Demo – An Integrated System-Preservation Workflow

Isgandar Valizada, Klaus Rechert, Dirk von Suchodoletz, Sebastian Schmelzer
Chair in Communication Systems
University of Freiburg, Germany

ABSTRACT

Preserving a full computer system for long-term access is in some cases the option of last resort. In these situations the system consists of complex workflows and tool-chains paired with custom configuration such that the dismantling of individual components is too time consuming or even impossible to be carried out. For such situations a defined *system-preservation workflow* is required, ideally integrated into a digital preservation framework to ensure future rendering ability and quality.

This demonstration presents an integrated system-preservation workflow, designed to be performed by non-technical users and to ensure long-term access (e.g. through emulation) with guaranteed system properties. Furthermore, the workflow leverages a distributed framework, enabling on-site preservation tasks, while making use of a common shared knowledge base and ready-made access preview technologies.

1. MOTIVATION

In certain situations system-preservation of a whole computer system is inevitable. For instance in a scientific environment certain computer workstations have grown over time. During a perennial research a project's fluctuation of personnel is common and therewith system knowledge is volatile. Complex software workflows have been created, usually involving tailored software components and highly specialized tool-chains paired with non-standard system tweaks. Rebuilding the system from scratch is a time-consuming task involving manual labor, and it requires a technically skilled operator. Thus, preserving the complete workstation is more economical and, if carried out properly, full functionality of the system can be retained with minimal effort.

Through a system-preservation process an image of the complete content of a computer's hard disk is made. This image, a virtual hard disk, can then be run again with virtual hardware, i.e. virtualization or emulation technology. While the technical problems of imaging as well as re-enacting of the workstation are solved in principle [1], a practical and usable solution including defined workflows and framework integration are still missing. Emulation projects like KEEP primarily focused on single object workflows and not yet providing the required functionality for system preservation.

The challenges of system-preservation tasks are manifold and mostly of technical nature. The tasks to be carried out require technical expertise on the targeted system, e.g. booting the system in a read-only mode to prevent system modifications and inconsistencies during the imaging process. Furthermore, the image needs to be post-processed and enriched with meta-data describing its original hardware environment.

To re-enact and test the preserved system image, knowledge of current emulation technologies is necessary. This knowledge may also include pre-processing steps to be carried out on the original system before imaging. Such tasks may include demotion of the system to default graphics drivers or disabling of external dependencies during the boot-process (e.g. mounting network shares, connections to license servers, etc.). Such external dependencies may be restored in the emulated environment in a post-processing step.

Therefore, the goal of providing a framework with an integrated workflow for system preservation is to enable non-technical users to prepare and perform a system preservation task and finally test the result produced before submitting the data to a digital long-term preservation repository. Thus, provided a successful run of the *system-preservation workflow*, a functional image of a real-life computer system is guaranteed to run on current emulator software. For long-term accessibility proper preservation planning is only required on emulation software. By providing immediate feedback, the owner of the preserved computer system is able to verify whether the subjective and objective significant properties were preserved.

2. TECHNICAL WORKFLOW

The system-preservation workflow and associated tools are an integrated part of the bwFLA framework,[1] designed to support the user in functional preservation tasks, especially leveraging emulation for object access. This demonstration describes the workflows and tools for system-preservation. As the framework does not provide a dedicated repository for digital object storage, the framework is de-

[1]Developed as a part of the ongoing Baden-Württemberg Functional Longterm Archiving and Access project, *http://bw-fla.uni-freiburg.de.*

signed to interface with OAIS compliant DP frameworks within ingest and access workflows.

The workflow is split into three stages: First, the workstation subject to preservation is prepared and tailored software for the imaging process is generated. In a second step the imaging process is carried out and finally the generated image is tested and enhanced with meta-data.

Preparation

1. In a first step the user characterizes the computer system to be preserved (in the following denoted as "target system") by choosing one of the yet supported operating systems and computer architecture. Handling of not yet supported operating systems will be discussed below.

2. To perform the imaging process, the target system requires certain technical functionality's, e.g. USB-port or optical reader (CD-/DVD) and the ability to boot removable media. Furthermore, a (standard) network adapter is required. Following, the user is interactively questioned if the target system meets these requirements. Depending on the choices made, the imaging process is prepared either to be carried out on the target system, or on a different (modern) computer system. The latter option requires dismounting the hard-drive of the target system.

3. To ensure better compatibility with hardware emulator software a knowledge-base on different operating systems regarding their compatibility with emulators and hardware dependencies is part of the bwFLA system-preservation framework.
   The user is presented with known issues based on his previous selections and step-by-step guides describing user-actions to be carried out on the target system. Common examples may include reducing screen resolution, reducing hardware driver dependencies or preparing network connections.

4. Based on the user's choices a specially tailored bootable image is generated and made available for download. The bootable image is either to be written on a USB pen-drive or CD/DVD. The boot-media contains a tailored Linux kernel suitable for booting on the target device, network configuration, and necessary credentials to connect to the bwFLA framework, e.g. to stream/upload the generated image in an automated way.

System Imaging

At this step the client uses the newly created bootable medium to boot the machine on which the preservation workflow is to be performed. After booting the bwFLA imaging system on the target system an automated process starts the imaging process, the gathering of the relevant hardware information about the target machine, and the uploading of this data to the bwFLA framework.

The only interactive choice allows the user to select the drive to be preserved, if multiple options are available. By default the standard boot-device is chosen. Currently, only complete block device preservation is supported. However, the ability of selective partition preservation is planned for

Figure 1: Target system booted from USB pen drive starting the system preservation process.

future work. During the imaging process the user is able to follow the progress on the Web-page that is monitoring the upload of the image data.

Verification and Submission

In a final step the disk image generated is post-processed (if required) for an appropriate emulator type, chosen based on selection made during the preparation process and information gathered during the imaging process.

Finally an bwFLA emulation component is invoked with the preserved system image and presented to the user. If the user approves the result, the image is submitted together with generated technical meta-data to a DP repository for further processing.

3. RESULTS & OUTLOOK

The workflow has been tested mainly on x86-based systems, mostly due to availability of high-quality emulators. However, adding new, yet unsupported systems is easily possible as long as the original workstation is able to boot a Linux kernel and connect to the network. In cases of older (potentially highly integrated systems) such as Notebooks without CD-ROM drives, USB-boot capabilities and even without network connection the external imaging workflow remains as a working option. However, in such a case the description of the original system remains to the user and has to be carried out manually.

With the integration of system-preservation workflows into a distributed digital preservation framework, a common knowledge base on preparing, imaging and re-enacting ancient computer system can be built, thus providing step-by-step instruction even to non-technical users. Having the feedback loop in available, the owner of a workstation subject to system-preservation is able to describe and approve desired properties of the system for later access.

4. REFERENCES

[1] D. von Suchodoletz and E. Cochrane. Replicating installed application and information environments onto emulated or virtualized hardware. In *Proceedings of the 8th International Conference on Preservation of Digital Objects (iPRES2011)*, pages 148–157, 2011.