Saving Software and Using Emulation to Reproduce Computationally Dependent Research Results

Euan Cochrane  
*Yale University*, euan.cochrane@yale.edu

Limor Peer  
*Yale University*, limor.peer@yale.edu

Ethan Gates  
*Yale University*, ethan.gates@yale.edu

Seth Anderson  
*Yale University*, seth.r.anderson@yale.edu

Follow this and additional works at: [https://elischolar.library.yale.edu/dayofdata](https://elischolar.library.yale.edu/dayofdata)

Part of the Architectural History and Criticism Commons, Architectural Technology Commons, Construction Engineering Commons, Digital Humanities Commons, Engineering Commons, Historic Preservation and Conservation Commons, Industrial and Product Design Commons, Interdisciplinary Arts and Media Commons, Life Sciences Commons, Medicine and Health Sciences Commons, Modern Art and Architecture Commons, Physical Sciences and Mathematics Commons, and the Social and Behavioral Sciences Commons

Cochrane, Euan; Peer, Limor; Gates, Ethan; and Anderson, Seth, "Saving Software and Using Emulation to Reproduce Computationally Dependent Research Results" (2019). *Yale Day of Data*. 1.  
[https://elischolar.library.yale.edu/dayofdata/2019/posters/1](https://elischolar.library.yale.edu/dayofdata/2019/posters/1)

This Event is brought to you for free and open access by EliScholar – A Digital Platform for Scholarly Publishing at Yale. It has been accepted for inclusion in Yale Day of Data by an authorized administrator of EliScholar – A Digital Platform for Scholarly Publishing at Yale. For more information, please contact elischolar@yale.edu.
Euan Cochrane, Seth Anderson, Ethan Gates
Yale University Library
https://www.softwarepreservationnetwork.org/eaasi/

Motivation
Using digital data necessarily involves software. How do we ensure long-term usability and computational reproducibility? How do institutions address usability challenges such as, license restrictions, legacy software, code rot, and dependencies? At Yale University, a team in the Library is working with the ISPS Data Archive to look into the application of a novel approach to emulation as a potential solution.

Software-dependent research reproducibility problems
• Original code tied to legacy software
• Legacy software that is no longer available
• Proprietary software that is difficult to package with reproducibility packaging tools
• The packaging runtime (e.g. Docker, Reprozip) is no longer supported on modern operating systems

Traditional solution
Author or archive re-writes the code; requires updating the scholarly record with modified code
Or, end-user modifies the code to use new function, if available

EaaSI solution

EaaSI for data curation
• Select existing software environments or build one from software installation media and online software sources
• Add code and/or data, and/or packages and save a derivative “content environment”
• Test the reproducibility
• “Publish” (publicly or privately) the content environment and get a handle and DOI
• Add link to archive’s discovery tool

EaaSI for researchers
• Access and reproduce legacy research using original code
• Seamlessly access proprietary legacy software
• Export data into open or more modern formats
• Access legacy databases in real time via secure proxy

Keep older software accessible!!!

EaaSI for Funders

Motivation
• Without original software even simple data objects like this Excel v3.0 (.xls) workbook can be distorted or have missing content
• In Quattro Pro running in Windows 95 (image 3) the workbook is missing the annotation explaining the data, an annotation that is visible when opened in Excel 97 in Windows 98 (image 4)
• Use the UVI’s matching API to confirm archived digital objects have the original software available in the EaaSI network