The KEEP Emulation Framework

Winfried Bergmeyer

Computerspielemuseum, Berlin, Germany
(bergmeyer@computerspielemuseum.de)

Abstract. As part of the overall KEEP Project the task of the Emulation Framework (EF) is to provide the emulation environments required for this purpose. In a very simple and comprehensible way the users can employ emulations for the representation and the performance of both digital objects in obsolete data formats and applications for antiquated computer systems. The virtual reconstruction of original playback environments can reproduce the original look-and-feel-qualities.
By means of this approach audiences unfamiliar with the very concept of emulations can employ them in a private context as well as in an institutional framework. Thus a digital object stored in an archive can be rendered in that digital environment best suited to it without the time-consuming procedure of having to equip the computer on which it is to be run with emulation-specific configurations.
The range of applications is huge: In addition to allowing games to be played with an original atmosphere it encompasses e. g. access to data in obsolete formats or data migration with original software. The decisive factor from the point of view of conservation is that the original stream is being employed. There are almost no limits as to the scope of emulators that can potentially be used inside EF. The first release will support the following platforms: x86, Commodore 64, Amiga, BBC Micro and Amstrad/Schneider CPC.

Keywords: KEEP, Emulation, Emulation Framework, Virtual Machine, Transfer Tool

1 Emulation as a concept of conservation

The advantages of emulation as an alternative form of preservation (if compared to data migration) are numerous: Any strategy of preservation has to guarantee the permanence of the upkeep of the digital object as well as that of its accessibility[1].

[1] In 1999 Jeff Rothenberg declared: "*The best way to satisfy the criteria for a solution is to run the original software under emulation on future computers*". Jeff Rothenberg (1999). Avoiding Technological Quicksand: Finding a Viable Technical Foundation for Digital Preservation: A Report to the Council on Library and Information Resources – Washington,

Migration, on the other hand, ensures a long-term usability only at the expense of changing the data formats: a text document in wordperfect format .wpd e. g. will be conveyed into one that promises enduring availability such as .rtf or .txt. But this entails a loss of original information. Although you could say that the original file is being preserved, it cannot be rendered any longer for want of the original environment. It is therefore in the strict sense no longer accessible.

A number of repeated transfers can entail in the course of time significant modifications, which are bound to impair the content as well. Highly complex and proprietary formats such as objects in CAD may already suffer losses with the first migration. In the worst case they may not be saved in other formats. Compiled programs can only be migrated by means of a recompilation. For this purpose the uncompiled code will be required which is often not available any more. Thus this procedure cannot be performed when dealing with computer games or with commercial software.

But the approach of preserving the hardware on a permanent basis is not feasible. We can observe both the hardware components of the computer as well as the peripheral devices for input-, output- or reading functions become rare or defunct. Enormous costs would ensue if the spare parts had to be produced again.

Employing emulation, on the other hand, allows us to appropriate the original bitstream without taking recourse to migration. The digital objects are rendered on the virtually provided original platform. This reproduction of an original look-and-feel can be important for quite a few objects, e.g. for computer games, digital art objects or in the case of poetry presented with a sophisticated typography. In guaranteeing permanent access to these objects emulation is a substantial advance in terms of quality assurance. Emulation may yield yet another advantage. The original environment as in the case of an obsolete data bank management system can be employed for the initial transfer of the data in other formats. Thus the correct processing of the source data is ensured.

2 Transfer process

Maintaining the original bitstream is part and parcel of any responsible form of long-term preservation. But compared with conventional media such as books, painting or prints digital data prove to be incredibly frail. Even an apparently small amount of informational loss (like the scratched surface of a CD) can invalidate all the data. Damage to a painting or the pages torn in a book do not prohibit the usage of the object or at least those parts left intact. With digital data this is not possible.

In the course of recent decades various technologies for storing data have been employed. This means that a lot of different reading devices are needed for the transfer. In addition to 3.5“ and 5.25“ disk drives (for which there are still reading devices) there were less popular formats such as 8“- or even 3“-disks. Furthermore we

DC: Council on Library and Information Resources, p VI. http://www.clir.org/pubs/reports /rothenberg /pub77.pdf.

have still some readers for tape-based storage system (like DLP- or LTO-tapes as well as audio cassettes). But the situation is much more complicated for reel-to-reel tapes or microcassettes. In the case of early computer game consoles cartridge systems were the most popular form and they represent a serious challenge for transfer processes.

Thus the very process of data transfer into a new storage system lies at the heart of any concept dealing with conservation. We have to distinguish between the saving of single files (texts, pictures, tables) and that of preserving the entire data carrier (like the cartridges of obsolete gaming consoles). For certain areas of application it is essential to transfer the medium in its entirety into a virtual image, which then in turn can be embedded as a data carrier (a disk, a CD-ROM or a hard disk) in the emulator. These images have one huge advantage. They can be integrated into a systematic long-term archive that can be stored in a system of hard disks.

3 Copy protection

Copy protection systems for computer games prove a double problem for the long-term preservation. On the one hand you cannot always successfully bypass the copy protection. Thus for example a systematic copy protection involving a dongle (i. e. a protection that requires certain hardware elements for the program to work) cannot be transferred into useable images without changing the original program code. (This is the problem with many CD-ROMs and DVDs.) Similarly substantial knowledge about the mechanisms of copy protection is required to produce working images of older protected floppy disks. And it is regrettably extremely difficult to identify a copy protection before starting the transfer. In the case of CD-ROMs and DVDs a variety of tools able to identify many protection systems is available. But in the case of floppy disks no such instruments have been created for data carrier of systems like C64 or Amiga.

And furthermore this bypassing is still an offence even if performed by an institution working under the premiss of conserving these objects of cultural value[2]. (There is different legislation in other countries.) In the case of the Computerspielemuseum we are technically in the position to circumvent many of those copy protection systems. However, this would represent a violation of existing German laws. We thus need legal changes in the future to ensure the preservation of the digital objects[3].

---

[2] *„Die Gedächtnisinstitutionen können ihren gesetzlichen Aufgaben im digitalen Umfeld nur eingeschränkt nachkommen. Eine effektive und umfassende digitale Langzeitarchivierung ist ihnen rechtlich nicht möglich. Es droht eine digitale Amnesie des Kulturellen Gedächtnisses.“* Digitale Langzeitarchivierung als Thema für den 3. Korb zum Urheberrechtsgesetz. Urheberrechtliche Probleme der digitalen Langzeitarchivierung (2011). Paper of the German competence network for digital preservation (nestor). http://files.d-nb.de/nestor/berichte/nestor-Stellungnahme_AG-Recht.pdf.

[3] Eberhard Hilf, Christian Keitel, Kai Naumann, Martin Iordanidis, Christina Bankhardt, Sven Vlaeminck, Reinhard Altenhöner, Sabine Schrimpf, Natascha Schumann (2010): Sozio-ökonomische Erfolgsfaktoren für die Langzeitarchivierung in Deutschland. nestor-

4 Emulators

The history of digital emulators commenced some 30 years ago. And interestingly enough it all started with a gaming console, ColecoVision produced by Coleco. There was an adapter available on the market allowing to play games initially developed for and to be run on the Atari 2600. And in 1985 Atari, in order to prove the capacity of the 68000-CPU and the TOS-systems software, in turn developed a Z80-emulator, which allowed to run software originally designed for the CP/M-operating system. This meant that a vast range of software was available for that computer system.

Presently a huge number of emulators exist for different consoles and platforms. In the majority they have been created inside the community of the retrogamers and for their purposes. But with more and more devices in circulation that are both portable and equipped with sophisticated graphics such as the IPhone or the IPad we find a growing number of emulators included for commercial products.

One current argument against the appropriation of emulators for the purpose of long-term preservation points to the fact that emulators are still tied to specific platforms. A reliable and permanent availability is not guaranteed as long as the introduction of new operating systems or new hardware technologies will necessitate a new porting of the emulators. In addition to recent attempts to develop Java-based emulators[4], we have the Dioscuri-Project at the Koninklijke Bibliotheek (Den Haag) which presents the first emulator designed to operate independently from specific platforms[5].

These are some of the obstacles on the path for using emulation as a fully fledged instrument for the purpose of preservation. But since keeping the digital tradition alive cannot be achieved without emulation, it is essential to develop concepts and technologies which will facilitate it and make it available on a permanent basis.

5 KEEP (Keeping Emulation Environments Portable)

Keeping Emulation Environments Portable (KEEP) is a medium scale research project started on 1 February 2009 co-financed by the 7th Framework Programme's ICT-3-4.3 Digital libraries and technology-enhanced learning priority[6]. The overall aim of the project is to facilitate universal access to our cultural heritage by developing flexible tools for accessing and storing a wide range of digital objects. Although primarily aimed at those involved in Cultural Heritage, such as memory institutions and games museums, the KEEP Emulation Services can also serve the needs of a wide range of organisations and individuals because of its universal approach.

Positionspapier zum Abschlussbericht der Blue Ribbon Task Force on Sustainable Digital Preservation, http://files.d-nb.de/nestor/berichte/nestor_Stellungnahme_BRTF.pdf.

[4] For example the Amstrad Emulator "JavaCPC" on Java basis. http://sourceforge.net/ projects/javacpc /files/

[5] http://sourceforge.net/projects/dioscuri/

[6] http://www.keep-project.eu

The following institutions are partners in this cooperation:

- Bibliothèque nationale de France (Paris) project coordinator
- Koninklijke Bibliotheek (Den Haag)
- Deutsche Nationalbibliothek (Frankfurt)
- University of Portsmouth
- Tessella
- Joguin SAS
- European Games Developer Federation
- Computerspielemuseum (Berlin)

5.1 The KEEP Transfer Tool Framework (TTF)

The development of a transfer tool framework (TTF) is to simplify and to automatize the workflow. Our main objective is to design a concept for the appropriate elements for the workflow, to compile the collections of meta-data required for this purpose and to mould these results into a project allowing long-term preservation. Existing systems for data transferal have been evaluated with regard to their suitability for the TTF. Simultaneously we have analyzed existing copy protection systems.

Part of the workflow is to monitor the transfer process. A failed attempt of transfer or a dysfunctional image may be due to one of the following reasons:

- The reading process was not performed without mistakes.
- A copy protection previously unperceived has undermined the process.

A further obstacle for the preservation is the multitude of data formats for images or the number of formats supported by those emulators currently available. In the field of computer games a lot of special formats have been established as the standards for the individual systems. These in turn have different potential to bypass copy protection.

These are a number of problems you can encounter while working on a transfer. The process of automatically transferring larger quantities of data media from a diversity of operating systems (which by necessity can be the only form of procedure that larger institutions can afford to work with) is still something of a challenge and something to strive for.

Thus we are preparing to embed the TTF in the Planets Interoperability Framework[7], which allows integration of the transfer workflow into an already existing framework for digital preservation. A KEEP study on the integration of the KEEP Transfer Tool with the Planets Interoperability framework states, that it is feasible from a technical and legal perspective. For this purpose a new API for the integration of the services and tools for the TTF has to be developed. After that the infrastructure within the Planets IF can be used for the transfer workflow.

[7]http://www.planets-project.eu/docs/reports/Planets_IF-D11_ConsolidatedReleaseDocumentation.pdf

5.2 The Keep Virtual Machine (KVM)

As one can see, the very name of the project, KEEP (i.e. Keeping Emulation Environments Portable) emphasizes the importance of the permanence accorded to the emulation environments. We are currently working on developing a virtualizer that can serve as a platform for many emulators already available. Portability and flexibility are the core requirements for Keep Virtual Machine (KVM). The immediate advantage is that it prevents the concept of stacked emulation, which is where multiple emulators each translate a platform for a specific era for another emulator. Stacking emulated computers creates a dangerous stack of dependencies that is vulnerable for errors and mistakes.

The KVM is based upon the concept of a virtual layer, based on the idea of Jeff Rothenberg[8]. The KVM implementation is defined by the selection of a base sub-machine, directly implemented by way of an emulator or a dynamic compiler written for a support machine, as well as a work sub-machine, whose complexity is higher or equal to that of the base machine, and for which applications are written. The virtual processors are a set of sub-machines that vary in complexity and efficiency, structured from the simplest to the most complex, featuring in that order: KVM0, KVM1, KVM2, KVM3 and KVM4. The most basic commands allow easy adjustment to fit in new platforms by means of simple and elementary operations, whereas you will find complex command structures at the top level that can communicate with systems such as Linux, Java or with the emulators themselves. These have to be customized initially to be able to cooperate with the KVM. Afterwards it will only be the other way round: the KVM has to be matched with new platforms. This will yield an extremely efficient virtualizer, whose emulators once matched can be employed for a long time. This will mean a major progress for a safe and sound appropriation of emulation technologies for the purpose of conservation.

5.3 The KEEP Emulation Framework (EF)

The theme central to this workshop is a presentation of the KEEP Emulation Framework. Employing technologies of emulation in institutions so far means a high amount of work input before anything is available for the end-user, e. g. somebody reading a book in a library or visiting an exhibition in a museum. To provide a variety of user-friendly emulators for different digital objects entails the following: Firstly the relevant meta-data have to be stored in the digital archive and secondly the different emulators have to be implemented and maintained. A process of automatization for the recognition and the verification of the data and the allocation of the appropriate

[8] Rothenberg, J., Preservation of the Times, The Information Management Journal, March/April 2002, Vol 36, No. 2, pp. 38-43. ISSN 1535-2897, available at: http://www.panix.com/~jeffr/Prof/Pubs/DigitalLongevity/arma.paper.from-journal.pdf

emulation environment will inevitably facilitate the process of issuing them to the end-user and eliminate a number of intermediate administrative procedures.

The KEEP EF has the resources to perform exactly this very process of automatization. Via a defined interface a request for information will be addressed to the digital archive of an institution and will activate the EF. The data-file (a single file or the image of a data-carrier) are identified by means of a format-specific registry and then the appropriate emulation pathways show up on the screen. Selecting a specific pathway will start the emulator and the software package necessary for performing this task.

The EF software consists of three parts: a Core Application, Software Archive and Emulator Archive. The Core EF is the technical heart of the system, performing the automatic characterization of file formats, selecting the required software and automatically configuring the emulation environment. It has a simple GUI to interact directly with the user. For selecting the software and emulator, the Core interacts with external services such as technical registries containing file format classifications, the Software Archive that contains software captured in predefined disk images and the Emulator Archive that contains the emulators available for the EF.

The Core EF, Software Archive and Emulator Archive are developed by Tessella with support from the National Library of the Netherlands (Koninklijke Bibliotheek, KB). The Core GUI is developed by the National Library of the Netherlands.

The software.

For the last few weeks KEEP EF has been made available on the Source Forge Portal for free download9. Java was chosen as the development language because of familiarity, widely supported libraries and overall portability; for the internal database H2 was chosen because of the small footprint and integrated web-interface. An installation program simplifies the installation process. These are the system requirements:

Processor	X86 32/64 bit 1.5 GHz or faster
Memory	At least 2 GB of memory
Disk space	200 Megabytes if free available space for the base install Depending on the number of emulators and software images, from 1 GB upwards
Operating system	Linux or Windows with compatible JRE and network support
Java Runtime Environment (JRE)	Oracle(Sun JRE version 1.6 or higher / compatible

The download package contains the following emulators:

- Qemu (x86)
- Dioscuri (x86)

[9] http://sourceforge.net/projects/emuframework/

- WinUAE /Amiga)
- VICE (C64)
- JavaCPC (Amstrad/Schneider)
- BeebEm (BBCmicro)

And the following format registries are included:

- Pronom/PCR
- UDFR

We have to emphasize that this package may only contain open source software. Thus you will neither find an operating system such as MS Windows operating system nor an application like wordperfect to come along with the bundle. Via the administrator's interface, however, the system can be expanded and modified to the individual requirements.

Functionalities for the end-user.

Before we will proceed to explain both the technical and the administrative features, we will briefly describe the capabilities of the system as well as the functionalities relevant for the end-user.

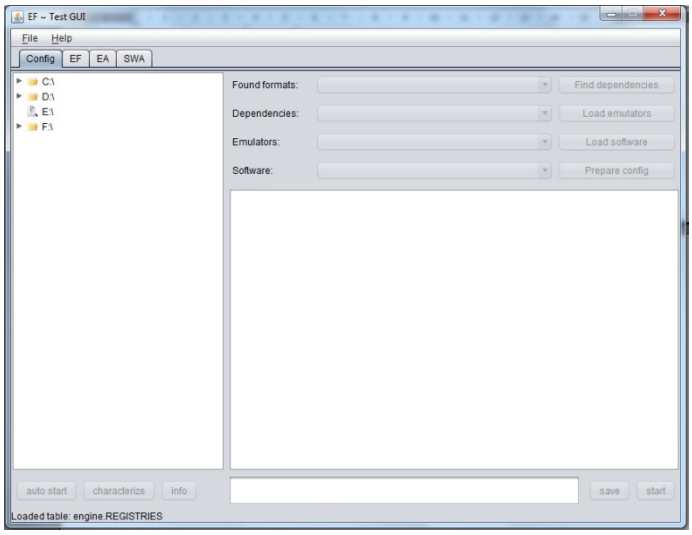

Fig.1. Java-based EF GUI

This is still an experimental graphical user interface to give a first impression of the functionality. At the end of the project there will be a more effective interface, especially for the administration. With the actual java-based Gui the user can select a digital object in the left column. To describe the next steps in the procedure, we will employ a d64-image of a game designed for the Commodore64 (*Ms Pacman*). Having selected the object by clicking on the icon, the lower region of the screen presents

three different buttons (*auto start*, *characterize* and *info*). If you click on the button for auto start, a window will pop up presenting the VICE-emulator and the game.

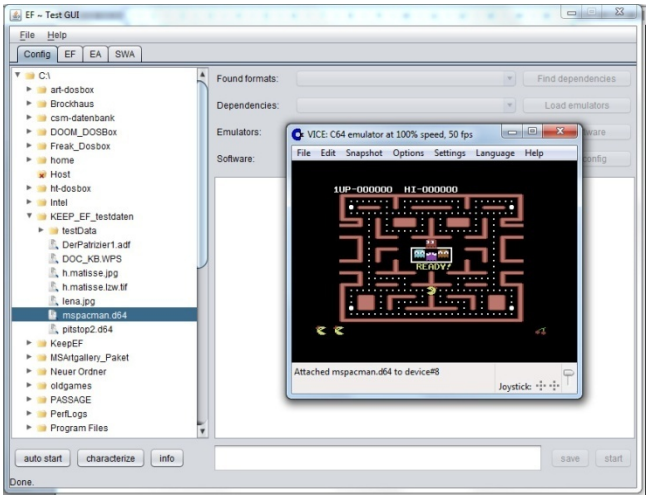

Fig.2. Commodore64 Emulator VICE with *Ms Packman* running

This intermediate step can be omitted. You can directly start the emulation from the digital archive. But in addition to this automatic procedure the user can select between several emulation pathways (if they are defined by the administrator).
To illustrate this approach we will select a JPEG-file: If you activate the button „characterize“, information in the right upper corner states that JHove has successfully identified the data format to be JPEG.

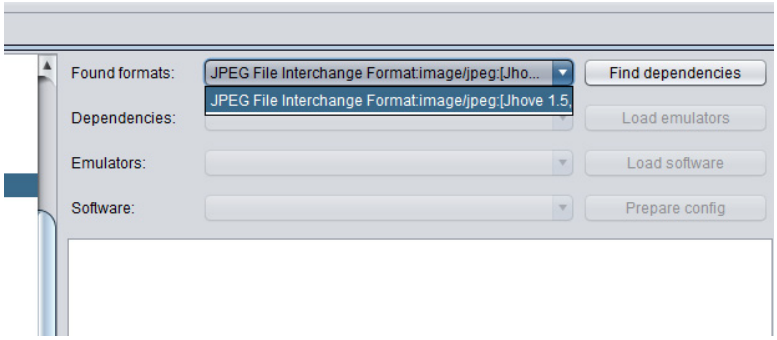

Fig.3. Format identification

In the next step the current dependencies will be shown, i. e. which emulation paths have been allocated to the data format by the administrator. In this case the following two are available:

- the program Blocek under FreeDos on a x86 system

- the program XzgV under Damned small Linux on a x86 system

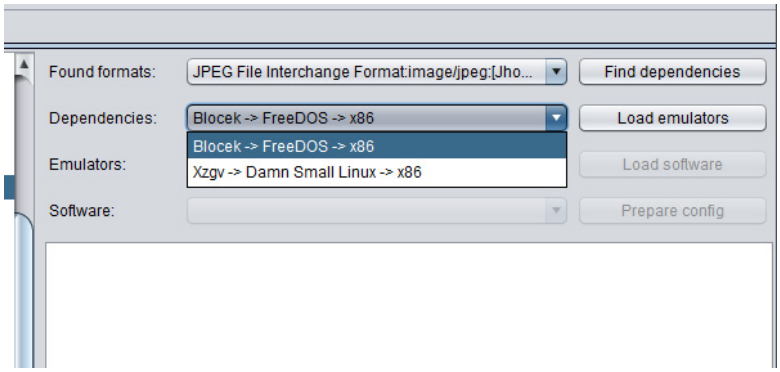

Fig.4. Dependencies

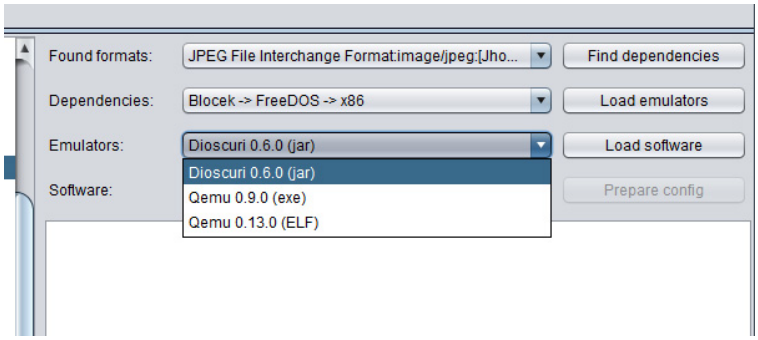

Fig.5. Available emulators

Now the user can select one of the paths and henceforth the emulators available, in this case the choice is between Dioscuri and Qemu.

The next step consists in opting for the appropriate software package. We see FreeDos being offered in connection with the Blocek-application.

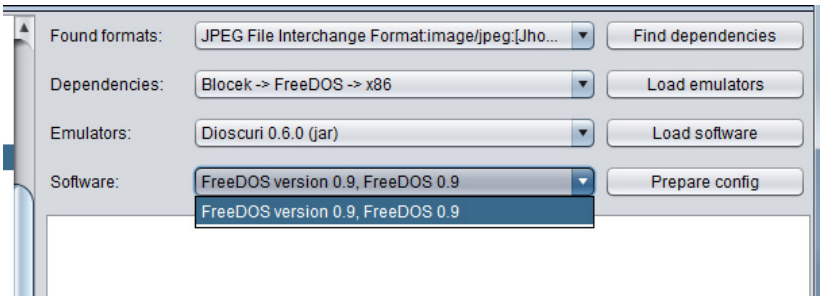

Fig.6. Available software

Finally the configuration is being prepared. This entails mounting file as a disk (drive A). The system mounts every object with less than 1.2 MB as a disk, any file larger than that will be tied in automatically as a hard disk.

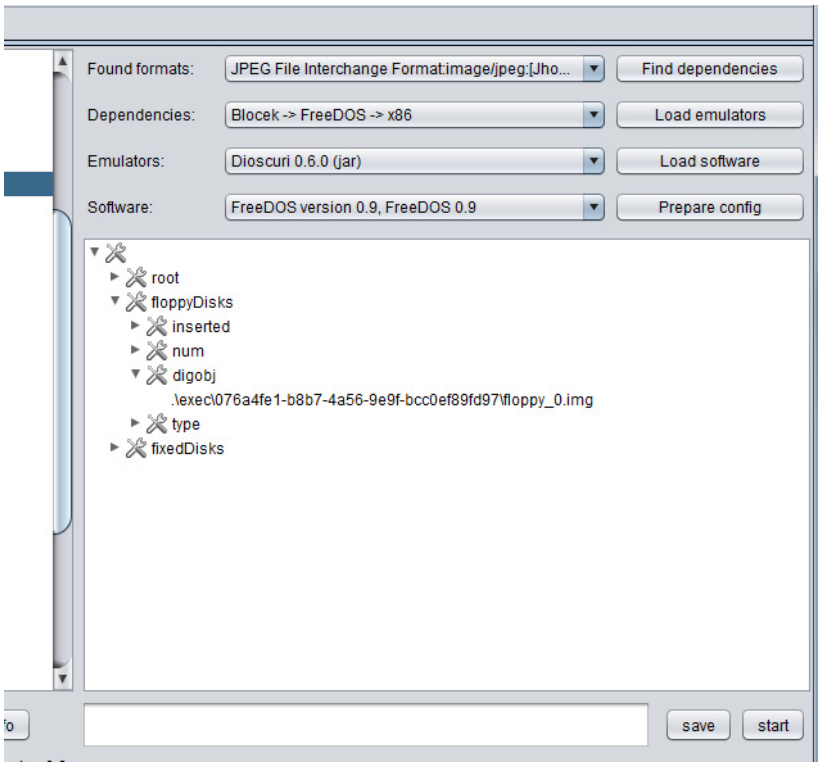

Fig.7. Available emulators

Once you are on the level of the emulation environment, this integration is necessary in order to have access to the object. By clicking on the button *start* the process of emulation will commence. When the program *Blocek* has been started on the Dos-Interface, the JPEG-file on the disk in the drive may be opened.

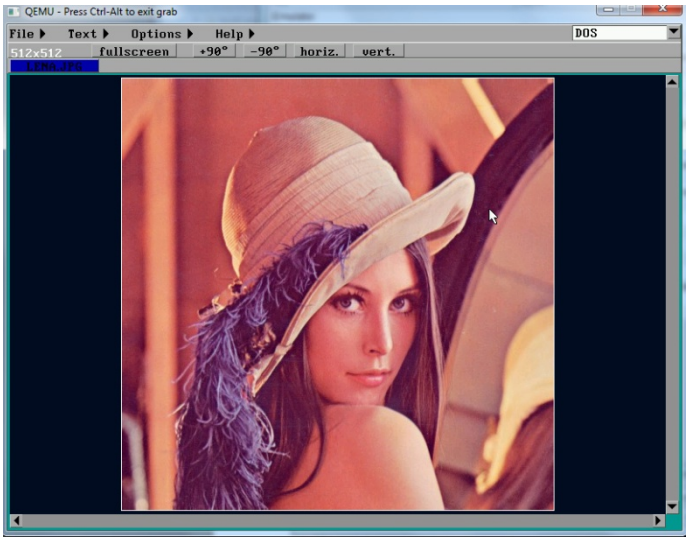

Fig.8. JPEG-image rendered with *Blocek*

By means of this selection the user can navigate and start and test different emulation environments. So let us now take a look at the EF from its technological and adminstrative side.

5.4 Administration area

The concept of the EF is based upon employing three different archives which may operate on different servers. The EF-Core-Archive needs to be installed on the very computer requested to present the emulation. But the archives for both the emulators themselves and the appropriate software have only to be connected with it by means of LAN or WLAN. This allows for a central administration of the emulators and the software for a number of computers or even for a network in an institution. We anticipate two groups of users: administrators and end-users.

In addition to the two format registries contained in the download package it will be possible to create your own registries and to integrate them. Thus individual solutions to fit the requirements of a particular institution can be created.

By means of this GUI the current preferences may be inspected and modified. But due to its complexity we will present only some aspects of the configuration.

Emulator Archive.

The administrator is enabled to integrate new emulators, if the ones available do not fit the standards or cannot emulate the hardware necessary for the presentation. The emulator itself will be integrated as a blob. There may be the need to both define new image formats and to register the linking with the new emulator.

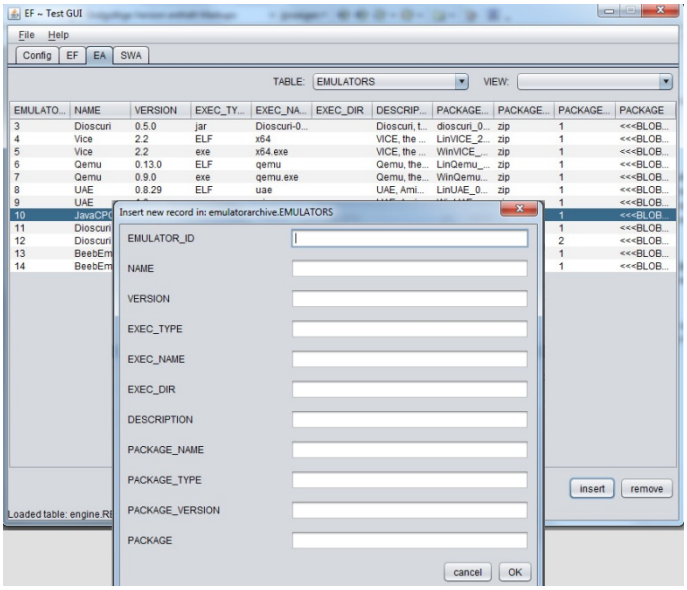

Fig.9. EF GUI with form for adding new emulators

Software Archive.

If there is an additional demand for an operating system and/or an application (word processing, CAD-programs, database systems) for the emulation environment to work beyond the mere emulation of the hardware, additional software packages will be required. These, too, will be stored as blobs in the database. So additional entries for applications, operating systems, platforms and file formats have to be made.

If all entries are correct, the verification of the new emulation pathway can commence via the viewing of the pathways.

FILEFORMAT_ID	FILEFORMAT_...	APP_ID	APP_NAME	OS_ID	OS_NAME	PLATFORM_ID	PLATFORM_N...
FFT-1007	Extensible Mark...	APP-1000	FreeDOS Edit	OPS-1000	FreeDOS	HPF-1004	x86
FFT-1008	Plain text	APP-1000	FreeDOS Edit	OPS-1000	FreeDOS	HPF-1004	x86
FFT-1009	JPEG File Inter...	APP-1001	Blocek	OPS-1000	FreeDOS	HPF-1004	x86
FFT-1010	Windows Bitmap	APP-1001	Blocek	OPS-1000	FreeDOS	HPF-1004	x86
FFT-1011	Graphics Interc...	APP-1001	Blocek	OPS-1000	FreeDOS	HPF-1004	x86
FFT-1012	Tagged Image ...	APP-1001	Blocek	OPS-1000	FreeDOS	HPF-1004	x86
FFT-1013	Portable Netwo...	APP-1001	Blocek	OPS-1000	FreeDOS	HPF-1004	x86
FFT-1009	JPEG File Inter...	APP-1002	Xzgv	OPS-1001	Damn Small Li...	HPF-1004	x86
FFT-1010	Windows Bitmap	APP-1002	Xzgv	OPS-1001	Damn Small Li...	HPF-1004	x86
FFT-1011	Graphics Interc...	APP-1002	Xzgv	OPS-1001	Damn Small Li...	HPF-1004	x86
FFT-1012	Tagged Image ...	APP-1002	Xzgv	OPS-1001	Damn Small Li...	HPF-1004	x86
FFT-1013	Portable Netwo...	APP-1002	Xzgv	OPS-1001	Damn Small Li...	HPF-1004	x86
FFT-1006	Portable Docu...	APP-1003	Xpdf	OPS-1001	Damn Small Li...	HPF-1004	x86
FFT-1007	Extensible Mark...	APP-1004	Beaver	OPS-1001	Damn Small Li...	HPF-1004	x86
FFT-1008	Plain text	APP-1004	Beaver	OPS-1001	Damn Small Li...	HPF-1004	x86
FFT-1014	Hypertext Marku...	APP-1005	Firefox	OPS-1001	Damn Small Li...	HPF-1004	x86
FFT-1016	Microsoft Word	APP-1006	MS Office Viewer	OPS-1001	Damn Small Li...	HPF-1004	x86

Fig.10. List of available emulation pathways

6 Resumee

The EF will simplify the appropriation of emulators for the usage of original digital objects substantially. And the considerable range of possibilities the administrators are provided with means that customized solutions for emulators (and hardware platforms) and software and for individual format registries can be integrated into a framework allowing for a high level of automatization.

In connection with the two big tasks – the transfer (TTF) and the hardware independency of the emulators (KVM) – which both are busily worked on, the KEEP-Project has come up with a solution how to simplify emulation and employ it as a feasible procedure for long-term conservation that in itself is also built to last.

7 References

1. Jeffrey van der Hoeven, Dirk von Suchodoletz: Emulation: From Digital Artefact to Remotely Rendered Environments. The International Journal of Digital Curation - Issue 3, Volume 4, (2009)
2. Karsten Huth: Probleme und Lösungsansätze zur Archivierung von Computer-programmen – am Beispiel der Software des Atari VCS 2600 und des C64. Berlin, (2004), http://www.digitalgamearchive.org/data/news /Softw_Preserv_ huth.pdf.
3. Bram Lohman, Jeffrey van der Hoeven: Building the emulator - A behind-the-scene look at development. Koninklijke Bibliotheek (KB), (2006), http://www.kb.nl/hrd/dd/dd_projecten/slides/eem_kbnatessella_jvdhoeven _blohman.pdf .
4. Lorie, Raymond: the UVC: a method for preserving digital documents – proof of concept, (2002), http://www-5.ibm.com/nl/dias/resource/uvc.pdf
5. Jeff Rothenberg: Avoiding Technological Quicksand: Finding a Viable Technical Foundation for Digital Preservation: A Report to the Council on Library and Information Resources – Washington, DC: Council on Library and Information Resources, (1999), http://www.clir.org/pubs/reports /rothenberg /pub77.pdf.
6. Jeff Rothenberg. Using Emulation to Preserve Digital Documents. Koninklijke Bibliotheek, the National Library of the Netherlands The Hague, (2000)
7. Dirk von Suchodoletz. Funktionale Langzeitarchivierung digitaler Objekte - Erfolgsbedingungen des Einsatzes von Emulationsstrategien. Freiburg im Breisgau, (2008), http://nbn-resolving.de/urn:nbn:de:0008-2008070219.
8. Randolph Welte. Funktionale Langzeitarchivierung digitaler Objekte – Entwicklung eines Demonstrators zur Internetnutzung emulierter Ablaufumgebungen. Freiburg im Breisgau, (2008)
9. Randolph Welte Klaus Rechert, Dirk von Suchodoletz. Emulation Based Services in Digital Preservation. Proceedings of the 10th annual joint conference on Digital libraries, (2010)

10. Hilde van Wijngaarden, Jeffrey van der Hoeven. Modular emulation as a long-term preservation strategy for digital objects. Koninklijke Bibliotheek, the National Library of the Netherlands The Hague, The Netherlands, (2005)