Emulation Reading Room Prototype

Sebastian Schmelzer
Albert-Ludwigs University
Freiburg, Germany
sebastian.schmelzer@rz.
uni-freiburg.de

Dirk von Suchodoletz
Albert-Ludwigs University
Freiburg, Germany
dirk.von.suchodoletz@rz.
uni-freiburg.de

Klaus Rechert
Albert-Ludwigs University
Freiburg, Germany
klaus.rechert@rz.
uni-freiburg.de

ABSTRACT

The electronic collections of today's libraries, museums and archives are growing and increasingly have a more relevant role in the holdings. Memory institutions must address users' need to access a widening range of digital artefacts. Often the formats of those artefacts are outdated and they cannot be run or rendered on today's systems any longer. This is where emulation can provide the required digital environments suitable for a given object type. Practical research is being done at Freiburg University for the Open Planets Foundation on how to integrate different emulators for a number of original environments into a single graphical desktop. In this case study, options for future reading room systems like stateless Linux workstations are evaluated and prototypical implementations are demonstrated.

1. INTRODUCTION

For modern memory institutions it would be desirable to offer their users access to a wider range of different object types within their original environment than currently possible in today's reading room systems, which are restricted to a number of multimedia formats and document types. Many artefacts of the growing electronic collections are outdated, meaning no modern application can render them in a usable and authentic way. These artefacts require appropriate front-ends in order to experience (render) them on today's systems.

The emulation of outdated digital ecosystems is the answer for accessing and experiencing an increasing variety of (otherwise inaccessible) file types. Compared to traditional file loading or program starting, the access to ancient artefacts involves a more complex workflow. On average, the visitor to memory institutions is not familiar with past computer architectures. The goal of researching the different approaches is to create different ways to automate access to the desired digital artefact. The concept is to have different prototypes for running such a service, such as "mimicking" the double-click on an object and starting an "automagic" loading into
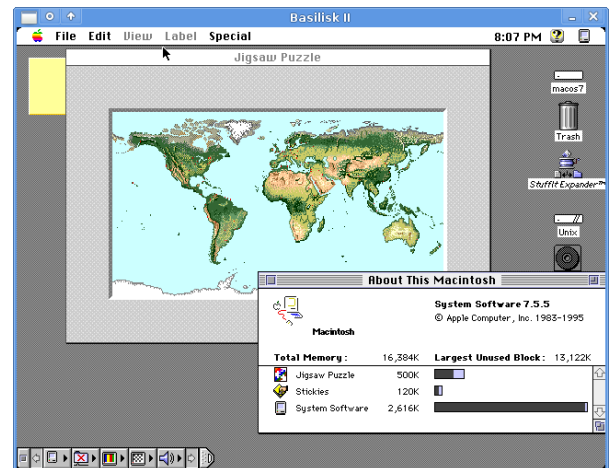
Figure 1: Mac OS 7 was a popular operating system used in many companies and institutions as well as by private individuals.

the proper application. This automation will assist the average user of a reading room digital object access machine, who is not very familiar with past GUI concepts and applications. Instead of simply loading a certain file into an appropriate application, a more complex procedure is to be executed in order to wrap the object and transport it into the original environment. Then the environment has to be turned on and the object loaded into it's original creating or viewing application.

2. ACCESS TO DIFFERENT EMULATORS

There is a large number of emulators, mostly programmed by enthusiasts available as open source. Often, more than one emulator could be used for a certain digital ecosystem of hardware and software. Especially for the x86 architecture, there exists a wide range of commercial and free emulators and virtualization tools like VMware, VirtualBox, QEMU, Dioscuri or DosBox. To offer the user easy access to the different combinations of original environments and emulated hardware, we produced a small application (Fig. 2) that reads the metadata from an XML file in order to provide a short description plus the machine and firmware information needed for the original environment to start. Thus it is easily possible for the user to compare the characteristics of the same original environment in different virtual machines or evaluate the rendering or execution of a wide range of digital
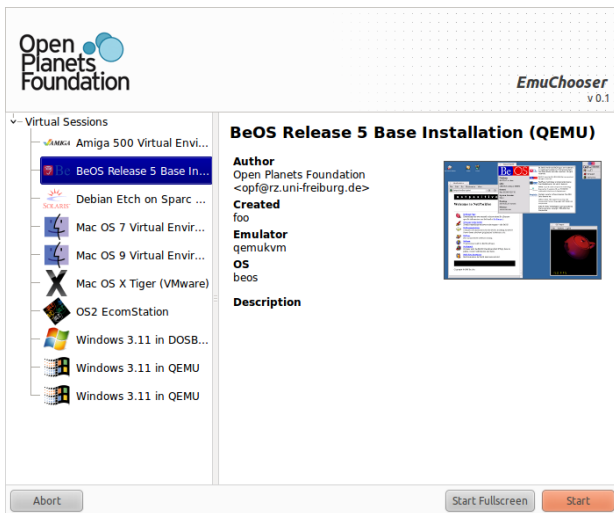
Figure 2: Selection to choose different original environments

artefacts [2]. The application is executed on the Linux platform, which is a very versatile base for hosting a wide range of emulators without incurring additional licensing costs in order to run a larger number of reading room workstations. Providing access to a wide range of different outdated computer platforms is the first part of an access strategy. As many users are no longer familiar with outdated architectures and user interfaces, the idea is to prepare the original environment, configure the emulator and load the artefact into the started environment automatically. After loading, the user information is provided with information on how s/he can use the artefact or navigate the original environment.

3. ASSISTED CREATE VIEW

Based on the concept of view paths, which formalize the pathway from a certain digital object like an Amiga computer game or Word Perfect document into its execution or rendering environment, we explore methods as to how this could be (partly) automated. A multitude of methods are being researched, ranging from altering the original environments in different ways to abstract automated handling of user interaction as discussed in [3]. Thus, ongoing research is looking into the application of automation procedures provided by the original operating systems and applications. This alters the approach chosen by Brown and Woods [5], who introduce custom binary executables and *AutoIt* scripts – a Windows GUI scripting package – into the original environments to automate the access to certain types of objects. Another option is to use VNC interfaces provided by the emulators or virtualization tools to run interactive environments unattended. This mechanism could be deployed to prepare the original environment in a way that avoids the need of any specific operating system or application knowledge. The system is developed in an abstract way in order to handle arbitrary environments and emulators. The concept of automated user interaction is broadened by looking into ways of providing the emulators with appropriate interfaces [1] or to explore the options of workflow automation more deeply by using the monitor interfaces provided by a number of emulators. These interfaces help to operate many of the emulator functions like (un)mounting removable media or sending keystrokes or mouse events in order to run certain actions unattended.

Many useful tools and applications are provided with a standard Linux installation: The preparation of floppy images or a wide range of container images with different filesystems is not a problem. Most of the relevant filesystems are directly supported by the Linux kernel which makes import and export of artefacts into and from the original environments easy. Additionally, converter tools like "qemu-img" or a wide range of decompressors help to program the several workflows involved. For the prototypical implementation we use the standard Linux scripting tools, or we programmed a small application such as the emulator chooser (Fig. 2). This tool provides the appropriate command lines pointing to the relevant resources like original system images and firmware roms or generates the configuration files needed for a range of virtualization tools. It could easily be extended to acquire license information from a centrally managed system. All sessions are run in non-persistent mode so that users can freely interact with the original systems without ruining the installations.

4. CONCLUSION

The direct access to many outdated environments still offers the most authentic experience. When provided within the premises of the memory institution, an emulation access workstation might be more favorable compared to remote access. It allows a more real-time, full screen experience with audio output and direct access to the peripherals if required. A stateless Linux system is an easy way to provide a reliable service enabling the users to fully interact with original environments run using today's hardware and does not risk rendering them unusable. The research described here does not focus on any file type detection during the whole procedure, but presumes that this information is available and encoded using the metadata of the files loaded. Besides a proper file format base and tool registry, a software archive of outdated software components is required [4].

5. REFERENCES

[1] Evgeni Genev. VNC Interface for Java X86-Emulator Dioscuri. Online, http://hdl.handle.net/10760/15102, October 2010.

[2] Mark Guttenbrunner, Christoph Becker, and Andreas Rauber. Keeping the game alive: Evaluating strategies for the preservation of console video games. *International Journal of Digital Curation*, 5(1), 2010.

[3] Klaus Rechert, Dirk von Suchodoletz, and Randolph Welte. Emulation based services in digital preservation. In *JCDL '10: Proceedings of the 10th annual joint conference on Digital libraries*, pages 365–368, New York, NY, USA, 2010. ACM.

[4] Maurice van den Dobbelsteen, Dirk von Suchodoletz, and Klaus Rechert. Software archives as a vital base for digital preservation strategies. Online, *http://hdl.handle.net/10760/14732*, July 2010.

[5] Kam Woods and Geoffrey Brown. Assisted emulation for legacy executables. *International Journal of Digital Curation*, 5(1), 2010.