Exhibiting Digital Art via Emulation

Boot-to-Emulator with the EMiL Kiosk System

Dragan Espenschied
Rhizome
235 Bowery
New York, NY, U.S.
dragan.espenschied@rhizome.org

Oleg Stobbe, Thomas Liebetraut and Klaus Rechert
University of Freiburg
Hermann-Herder Str. 10
79104 Freiburg, Germany
{first.lastname}@rz.uni-freiburg.de

ABSTRACT

The availability and accessibility of digital artworks is closely tied to a technical platform, which becomes quickly unavailable due to a fast technical life-cycle. One approach to keep digital artworks performing is to replace physical hardware parts with emulation. Preparing an emulator to publicly display digital art is typically time-consuming and, more importantly, usually done on a case-by-case basis, making each installation a unique and costly effort.

We present an adaptation of the Emulation as a Service framework to be deployed on a self-contained USB-stick, booting directly into a prepared emulated environment. Furthermore, we report from practical experiences using the system in two museum exhibitions.

Keywords

Emulation, Digital Art, Net Art, public access, exhibition

1. INTRODUCTION

With a growing amount of digital artworks relying on outdated hardware to perform, art museums and galleries are struggling to publicly present them for example in historical exhibitions dedicated to a certain period or movement within digital art.

Especially net art, with its requirements for active usage and network connections, has been posing challenges since its inception in the 1990's, with works being difficult to exhibit long before they became historical. While many art institutions have consequently moved digital art into the space of special events, those committed to presenting historical perspectives of digital art have created a demand for sophisticated emulation setups. Preparing an emulator to publicly display digital art is typically time-consuming and, more importantly, usually done on a case-by-case basis, making each installation a unique effort.

When artistic or curatorial intents demand that not only the artworks' computational part is retained, but also the "look & feel" of certain pieces of historical hardware is required (monitors, input devices), exhibitions can grow into hard to manage, very fragile undertakings, placing an undesirable strain on institutions regarding technical and personal resources. The main reason for this is not so much the required hardware, which in the case of net art has nothing unique to it and is easily replaced (e.g. no monitors have to be physically manipulated). Problems of scale rather arise on the computing side, when multiple, technically differing computer systems have to be configured at the software level to behave in the intended way while being replaceable in the case of hardware failure.

To address the aforementioned challenges, this paper presents a technical solution and a workflow based on the Emulation as a Service framework, making use of a range of emulators and guided web-workflows for preparing a specific emulation setup. Furthermore, we present an adaptation of the EaaS framework to be deployed on an self-contained USB-stick, booting directly into a prepared emulated environment. Finally, we report from practical experiences using the system in two museum exhibitions.

2. EXHIBITING NET ART

Net art is an art form with its root in the early 1990's, using mainly the World Wide Web as an artistic medium. Most net art has not been created to be presented in a bricks-and-mortar gallery, but with the Web itself being the point of contact with the audience. As that, net art is one of the least self-contained digital art forms, with lots of complex technical and infrastructural dependencies required for its performance.

Yet different cultural institutions have brought net art into their galleries, due to its cultural significance. Additionally, institutions have the chance to present historical net art that has become inaccessible or distorted on the Web, because of data loss or changes in consumer devices and software since a work was created. Gallery visitors can be presented historically accurate settings, adding legacy software and hardware, something that the Web can not offer.

Three main ways to publicly present net art in physical space have been established very early in the practice's history:

- *The unmodified, connected computer*
  Example: Documenta X, Kassel, 1997
  Off-the-shelf consumer devices, with typical software required to access the web, are used to present artworks. Visitors of the space see the familiar devices and interfaces and are able to fully interact with them. This matches the intended environment for the artworks, but inevitably leads to the audience modifying the setups to the point when they become technically un-usable or don't behave as intended in a very short amount of time. Gallery goers reading their email on gallery computers instead of focusing on the art has been a common sight, and still represents the least

problematic ”unintended use”.

- *Locked-down Kiosk systems*
  Example: net_condition, ZKM, Karlsruhe, 1999
  To prevent the aforementioned ”unintended use”, kiosk systems with very restricted interfaces are used so the audience has no way of modifying the computer set-ups or even ”surfing away” from the artwork. While this makes the exhibition practical, in many cases these restrictions hamper the affect of the artwork, for example by removing visible URLs or common elements the works refer to, like navigation buttons or widgets of the operating system. Given that most net artworks are not created for kiosk systems, there is also no guarantee that they would even perform as intended.

- *Video documentation*
  Example: most art fairs ever since
  In the face of the aforementioned complexity, many institutions fall back on digital video to show any kind of digital art. While this is definitely the easiest approach, for various reasons it is in many cases unable to transport or represent an artwork that was not created as a video.

While institutions have to weigh the pros and cons for each of these presentation forms, legacy artworks add yet another dimension of issues: If an artwork benefits from being presented on contemporaneous hardware, old computers are usually either not available, very hard to maintain, or tend to fail when being used again after a long time of inactivity.

3. AN EMULATION KIOSK-SYSTEM

Recently, emulation frameworks have made great advances, in particular hiding technical complexity and by using web standards for delivery [6]. A technical emulation framework for public display has to be different from web-based emulation setups [2]. Running emulators on local machines (standard PCs) can be an interesting alternative for reading-room setups or museum displays, where cluster- or cloud-computing options are not suitable. For example, when running locally, emulators can provide much better response times then when run on remote infrastructure.

In an exhibition or reading room situation, the emulation setup typically needs to render only a single, specifically prepared artwork. To make such a system cost-efficient, a re-useable technical design is necessary, ideally only disk images and objects should be exchanged and a wide range of emulators should be supported. Furthermore, the technical system should be self-contained, such that it can be used without a network connection or similar additional requirements to the surrounding technical environment.

For exhibiting artworks with interactive components, the physical-technical context can be important, e.g. how the user interacts with the work. This is especially true for historical pieces. Hence, even though using a contemporary computer system (most importantly a contemporary CPU to run emulators) the environment should be flexible enough to support old (or old looking) peripherals or displays. Compared to a web-based presentation, local execution of emulators allows to connect peripherals, such as joystick or printers, different display options, e.g. CRT monitors, projectors etc., and supports an authentic user experience for applications such as games, software based art or net art by providing native fullscreen display and practically zero (input-)latency.

Finally, the system needs to be adapted for public display, in particular protecting the artwork and the installation from undesired manipulation. For interactive works, where a user is even asked to change the environment through usage, the system containing the artwork should be reset for every visitor so they encounter the work in its intended state. Especially for long-term exhibitions, the system needs to be simple to setup (e.g. simply power on a machine) and simple to fix, if the setup has crashed.

4. TECHNICAL DESIGN

In the course of the EMiL project[1] an emulation-based access framework for multimedia objects in libraries and museums has been developed. The EMiL system is an advancement of the bwFLA/EaaS system and aims at integrating with different catalogues and long-term preservation systems. The project consortium consists of the German National Library, the Bavarian State Library, Karlsruhe University of Art and Design and the University of Freiburg. The project is funded by the German Research Foundation (DFG). As a result of the EMiL project the EaaS system has been modified to run within a so-called live-system but using a common EMiL/EaaS codebase.

In particular, the EaaS system – originally developed as a cloud-enabled, web-based system – was adapted to make use of local hardware, in particular running emulators on locally available CPUs, use the machine’s input devices as well as local available graphics hardware and display(s) attached. The live-system is tailored to run directly from a USB stick, such that it boots any standard computer system with no additional preparations or installation requirements. To achieve optimal hardware support, the EaaS live-system is derived from an Ubuntu live-system.[2] As the EMiL live-system was designed especially for library reading rooms or museum exhibitions, all access to the Linux system is restricted by default and users can only select objects/environments to emulate and interact with the emulation UI.

Currently, the live-system contains two disk partitions. The first partition contains a read-only file system containing a ready-made installation of all necessary software components, i.e. emulators, the second partitions is writeable and contains by default two folders:

- `configs/` contains configuration files
- `image-archive/` an optional image-archive

While the first partition has a fixed size (currently about 1 GB), size and filesystem type of the second partition can be changed by the user, as long as the filesystem is supported by a current Linux kernel. For a demo setup[3], we choose the proprietary filesystem *exFAT*[4] in order to support virtual

[1] Multimedia Emulation, http://www.multimedia-emulation.de/

[2] https://help.ubuntu.com/community/LiveCD

[3] A sample USB image can be downloaded http://bw-fla.uni-freiburg.de/usb-demo.img We recommend to use a fast USB 3.0 stick, with at least 8 GB capacity.

[4] exFAT, https://en.wikipedia.org/wiki/ExFAT

disk images larger than 4 GB and to be compatible with most major desktop operating systems.

4.1 System Configuration

The configuration directory (`configs`) contains

- `common/` common configuration, e.g. configure the inactivity timeout;
- `remote/` configuration of remote image- and object-archive;
- `local/` configuration of a local image- and object-archive;
- `X11/` custom `xorg.conf` options, only required for old CRT monitors without EDID[5] support or to use non-standard input peripherals.

The local configuration is always preferred by the system. Only if no image-archive (respectively object-archive) folder is present on the data partition, the remote configuration is used.

For debugging purposes, a *devmode* switch is available that disables safeguards against accessing the underlying Linux system, allowing full terminal access and access to the systems log-files while running on the target machine.

4.2 Object and Image Preparation

In order to setup a custom emulation environment, the user needs to provide a disk image, supported by one of the emulators, a digital artefact to be rendered using the disk image's operating system and installed software and metadata describing the complete setup.

The most simple way to produce and test a desired emulation setup is to use the EaaS web-based environment. Workflows can be used to adapt existing disk images, for instance, installing additional software, testing artefact's rendering performance or configuring the environment to autostart the rendering process. The result can then be downloaded and copied to the USB-drive's second partition (image-archive). Alternatively, emulation environment metadata can be edited manually.

Alternatively, the USB live-system integrates well with an existing EaaS environment, by configuring a remote image-archive and/or object archive. In this setting, emulators still run on the local CPU and are able to make use of locally attached peripherals, while content (images and/or objects) is served through the network. Currently, the USB live-system requires a cable network with enabled DCHP service to function. WiFi connections are not yet supported.

The installation can either be configured to boot directly into a specific environment by putting a file (`environment-id.txt`) into the top-level directory of the second partition. The file should contain only the ID of the environment to load. You can find the ID of an environment in its metadata.

Furthermore, the live-system supports a reading-room setup with web-based user interface, which allows users to choose an environment. This setting is default, if no specific environment is set via `environment-id.txt`. In this setting, the user is able to switch between a full screen view and a web-based view (*CRTL-ALT-F*). In the non-fullscreen mode, the user may have options to cite an environment, create a screenshot, change a medium, etc.

5. STAGING THE 20 YEARS ANNIVERSARY EXHIBITION OF MBCBFTW

In 2016, Olia Lialina's pioneer 1996 net art piece *My Boyfriend Came Back From The War* had its twentieth anniversary. Haus der elektronischen Künste (HeK) in Basel, Switzerland, ran a retrospective exhibition of this work, combined with versions created by other artists [4], running from January 20 to March 20 2016.

For the exhibition, four EMiL-based live-systems were used, running on standard Intel NUC Mini PCs. USB 3.0 thumbdrives were prepared containing one artwork each, as well as disk-images of the the required environments and operating systems – in this case Windows 98 and Windows XP, running Netscape 3 and Internet Explorer 6. The operating systems and browsers were set up using EaaS web workflows and then exported to the USB drives to auto-boot. The environments were configured to automatically start the browser containing the desired artwork, but were otherwise not locked down or limited in use. The audience was able to freely interact with the complete environment, for example using the Windows "Start" menu to run Microsoft Paint, but any changes made to the environment were reset after five minutes of inactivity.

The network was set up to transparently connect to a locally running web archive server based on Rhizome's Webrecorder [5], so correct URLs would be displayed in the browsers even for long-defunct web sites. Since all web traffic was handled by the web archive, the gallery audience would not be able to leave the boundaries defined by curation. The web archive server was configured to only deliver dial-up speed connections.

Using standard adaptors, hardware contemporary with the works was connected to the modern Mini PCs: end-user grade 14" and 17" CRT screens, one 15" LCD screen, and ball mice delivered authentic historical input/output devices. Additionally, period computer cases were used as props, with cables placed as if the tower was connected to them (Fig. 1). The historic hardware was lent from the collection of the media restoration department at the Bern University of the Arts.

Since some of the CRT screens were unable to communicate their technical capabilities to the EMiL Linux kernel (either because they were built before I2C/Display Data channel was standardized, or they simply didn't support either interface), graphics modes and horizontal and vertical sync had to be forced via software settings. Since this poses a risk for damaging the monitors, the required modes had to be tried carefully. In general, when legacy CRT monitors are used, the risk of them failing is relatively high even when all settings are correct, just because of their age. It is advisable to have backup monitors in place for that case.

In other cases, if the data exchange between display and kernel works, and the requested graphics mode is much lower than the monitor's recommended default, a too-high resolution might be selected by the kernel, presenting the emulator's visuals centered on the screen instead of fullscreen. This is desirable to avoid image distortion when emulating a 4:3 display output to be shown on a 16:9 LCD screen for instance, but doesn't make sense on a low-end CRT. In this

[5] VESA Enhanced Extended Display Identification Data Standard, Video Electronics Standard Association (VESA), Feb. 9, 2000

case, again, the graphics mode has to be forced via software.

Legacy USB 1.0 peripherals, in this case the ball mice, which were connected to the Mini PCs via standard PS2-to-USB adapters, can cause a whole USB controller to switch back to the very slow USB 1.0 mode. As the EMiL emulation system boots from an external USB 3.0 thumbdrive, it is important to use computers with at least two separate USB controllers, so that the peripherals' bus is separated from disk access. In the case of the Intel NUC systems, the solution was to connect mice and keyboards on the front USB sockets and the thumbdrives on the back.

After these issues had been solved for the HeK exhibition, it was possible to send the emulation-based artworks on bootable USB thumbdrives via FedEx to other galleries and museums, who would again use standard PCs to exhibit them. These other institutions sourced legacy input/output devices (CRT screens and mice) from cheap Ebay offers or could used old equipment they still had in their possession.

From February 19 to March 30 2016, the exhibition was shown at MU in Eindhoven, The Netherlands. One additional work, a fast-paced action game, was added to be shown via EMiL running a MacOS 9 system with keyboard interaction. Thanks to the abstraction offered by bwFLA, this operating system was configured and exported to EMiL within the same web workflow as the Windows systems. The overall exhibition design was changed, but adhering to the same principles of legacy hardware and props. Glass table tops, exposing the Mini PCs running the emulators, were used to highlight the staging aspect.

Both the technical staff at HeK and MU have reported that the emulators have run with great stability throughout the whole exhibitions' times.

Parts of the exhibition were also shown at the exhibitions *Electronic Superhighway (2016 – 1966)* at the Whitechapel Gallery, London, UK [3] and *Mashup* at the Vancouver Art Gallery, Vancouver, Canada [1], using the exact same techniques.

6. CONCLUSION

The combination of emulated, fully accessible legacy environments, reduced network speed, web archives, legacy input/output output devices and props provided a rich, narrative techno-cultural context for the presented net artworks and defined a very practical definition of the artworks' boundaries.

EMiL has greatly normalized the work required for exhibiting complex net art in physical space.

Future work on the EMiL system will improve exporting and update mechanisms for emulators stored on local disks or thumbdrives, offer more local settings and simplify the setup process for graphic modes.

Digital art produced right now for current operating systems like Windows 10, Linux, or Mac OSX will be possible to be re-enacted in the future using the same techniques, since integration work for these and more legacy systems is ongoing.

7. REFERENCES

[1] D. Augaitis, B. Grenville, and S. Rebick, editors. *MashUp: The Birth of Modern Culture*. Vancouver Art Gallery, 2016.

[2] D. Espenschied, I. Valizada, O. Stobbe, T. Liebetraut, and K. Rechert. (re-)publication of preserved, interactive content âĂŞ theresa duncan cd-roms: Visionary videogames for girls. In *Proceedings of the 12th International Conference on Digital Preservation (iPres15)*, 2015.

[3] O. Kholeif, editor. *Electronic Superhighway.* Whitechapel Gallery, 2016.

[4] O. Lialina and S. Himmelsbach. *My Boyfriend Came Back From The War - Online Since 1996.* Christoph Merian Verlag, 2016.

[5] M. McKeehan. Symmetrical web archiving with webrecorder, a browser-based tool for digital social memory. http://ndsr.nycdigital.org/symmetrical-web-archiving-with-webrecorder-a-browser-based-tool-for-digital-social-memory-an-interview-with-ilya-kreymer/, 2016.

[6] D. S. Rosenthal. Emulation & virtualization as preservation strategies. https://mellon.org/resources/news/articles/emulation-virtualization-preservation-strategies/, 2015.

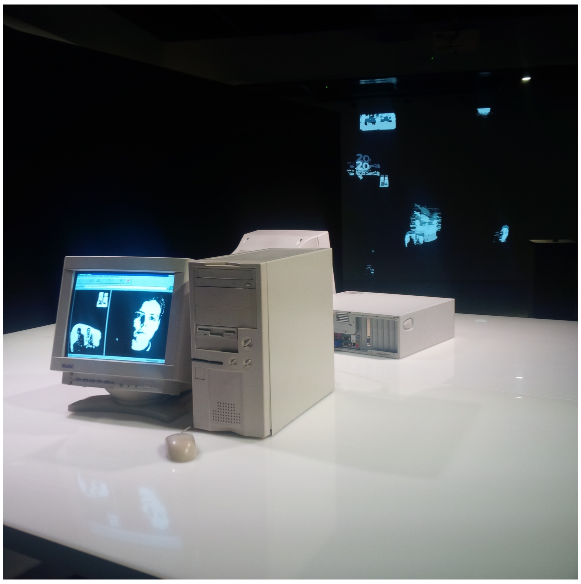

Figure 1: The MBCBFTW exhibition at HeK featuring mockup a "tower" computer case and a CRT screen connected to the Intel NUC (mounted concealed below the desk).