Remote Emulation for Migration Services in a Distributed Preservation Framework

Dirk von Suchodoletz
Department of Computer Science
University of Freiburg
Freiburg i.B., Germany
dirk.von.suchodoletz@rz.uni-freiburg.de

Klaus Rechert
Department of Computer Science
University of Freiburg
Freiburg i. B., Germany
klaus.rechert@rz.uni-freiburg.de

Isgandar Valizada
Department of Computer Science
University of Freiburg
Freiburg i. B., Germany
isgandar.valizada@rz.uni-freiburg.de

ABSTRACT
Previous studies have shown the feasibility of migration services when using emulation technology. To make rather complex setups of original digital ecosystems usable in standard mass migration workflows, a separation of the systems running and their interfaces is required. Here remote emulation can become a crucial building block of future distributed preservation workflows and access systems.

In this paper we develop the requirements and a component model for *migration-by-emulation* services in a distributed environment based on division of labour. We suggest a modular system which offers interfaces to be accessed by standard preservation frameworks. It provides Web services allow access to original system environments via emulation engines with additional methods for automated interaction. The proposed migration units support versatile migration services and offer a wide range of file conversions based on a digital artifacts' original applications. The component-based architecture allows the distribution of system components among specialized memory institutions.

Categories and Subject Descriptors
H.3 [**Information Storage and Retrieval**]: Digital Libraries

Keywords
Digital Preservation, Emulation, Migration, Workflow, Automation, Original Environments, System Design

1. INTRODUCTION
Access to digital objects (DO) is at risk if their format is no longer supported by actual digital environments. Preserving digital artifacts for future access requires long-term support for applications and/or operating systems capable of accessing or running them. The emulation of original hardware/software environments provides the opportunity of using objects in their creation environment.

In most cases the applications or operating systems developed by the format vendors or software producers are the best candidates for handling a specific object of a certain type. The vendors are expected to have the most complete knowledge about their own data formats and the information available publicly is often incomplete or non-existent, especially regarding proprietary formats. Thus, in many cases there are no alternatives to access those objects within their original environments.

However, the ability to access obsolete DOs only by their corresponding original applications limits future access, especially if a certain environment is not available any more. This hindrance to future usage could be overcome by either migrating the object into an actual format accessible with todays tools or using its original environment. Emulation is the best way to reproduce original environments, which themselves provide the base layer for very flexible multiple migration input-to-output format scenarios. Furthermore, such emulation-based migration paths can be verified and evaluated in terms of quality and costs like traditional command-line conversion tools.

Performing migrations manually for every digital object is not a feasible strategy in many cases. Due to the large quantity of DOs held by many institutions, it would become a time-consuming and costly task. Additionally, depending on the original environments, many archivists or private users aren not knowledgeable about how to install a certain application or operating system or how to handle a certain emulator. Thus, a major prerequisite for making use of the emulation of original digital ecosystems in digital preservation (DP) workflows is the separation of the system running from the user in- and output. This allows the offering of emulation services over the network and the automation of user interaction. Based on previous studies, a system framework is required, such that complex migration tasks can be carried out in a scalable and controllable way. It should be possible to plug these services into existing preservation frameworks such as provided by PLANETS [8] or Rosetta.[1]

[1] A preservation framework developed and marketed by ex Libris.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
iPRES2011, Nov. 1–4, 2011, Singapore.
Copyright 2011 National Library Board Singapore & Nanyang Technological University

Following the aforementioned discussion, the suggested component system should meet a number of requirements:

- *Large Scale and Large Quantities*
  In order to provide large-scale migration facilities based on emulation, atomic migration components are required. Each atomic migration unit could be be run in parallel, and therefore being applicable for Cloud scenarios or the like.

- *Large Variety of Migration Paths*
  By utilizing a distributed framework, more complex migration paths become possible. Institutions can specialize in special subsets of digital objects or system environments. These can be made accessible as atomic migration units, which then can be connected to complex migration paths and DP workflows in general.

- *Accessibility for a Wide Range of User Groups*
  Through decentralized and specialized competence centers in emulation, practical knowledge is made available on the ancient system environment preparation, running, and maintenance, and applications can be shared within the DP community. Migration and verification workflows based on emulation thereby become more (cost-)efficient, especially for small tasks.

2. RELATED WORK

Remote emulation has been a topic in digital preservation for a while. Several approaches have been explored and some stable prototypes have been shown to be quite feasible. There are several ways to achieve the separation between a server able to run the complex tasks and a rather simple client mainly limited to user in- and output. This concept of separation is used more and more often to run complex computer games over the network without requiring the installation and permanent updating of client software on the user's side. Several ways of implementing a remote emulation service have been researched in the last couple of years. At first this research focused on direct user interaction with the original environments [18, 9, 7]. More recently, other uses like automation of workflows have become relevant [23].

The variants of a remote emulation service or emulation streaming service can be distinguished in several ways. There are various approaches to the transport protocol used to send the screen output to the user's device and receive the user's input. Some of the protocols are public and many open source implementations exist [24]. Others may have the added functionalities of audio stream transportation, removable block device data, or implement remote USB.

At Victoria University in Wellington, New Zealand, a prototype was implemented to demonstrate the feasibility of running a game on the visitor's smartphone without the need to install any software besides a simple streaming client [4]. A very popular method for remote access is the VNC protocol [13] implemented in a wide range of operating systems and appliances. Some emulators like QEMU or virtual machines like Virtual Box and VMware implement direct access to the virtual screen and input devices. Additionally, VNC was added to Dioscuri [6] as an outcome of a student's thesis. Since VNC implements mostly screen rendering plus mouse and keyboard input, other remote protocols like RDP or Citrix[2] seem to be more attractive as they are capable of transporting audio streams or even block devices over the net. But they are proprietary and the number of server and client implementations is comparatively limited. Since they are dependent on the direct implementation of remote access in the emulator, a large number of them can not be used. Following the research into GRATE [24] during PLANETS [5] a VNC enabled prototype of a remote emulation was developed for the OPF[3]: It realizes two types of services – a migration-by-emulation service and a create-view service (e.g. [11]).

3. DISTRIBUTED MIGRATION BY EMULATION

Migration-by-emulation describes the concept of using the original or a compatible environment of a designated digital object running in a virtual machine and thus replacing the original hardware and/or software stack. This approach avoids the often impossible alteration and adaptation of outdated software to present-day environments. An abstract and generalized migration-by-emulation workflow is depicted in Figure 1. A virtual machine runs within the host environment, which contains the selected original system environment suitable for handling a certain type of digital objects. The original system environment is either reproduced from original software stored in the software archive or cloned from a prototypical original system (cf. [19, p. 165]). The selection of an appropriate system environment for each object type can be described as a so-called *view path*, a pathway pointing from the digital artifact into its original rendering or execution environment [17].

To make migration-by-emulation deployable in large-scale preservation scenarios without relying on user interaction, the user's function is replaced by a workflow execution engine [12]. This requires appropriate interfaces in order to use emulators [20]. In contrast to simple command-line input-output migration tools, a migration-by-emulation service needs a more complex initial setup:

- *System Emulation*
  Hardware emulation including a full reconstruction of outdated components. For instance, a i386 CPU, ISA Systembus, PS/2 mouse and AT keyboard and VESA compatible graphics are minimal requirements, e.g. for Windows 3.11.

- *System Environment*
  An appropriate runtime environment (e.g. a disk image file) preconfigured with operating system, necessary drivers and tools, and the required target application. Furthermore, each environment specifies at least one transportation option, defining how digital objects can be injected into and extracted from the virtual environment. Examples range from different kinds of floppy-disk images to hard disk container formats and advanced networking options.

[2]cf. [24] for an overview and evaluation of the various protocols and their usability in remote emulation.
[3]Open Planets Foundation, non-for profit PLANETS follow-up, *http://www.openplanetsfoundation.org*

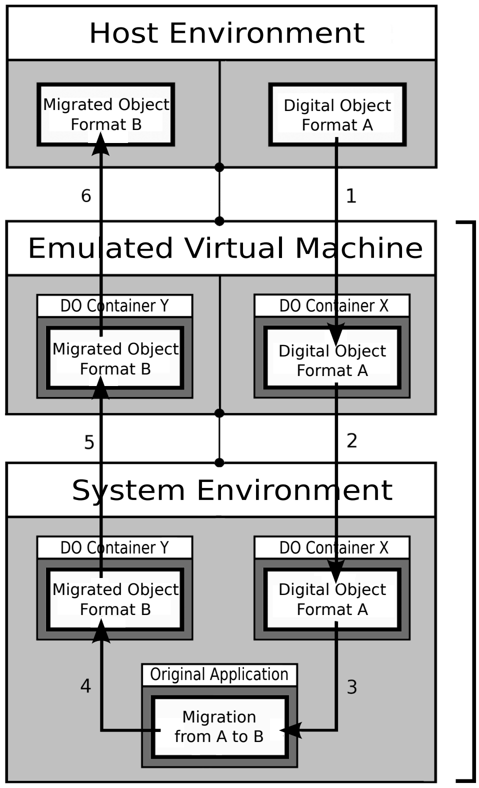

Figure 1: Migration-by-emulation workflow and involved system components.

- *Interactive Workflow Description*
  An abstract description of all interactive commands to be carried out in order to perform a certain migration. Such a description consists of an ordered list of interactive input actions (e.g. key strokes, mouse movements) and expected observable output from the environment (e.g. screen- or system-state) for synchronization purposes.

3.1 Base Layer – System Environments

A core requirement for the described workflows is the availability of the original system environment. This is the context in which the digital objects are rendered or run. It has to combine suitable hardware and software components so that the object's creation environment or a suitable equivalent can be reproduced. As many original system environments cannot be run on today's computer hardware, emulated hardware is used. Besides supporting the requirements of the original environment, the emulator has to be equipped with appropriate interfaces for automation and framework integration [20].

No matter which emulator is chosen, contextual information about the original environment in which the digital artifact was created is also required. For example, the answers to questions such as "Which operating systems is WordStar 7.0 compatible with?" are less obvious today than twenty years ago. To overcome this knowledge gap, the process of computing the actual needs for an authentic rendering environment is formalized by the concept of view path [19, 17].

3.1.1 Container Preparation

In order to perform the migration using the original tools in their original environments, it is necessary to provide these tools in order to operate on the digital objects of interest. The objects must be injected into the emulated environment from the actual one, migrated and, finally, derived objects produced by those applications must be extracted. Injection/extraction of data into/from the emulated environments is possible via emulated data storage devices, e.g. via floppy, hard disk, and CD-ROM drives or via virtual network connection if supported by the emulator and the operating system. These can be seen as gateways for binary data exchange between the real and emulated environments. Usually emulators support the emulation of at least one storage device.

For example, with QEMU it is possible to activate the emulation of a floppy drive with a virtual floppy disk using the following argument pair: `qemu -fda floppy.img`. Here the file floppy.img refers to a virtual floppy disk prepared in the actual working environment of the user. The file contains data in the form of files subject to injection into the emulated one. Any modifications performed on the virtual floppy disk inside the emulated environment will be reflected in the corresponding image file. It can afterward be mounted onto the filesystem of the actual system in order to acquire the modified data.

Floppy disks are standardized for a wide range of different computer platforms and are a comparatively simple solution, but they are limited in size. An alternative method of data injection/extraction is the use of hard disk drives. They are modifiable and their size can be adjusted as desired. This allows for the injection of large quantities of DOs into the emulated environment. The production of empty hard disks can be automated, similar to the process of creating floppy images. First, the desired amount of storage space for the disk image is allocated. In a second step, a disk partition with a suitable file system has to be created. The first partition in the hard disk starts at 32256 byte offset. Finally, starting from the aforementioned offset, the allocated storage is formatted with the required file system.

3.2 Migration Component

The *migration component* (MC) is the main module visible to the end user, by exposing a simple *migrate* interface for (possibly) complex DO migration from format fmt_A to format fmt_B. The user requests a migration by providing a (set of) digital object(s) to be migrated, the requested final format, and a set of parameters. These parameters may restrict the migration path length set quality or cost criteria for the migration process. Based on these criteria individual migration steps are identified. Figure 2 illustrates the general mode of operation of a MC.

In order to ensure the remote-accessibility requirement, this component is to be implemented in the form of a Web service. Its integration into existing DP frameworks would allow its usage both as a stand-alone tool and as a part of more sophisticated preservation workflows. The PLANETS Interoperability Framework (IF) [8] is a suitable candidate for the integration requirement. In this framework each preservation tool can be invoked according to one of the predefined code interfaces. The chosen interface depends on the tool's role in the scope of the DP (e.g. object migration, characterization, viewing, comparison).

On a low level, the migration component would then represent the implementation of the PLANETS IF *migrate* interface. The migration-by-emulation Web service could be invoked by using it's WSDL description file.

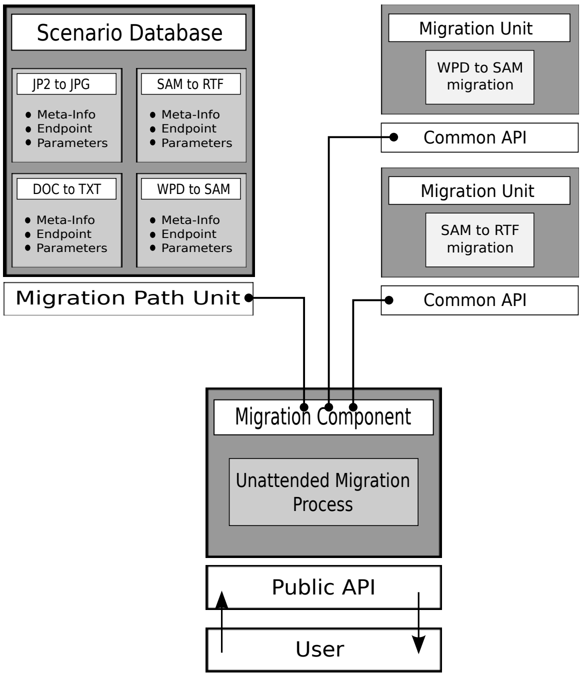

Figure 2: Migration component overview.

Based on the resulting identified migration path, the MC instantiates each node as a single migration unit. Beside this, the MC takes care of intermediate results and, if necessary, error reporting and recovery.

3.3 Migration Scenario

A migration scenario database describes atomic units for migration from format ftm_A to fmt_B and maintains available information about all atomic migration scenarios. "Atomic scenario" refers to a scenario not involving intermediate migrations. Such a scenario consists of a Web service endpoint and necessary meta-information (e.g. input/output formats, efficiency level, single/multiple DO support, author, timestamp). The input/output format identifiers need to be defined according to a conventional format registry system (e.g. Pronom[4]). Optional parameters may be required at instantiation time.

More complex format migrations cannot be carried out in a single step. Usually, several intermediate steps are required,

[4] The technical registry PRONOM, *http://www.nationalarchives.gov.uk/pronom*

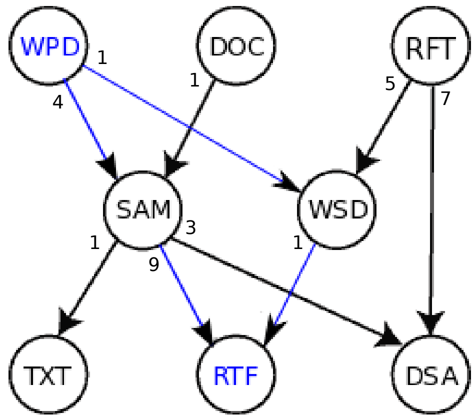

Figure 3: Migration graph. Nodes depict file formats, edges depict cost or quality weights.

but also several distinct paths are available between fmt_A and fmt_B. Hence, the scenario database can be represented as a directed graph with nodes representing supported file formats and edges describing weights, for instance based on the resulting quality or cost of a specific format migration. Figure 3 shows a possible migration graph between various text document file formats.

Depending on the migration requested by the user, the complete migration paths corresponding to it are to be calculated, being formed from the atomic ones. At this step the path preferences specified by the user in the parameter list can be taken into account. After the calculation, the component receives the path in the form of ordered lists of atomic scenario identifiers, which then will be instantiated and controlled by the migration component. In a distributed sce-

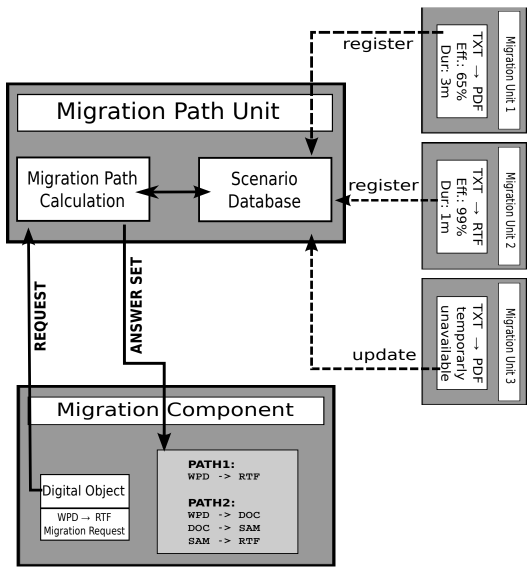

Figure 4: Migration path generation and calculation.

nario, different sites can register migration units in a centralized scenario database. Furthermore, based on recent runtime results and/or user feedback, quality measures like authenticity, reliability and runtime costs can be updated in the database (illustrated in Fig. 4).

For instance, a WPD (Word Perfect 6.0) to RTF (Rich Text Format) migration is requested. A user invokes the migration component by providing the required input files: an archive of DOs, input format – WPD, output format – RTF, and a parameter defining a maximum path length of two atomic migration steps. The migration component queries the scenario database through the migration path unit and receives a set of different migration paths. Among them only one path satisfies the maximum migration path length condition: WPD to SAM and SAM to RTF. First, a WPD to SAM migration is performed. The endpoint defined in the corresponding bundle is used to invoke the responsible migration unit. The meta-information of the bundle contains the field indicating that this service is able to perform the migration of multiple DOs in one operation.

Additionally, migration scenarios which produce two different output objects of different formats from one original artifact would be possible. Therefore, the DOs can be passed in an archive container (e.g. ZIP). After a migration an archive of migrated DOs is returned along with the success status.

According to the meta-information describing the second migration unit (SAM-to-RTF), the service is unable to perform the migration of multiple DOs in one turn. Therefore, the archive of intermediate DOs is extracted by the migration component to a local temporary storage. From there the intermediate DOs are migrated one by one and the results of the migration are retrieved. The resulting DOs in RTF format are packed into an archive and returned to the user.

If a representative test set of input and output files are kept for each migration scenario, software updates or other minor changes on the emulated system environment can be tested in an automated way for compatibility.

Such migration scenarios can then be registered with the preservation framework as simple migration services.

3.4 Migration Units

The combination of emulator and original environment is itself not sufficient for automated migration workflows. It just provides the base to view, modify or migrate a digital object manually. Thus, a *migration unit* (MU) combines the system environment with an interactive workflow description (IWD). The IWD is an abstract description of a recording of all user interactions of a specific task. Such interactions of a human user with the computer UI of the original applications is represented by a series of input events, such as mouse clicks/movements and key strokes. These events can be simulated using remote desktop control systems like VNC using an application like VNCplay [25], allowing them to be performed in an unattended manner [10].

3.4.1 Workflow Recording

To create a specific MU the original system environment is to be combined with an IWD describing the required user interaction. This is realized by running actions corresponding to manually prerecorded migrations, for instance a WPD-to-RTF conversion using the original application WordPerfect 6.0 in Windows 3.11. The input events produce the following actions in the emulated environment:

1. The WordPerfect 6.0 is executed via a mouse click on its icon.
2. The "Open" menu of Word Perfect 6.0 is chosen.
3. In the opened dialog box, the DO on the attached hard disk drive is chosen and loaded.
4. The "Save As" menu of Word Perfect 6.0 is chosen.
5. In the opened dialog box the new file name is chosen according to the conventional naming scheme and the DO is saved in RTF format on the same hard disk. The migrated object is produced at this point.

Fig. 5 illustrates the workflow of a migration unit creation. These or similar workflows are recorded by the *scenario preparation component* to generate abstract interactive workflow description for specific user interaction. A more complex example could be setup as follows: An original environment contains an AMI Pro 3.1 installed on Windows 95 with a virtual Postscript printer running in QEMU. This allows the recording of several different scenarios:

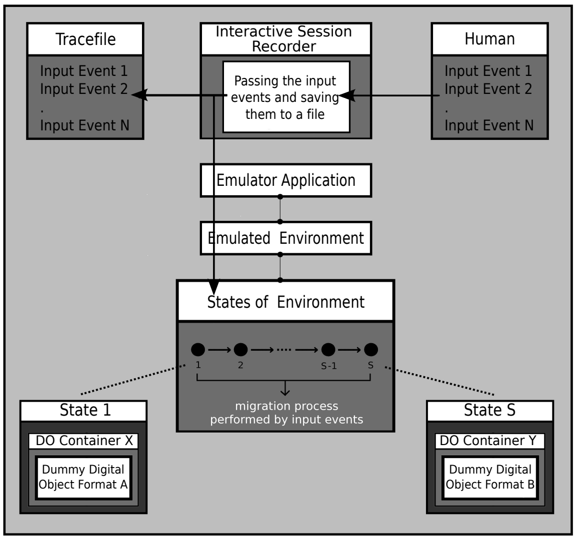

Figure 5: Workflow of creating a complex migration unit.

- Lotus AMI Pro to TXT
- Lotus AMI Pro to RTF
- Lotus AMI Pro to Postscript (PS) using a virtual printer
- Combinations of the aforementioned atomic migration steps, to produce two different objects from one input, like TXT and PS

- Non-native formats by different input filters available from AMI Pro

The last one might help to deal with unavailable applications for certain deprecated formats or could provide a base for rendering comparisons of other tools. All the mentioned options use exactly the same base environment.

The workflow recording is then kept as metadata for a certain stored system environment which is provided by some backend store keeping the software archive [21, 16] of original environments. There could be several recordings registered with one original environment. In theory – as this has not been thoroughly tested yet – the emulator should be exchangeable with a computer architecture compatible type, as the original system should perform roughly the same way on the other emulator. This was tested for a chain of updates over a number of different versions of QEMU [22].

3.4.2 Workflow Replication

The recorded set of actions is later executed by the MU for each DO in an automated manner. After sending the last input event and observing its expected outcome, the interactive workflow replication service finishes its work. The workflow replication service sends a request for the termination of the emulated environment, using the current session identifier. The emulation service frees all resources, detaches the hard disk image with the newly migrated DOs and returns it to the workflow replication service.

The operating system and the original tool are used: in this case Windows 3.11 and WordPerfect 6.0 are installed. According to the workflow, the migration is invoked by providing it with the archived DOs and optional parameters. The workflow replication service parses the optional parameter list and acquires the references to the three necessary objects: a suitable system environment, the interaction workflow description and the target emulator. It then extracts the received DOs from the archive to the local storage, while calculating their total size. The DOs are renamed according to the predefined naming scheme and the correspondence between the real and virtual names is stored separately. The next step is the DO container creation and injection of data.

The DO container creation method is invoked and information regarding the desired filesystem of the hard disk drive and its size is provided. The size is set to a value large enough to hold the DOs and the resulting output files, which will be produced in the emulated environment. The DO container creation subunit acquires the input data and produces the hard disk image file. It then injects the DOs without changing their filenames. After the operation is completed, it returns the prepared container to the workflow replication service.

When the DO container is prepared, the emulated system environment can be started. The workflow replication service invokes one of the emulation service endpoints stored in the scenario database and one of the suitable emulator associated with it. It then invokes the respective emulation service by providing the following: the emulator ID, the operating system image of the system environment object, any secondary objects and/or directives necessary for the emulation of this operating system, and the prepared container with the DOs in it.

The system emulation service receives the input data and starts the appropriate emulated system environment. It also attaches the hard disk container with the DOs. The remote desktop control is activated. The system emulation service returns a success message with a port number for connection as well as the session identifier. The workflow replication service then initiates the replication of an interactive session.

4. DP-FRAMEWORK INTEGRATION

The Java-based migration-by-emulation service prototype provides all main components. Those include basic migration units, migration component services as well as a simple scenario database. A further challenge is the integration of the developed components in standard DP-frameworks (e.g. PLANETS) and their associated workflows. The specific objectives to be pursued involve enabling interactive user access to original system environments to create new environments, create or modify existing ones but also to create abstract interactive workflow descriptions. Having original environments and IWD for certain file types, these migrations have to be made available as simple steps of more complex DP-workflows.

4.1 Preparation User Interface

The PLANETS *view* Web service interface is designed to render an arbitrary digital object. The service takes a digital object and returns an URI which points to the rendered result. If the digital object requires a running rendering engine, the service offers methods for querying the engine's state and allows sending commands to it. This interface is suitable to integrate the user preparation and recording workflows within the PLANETS framework.

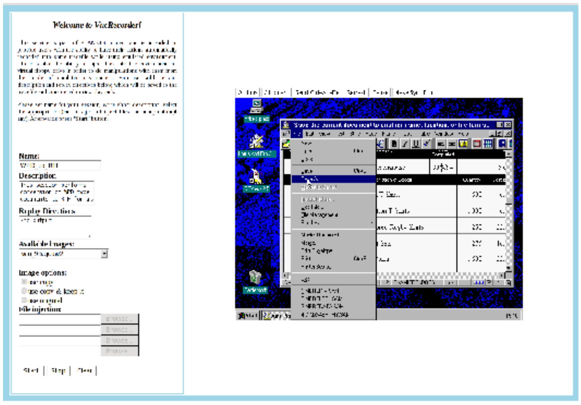

Figure 6: Grate-R interface to the PLANETS service for interactive workflow recording.

Preconfigured original environments are deployed and the Grate-R (cf. Section 2) VNC record service is used to generate an abstract interactive workflow description. They were then attached by hand to the appropriate original environment to form a proper migration unit.

4.2 Testbed Migration Workflows

The user interface to existing migration-by-emulation workflows is realized via the standard PLANETS testbed GUI [14, 1]. The services are called from within the testbed standard procedures. As migration-by-emulation services should be accessible the same way as standard command line tools they are registered and deployed using the same methods within the testbed. Thus, the testbed user interacts with the Web interface in the usual way.

Since there is no suitable PLANETS wide tool registry available supporting complex view path calculation, the migration path computation is not part of the framework yet. The PLANETS testbed was originally designed to retrieve the available (migration-)services from the PLANETS service registry. Each deployed service registers itself and describes its capabilities (here migration paths) using a `describe()` method. However, migration-by-emulation services can become more complex then input-output oriented migration tools or methods (e.g. complex migration graph). The *migration component* was designed to hide some complexity from the user e.g., automated complex path selection based on cost parameters provided by the user.

Due to the construction of the PLANETS framework, either single step migrations or preselected complex migration paths can be exposed to the testbed users. More complex migrations have to be constructed within the framework's capabilities. To provide all migration options to the PLANETS testbed user, a dynamic `describe()` method can be used, retrieving all available endpoints from the scenario database. The coupling of atomic migration units depends on the user for now.

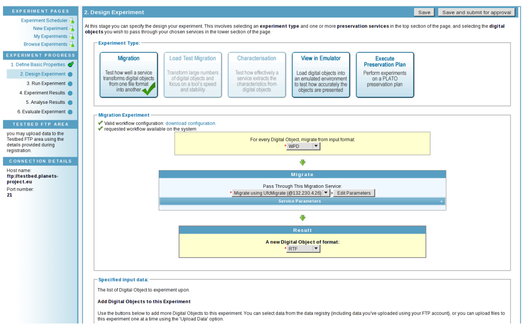

Figure 7: Selecting the Word Perfect to RTF migration in the Testbed.

4.3 Experiments and Results

Two different migration services were registered within the testbed. One of them is a truly atomic migration accepting WPD as input and producing RTF as output. Thus the resulting file is directly delivered to the user after the procedure succeeds (Fig. 7). The second service is more complex as it takes an AMI Pro text document (SAM) as input and produces two different outputs, a TXT and a PDF. The TXT is the result of a classical "save-as" migration. The PDF is generated by sending the document to a virtual printer generating PS as output. This file is then loaded to the Ghostview application, which renders a pdf from it.

Additionally, a virtual disk handling service was programmed to produce disk image containers for different emulators with the option of specifying a range of supported filesystems understood by the original system environments. The creation of a QEMU compatible container with a FAT filesystem is comparably simple, other containers and filesystems are supported as well by using the "qemu-img" container conversion tool.

The WPD to RTF service deploys the original DOS Word Perfect 6.0 application running in QEMU using a Windows 3.11 environment with mouse and keyboard interaction (Fig. 8). The procedure was tested with a small number of different WPD files.

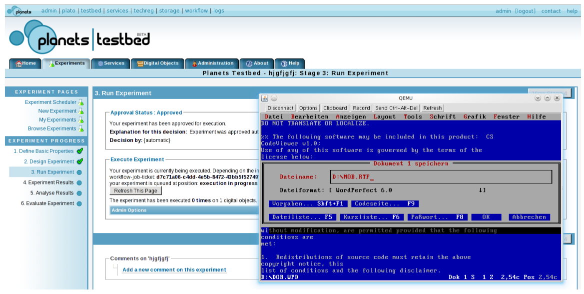

Figure 8: Running Word Perfect migration in Testbed. QEMU screen output attached for monitoring purposes.

The second migration unit was deliberately made to be more complex. We wanted to demonstrate the feasibility of producing more then one output file from a single input. This helps to evaluate and compare different workflows regarding runtime, reliability and complexity. The experiment was run on a set of a couple AMI Pro files containing between three and 15 pages. The average runtime took 2 minutes 33 seconds to complete. Of course it would have been possible to execute the PS to PDF migration by a simple command line tool in another atomic migration. But we wanted to stress the interaction playback component on a longer running workflow involving more than a single application. The procedure did not fail in all experiments we ran on the testbed.

After invoking the migration service locally the same test was repeated by calling the service over the Internet from some geographically different PLANETS instance. It ran successfully and the different execution time difference for the conversion was insignificant.

Due to limited capacity in time and the small number of available objects for testing, the evaluation of the migration services was yet short. Thus the figures collected need still to be proven for a larger number of objects running in migration processes on different machines.

A major drawback in all experiments was the availability of a proper set of test files. Such a ground truth set of files enriched with descriptive metadata on features like length, containing pictures, special font sets, complex layouts and other features would be very helpful to evaluate and compare preservation workflows and migration tools.

4.4 Next Steps

Still a problem is the black box character of the running migration unit (cf. [11]). In the present state of development, not much feedback on the state of active migration workflows is available and reported back to the user. While it is generally possible to attach another VNC viewer to the running emulator the migrate interface lacks any methods for querying its execution status in run-time. Thus, research at Freiburg University focuses on the VNCplay tool and the files used by it. The IWD file could be exploited to gather information on the actual state of an ongoing migration. Every interaction and expected screen result relates to a certain state of the process. This could be used to generate progress information or produce more meaningful error messages. Having this in place, the runtime of the entire process should be optimized by eliminating redundancy in the IWDs.

In future steps the authors hope to enhance the service to handle more emulators beside QEMU and include additional non-x86 original environments. Plus, more runtime improvements like system resume-restart or multi-object migrations will be looked into. The additional experience and information should help for a better preservation planning by providing quality metrics and cost estimations [3].

There are interesting alternative approaches like Polyglot for automated file conversions. The service was developed by [15] originally to convert 3D model files into different formats. The underlying concept is quite different to the presented one and is worth comparing and benchmarking.

5. CONCLUSION AND OUTLOOK

Migration-by-Emulation services allow a wide range of different file format conversions. Compared to simple command line migration tools those services are more complex to setup and deploy. Nevertheless after having the workflow established a new migration scenario is simpler to be integrated. Often only a new workflow is to be recorded to add another service like the conversion of WPD to TXT or Postscript. Even if a completely new input like Wordstar files are to be supported, only an appropriate original environment is to be extended without the requirement to program a new tool wrapper. Thus, the system presented allows to easily compare the migrations run on different original environments.

The operation and management of migration-by-emulation services as presented could be decentralized and several institutions could share the workload or specialize on certain environments and share their expertise with others.

Our implementation focused on the feasibility of the preservation framework integration. Future research is dedicated to the speedup of workflows by looking into the VNC recording and playback. The interactive workflow descriptions are a good starting point for optimization. They could be enriched with additional metadata to use it for progress reporting. A certain state in the metadata directly correspondents to a state of the migration workflow and could be reported back to the preservation framework.

The interative workflow description could be modularized to better identify the different stages, like original operating system booting, application starting, artifact loading, and saving in a new format. This information could not only be used for feedback but to identify checkpoints. Those checkpoints could help with error recovery for restarting the procedure after failed attempts. Additionally, these workflows could help to evaluate future versions of emulators before they get integrated into preservation systems [22]. These issues are part of the ongoing research at Freiburg University.

Depending on the future needs arising from the requirements of Migration-by-Emulation, a couple of different strands seem to be worth exploring in the coming years. Especially the splitting of the preservation application into a simple user's front-end and an easy-to-extend server backend is very attractive, as it could be adapted to many preservation framework services. Nevertheless, future emulator research and development should take the "preservation-awareness" regarding automation and long-term support more into consideration. With the ongoing research in the KEEP[5] or SCAPE[6] projects new solutions and progress in wrapping emulators and handling large scale migrations could be expected. Beside this, new insight into emulation metadata and tool registries [2] will help to automate more steps of the migration-by-emulation workflow. Nevertheless, a number of challenges like software archiving, emulation knowledge base and a proper definition of a test set of digital artifacts remain open.

6. REFERENCES

[1] B. Aitken, S. Ross, A. Lindley, E. Michaeler, A. Jackson, and M. van den Dobbelsteen. The planets testbed – a collaborative research environment for digital preservation. In M. Lalmas, J. M. Jose, A. Rauber, F. Sebastiani, and I. Frommholz, editors, *Research and Advanced Technology for Digital Libraries, 14th European Conference, ECDL 2010, Glasgow, UK, September 6-10, 2010. Proceedings*, volume 6273 of *Lecture Notes in Computer Science*, pages 401–404. Springer, 2010.

[2] D. Anderson, J. Delve, and D. Pinchbeck. Towards a workable, emulation-based preservation strategy: rationale and technical metadata. *New review of information networking*, (15):110–131, 2010.

[3] C. Becker, H. Kulovits, M. Kraxner, R. Gottardi, A. Rauber, and R. Welte. Adding quality-awareness to evaluate migration web-services and remote emulation for digital preservation. In *Proceedings of the 13th European Conference on Digital Libraries (ECDL09)*, 2009.

[4] V. Delwardia, S. Marshall, and I. Welch. Experiments in Remote Mobile Gaming. In *AUIC: Australasian User Interface Conference*, 2009.

[5] A. Farquhar and H. Hockx-Yu. Planets: Integrated services for digital preservation. *International Journal of Digital Curation*, 2(2), 2007.

[6] E. Genev. VNC Interface for Java X86-Emulator Dioscuri. Online, http://hdl.handle.net/10760/15102,

[5]See the KEEP emulation framework information plus development pages at *http://www.keep-project.eu* and *http://emuframework.sourceforge.net*

[6]See project pages at *http://www.scape-project.eu*

October 2010.

[7] M. Guttenbrunner, C. Becker, and A. Rauber. Keeping the game alive: Evaluating strategies for the preservation of console video games. *International Journal of Digital Curation*, 5(1), 2010.

[8] R. King, R. Schmidt, A. N. Jackson, C. Wilson, and F. Steeg. The planets interoperability framework. In *Proceedings of the 13th European Conference on Digital Libraries (ECDL09)*, pages 425–428, 2009.

[9] D. Pinchbeck, D. Anderson, J. Delve, G. Alemu, A. Ciuffreda, and A. Lange. Emulation as a strategy for the preservation of games: the keep project. In *DiGRA 2009 – Breaking New Ground: Innovation in Games, Play, Practice and Theory*, 2009.

[10] K. Rechert and D. von Suchodoletz. Tackling the problem of complex interaction processes in emulation and migration strategies. *ERCIM News*, (80):22–23, 2010.

[11] K. Rechert, D. von Suchodoletz, and R. 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.

[12] K. Rechert, D. von Suchodoletz, R. Welte, M. van den Dobbelsteen, B. Roberts, J. van der Hoeven, and J. Schroder. Novel workflows for abstract handling of complex interaction processes in digital preservation. In *Proceedings of the Sixth International Conference on Preservation of Digital Objects (iPRES09)*, 2009.

[13] T. Richardson. The rfb protocol. Online, *http://www.realvnc.com/docs/rfbproto.pdf*, 2009.

[14] R. Schmidt, R. King, A. Jackson, C. Wilson, F. Steeg, and P. Melms. A Framework for Distributed Preservation Workflows. *International Journal of Digital Curation*, 5(1), 2010.

[15] University of Illinois at Urbana-Champaign, NCSA. Towards a universal file format converter. Online, *http://isda.ncsa.uiuc.edu/NARA/conversion.html*, 2011. Online resource.

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

[17] J. van der Hoeven and D. von Suchodoletz. Emulation: From digital artefact to remotely rendered environments. *International Journal of Digital Curation*, 4(3), 2009.

[18] R. Verdegem and J. van der Hoeven. Emulation: To be or not to be. In *IS&T Conference on Archiving 2006, Ottawa, Canada, May 23-26*, pages 55–60, 2006.

[19] D. von Suchodoletz. *Funktionale Langzeitarchivierung digitaler Objekte – Erfolgsbedingungen für den Einsatz von Emulationsstrategien*. Cuvillier Verlag Göttingen, 2009.

[20] D. von Suchodoletz. A Future Emulation and Automation Research Agenda. In J.-P. Chanod, M. Dobreva, A. Rauber, and S. Ross, editors, *Automation in Digital Preservation*, number 10291 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2010. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Germany.

[21] D. von Suchodoletz. Das Softwarearchiv – Eine Erfolgsbedingung für die Langzeitarchivierung digitaler Objekte. *Mitteilungen der Vereinigung Österreichischer Bibliothekarinnen und Bibliothekare*, (63):38–55, 2010.

[22] D. von Suchodoletz, K. Rechert, and A. N. Tchayep. QEMU – A Crucial Building Block in Digital Preservation Strategies. In W. Müller and F. Pétrot, editors, *1st International QEMU Users' Forum – DATE 2011 Workshop*, Grenoble, France, 2011.

[23] D. von Suchodoletz, K. Rechert, R. Welte, M. van den Dobbelsteen, B. Roberts, J. van der Hoeven, and J. Schroder. Automation of flexible migration workflows. *International Journal of Digital Curation*, 2(2), 2010.

[24] R. Welte. *Funktionale Langzeitarchivierung digitaler Objekte – Entwicklung eines Demonstrators zur Internet-Nutzung emulierter Ablaufumgebungen*. Südwestdeutscher Verlag für Hochschulschriften, 2009.

[25] N. Zeldovich and R. Chandra. Interactive performance measurement with vncplay. In *ATEC '05: Proceedings of the annual conference on USENIX Annual Technical Conference*, pages 54–64, Berkeley, CA, USA, 2005. USENIX Association.