EaaS can be a part of that infrastructure.
- <https://github.com/uvalib/curio/> briefly explains the Curio interface (UVA)
- Slide 12 ([UVA workshop Metadata and Description deck](#)) provides crosswalk between EaaS model and UVA’s ArchivesSpace profile
- MARC crosswalk to ArchivesSpace profile, for software (linked from slide 15 of [UVA workshop Metadata and Description](#))
- From [UI 2019 SAA Popup slides](#):
 - Slide 33 (“The big picture: The preservation of software and software-dependant digital materials are more than technological problems”)
- [GT EaaS landing pages](#), incorporating EaaS into the RetroTechOnline website, and placing the emulations within a larger story and context (and infrastructure picture)
- [GT Final report](#), pg. 6: “Based on our main goal, we invested most of our efforts into two phases of the digital curation lifecycle in particular--collection development and access.”
- [GT README file from the Laravel github repo](#) could be something that other ArchivesSpace users could try out.
- GT [Cohort4Lib slide deck](#) (Slides 11 and 12) have good illustrations of the technical ecosystem involved in RTO.



- [GT final report](#), 2nd paragraph of Challenges and Lessons Learned section discusses integrations into other systems, and GATech’s thinking about pursuing that approach.
 - [UA IDCC paper](#), pg. 4: “Among the inaugural visitors were data librarians, physicists, electrical engineers, language researchers, archivists, and historians.”
 - [UVA Final Report](#):
 - pg. 4: “leaning on the expertise of our colleagues whose work may not fall under what people may currently think about as contributing to “software preservation” or “digital preservation”. This includes expertise in metadata, information policy, architecture, public services, and reference.”
 - Page 6: “In developing the system design for our project goals, we started first from the ideal vision for access and then worked backwards to determine the needs to support it.”
 - The amount of integration in the Technical section may prove to be an ideal to which other organizations aspire.
 - UVA Systems and Metadata Diagram:
<https://docs.google.com/drawings/d/1KvL1oVExSMDULB7k5R14tfUaqM-IkQdlQa6Yzt09Qwg/edit>
-

The law and software preservation

An often stated barrier for using legacy proprietary software, some amount of confidence has recently entered the domain of software preservation, and the FCoP cohort’s documentation has provided a few good examples of the state of affairs and how to leverage new tools like the ARL Code of Best Practices in Fair Use for Software Preservation.

References:

- [Brandon Butler’s UVA workshop slide deck](#) is a good distillation circa August 2019. It also provides a valuable summary and potential entry point for people learning about this topic for the first time.
 - From [UVA Software Questionnaire \(V1.0\)](#): question about license keys is important, especially Butler’s distinction regarding physical vs. cloud credentials. Begs the question: is there a use in maintaining a collection of license keys?
 - GT’s review of the ARL Code of Best Practices, which provides an implementation of the ARL Code of Best Practices in Fair Use for Software Preservation, documenting the analysis that will guide future access strategy:
<https://drive.google.com/file/d/1ml7D0xCX03nKoGgIKDhBTxfjwTVroKo/view>
-



The wonderful and frightening world of metadata

Project documentation from the cultural heritage sector would be incomplete without some sort of metadata documentation. The FCoP documentation is no exception. Specifically, the University of Illinois (through the Digital Content Format Registry) and the University of Virginia (through their emulation integrations with ArchivesSpace and a local discovery interface) considered many ways in which robust descriptive and technical metadata for software and software emulation instantiations would facilitate more effective software preservation and emulation work, and they documented their extensive attempts to create those descriptions.

References:

- [UVA MARC Field Look-up for Software Description in ArchivesSpace](#)
- [UVA Archival Description Strategies for Emulated Software](#)
- [UI Medusa Digital Content Format Registry](#): What it is and How to use it: (as a way to understand the DCFR):
 - The tabular descriptions in “General Digital Content Format Registry Entry Fields” seem useful for people looking for examples on how to describe/document file formats, rendering experiences, and normalization paths.
 - Pg. 1: “However, given the multiplicity of ways in which certain types of files may have been produced, it is unlikely that any reference resource will be able to account for the many varieties that may occur in a particular archives hence our interest in maintaining a knowledgebase informed by local collections.”
- From UI [Hembrough Software Collection Project Plan](#):
 - 3.a: “Note on controlled list: In consultation with a colleague at Stanford University, Charlotte Thai, who is creating metadata for the Steven M. Cabrinety collection of historic micro computing software in partnership with the National Institute of Standards and Technology (NIST), it was determined that no authoritative list of software categories is currently available. Charlotte indicated that NIST has developed a local controlled vocabulary which is in use in the Cabrinety finding aid (see: <http://www.oac.cdlib.org/findaid/ark:/13030/kt529018f2/>).” [emphasis added]
 - 3.b: “The UIUC controlled list of software categories is based on the Cabrinety/NIST categories as well as those established by [the creator] within his disk labeling schema.”
- [UI Software spreadsheet field reference \(data dictionary\)](#): a starting point for basic item-level description of software titles
- [UI Data Interest Group presentation script \(2019-11-20\)](#):
 - 11.a.iv: Digital Content Format Registry: “The research focus of this tool is to document local knowledge gathered about how to identify and render challenging file formats – particularly formats that present challenges including being associated with a specific version of proprietary software.”



- 11.a.vi: “Information about these formats also tends to be weak or non-existent in international or large-scale file format identification tools”
 - 12.a: “creation context” and “files of interest” are data points that could be collected by curators. A type of information that cannot be automated.
 - 12.a.2.ii: “document use” --> related to “creation context” concept
 - [UI 2019 10 SoftwarePreservation MetadataCaptureLocations.xlsx](#):
 - 3 columns for the DCFR seem particularly noteworthy:
 - New digital content format
 - Rendering profile
 - Normalization
 - The 4 EaaS columns provide a walkthrough of how to add and use a software object to the EaaS interface. A running theme through these columns is that the user needs to know the OS and hardware environments in advance of attempting to create an EaaS environment where the software and related files can be rendered.
 - [UVA Final Report](#), Page 5: “we learned that both Wikidata and WorldCat were excellent sources for tracking down technical metadata for obscure or deprecated commercial software titles.”
 - [UI 2019 SoftwareInstall Workflow BLANK.xlsx](#): qualitative technical metadata template
 - [UI ReadmeTemplate example.txt](#): low barrier technical metadata format
 - [UI illinois-fcop-collections-software-inventory-spreadsheet.gsheets](#): software inventory based on creator use. Another example of UA’s operational mode of thinking?
-