iPRES 2009

THE SIXTH INTERNATIONAL CONFERENCE ON THE PRESERVATION OF DIGITAL OBJECTS

Proceedings

October 5-6, 2009
Mission Bay Conference Center
San Francisco, California

California Digital Library

Novel Workflows for Abstract Handling of Complex Interaction Processes in Digital Preservation

Klaus Rechert
University of Freiburg
Hermann-Herder-Str. 10
79104 Freiburg, Germany
klaus.rechert@rz.uni-freiburg.de

Dirk von Suchodoletz
University of Freiburg
Hermann-Herder-Str. 10
79104 Freiburg, Germany
dirk.von.suchodoletz@rz.uni-freiburg.de

Randolph Welte
University of Freiburg
Hermann-Herder-Str. 10
79104 Freiburg, Germany
randolph.welte@rz.uni-freiburg.de

Maurice van den Dobbelsteen
National Archives of the Netherlands
Prins Willem Alexanderhof 20
2595 BE Den Haag, The Netherlands
maurice.van.den.dobbelsteen@nationaalarchief.nl

Bill Roberts
National Archives of the Netherlands
Prins Willem Alexanderhof 20
2595 BE Den Haag, The Netherlands
bill.roberts@nationaalarchief.nl

Jeffrey van der Hoeven
Koninklijke Bibliotheek
Prins Willem Alexanderhof 5
2595 BE Den Haag, The Netherlands
jeffrey.van.der.hoeven@kb.nl

Jasper Schroder
IBM Netherlands B.V.
Johan Huisingalaan 765
1066 VH Amsterdam, The Netherlands
jasper@nl.ibm.com

Abstract

The creation of most digital objects occurs solely in interactive graphical user interfaces which were available at the particular time period. Archiving and preservation organizations are posed with large amounts of such objects of various types. At some point they will need to, if possible, automatically process these to make them available to their users or convert them to a valid format. A substantial problem in creating an automated process is the availability of suitable tools. We are suggesting a new method, which uses an operating system and application independent interactive workflow for the migration of digital objects using an emulated environment. Success terms for the conception and functionality of emulation environments are therefore devised which should be applied to future long-term archiving methods.

Introduction

The later preservation of digital objects poses completely different requirements from those at the original creation of the objects. The creation of most digital objects occurs solely in interactive graphical user interfaces which were available at the particular time period. For example an important long-lasting spare part for the aviation industry was designed with a specific CAD program which by now has disappeared from the market. Or, a museum has the rights from a famous author who wrote his books in a word processor on the Amiga, a now obsolete personal computing platform.

Archiving and preservation organizations already have a large quantity of such objects in various types and the scale is only growing. The organisations have to make the objects available to their users now and they have to safeguard the digital longevity of these objects. Consequently, at some point the organisations will need to make a presentation copy of the objects and they will need to convert the objects to a current, sustainable file format. Due to the scale the only financially and organisationally feasible way to support both actions and to achieve both goals is some sort of automated process.

The Interactivity Problem

A substantial problem in creating an automated process is the availability of suitable tools. A digital object is in most cases best viewed in the application it was created or in its original environment. Many of these programs were designed as interactive applications, while most are without interfaces to automation. For example, this means that graphic, product design, audio/video or word processing programs cannot perform basic tasks such as the opening and saving of a file in another format as an unattended and fully automated task. The attempt to add such functions (van der Hoeven, van Diessen, and van der

Meer 2005) to an application is generally very complex and sometimes even impossible since the source-code and the required knowledge are no longer available. For a large number of applications the cost would be very high and would only be worth the effort for very popular formats. Also, it is becoming increasingly difficult to find suitable staff which is familiar with the always ageing user environments of older computer systems. It is not possible to rely on the availability of import functions in current products. Additionally, human interaction in very repetitive tasks is error-prone and expensive.

The traditional approach to help the user to automate interactive tasks to a certain degree is the use of so-called macro-recorders. These are specialized tools or functions of an application or operating system user interface to capture sequences of actions carried out, e.g. create a new file, open the address database and selecting an address, copy some text and save or print the file for serial letters. However, this functionality is not standardized, differs in its usability and features. Special tools might be needed and the knowledge of the applications and operating systems is required. So, the approach suggested here predicts that there occurs a technical and organizational separation between the machine used for workflows and the in/output. As is shown here, emulated or virtualized environments are particularly well suited for this.

Given these challenges we are suggesting a new method, which uses an operating system and application independent interactive workflow for the migration of digital objects using an emulated environment.

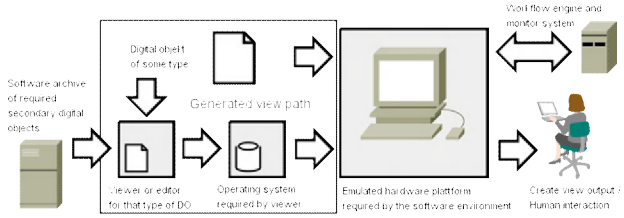

Figure 1: Emulation workflow deploying a View-Path to operate on a digital object executed by a special workflow processor.

View-Paths as Rendering Formalization

Digital objects of interest, labeled as primary objects within this paper, cannot be used by themselves, but require a suitable context to be accessible. This context must combine hardware and software components in such a way that the creation environment or a suitable equivalent is generated for the primary objects.

The reproduction of suitable environments for a digital object of interest or corresponding equivalents is formalized by so called View-Paths or pathways. These are directions from the primary object of interest into the actual environment of the user (Oltmans, van Diessen, and van Wijngaarden 2004; von Suchodoletz and van der Hoeven 2008). Figure 1 illustrates the application of a View-Path, showing a typical vector originating from the primary object to its creation application, the required operating system up to the resultant hardware emulator. The additional components needed in this process are called secondary objects.

Having an emulator and contextual information contained in a View-Path, some implications are still left before the digital object can be rendered. First, the digital object has to be characterised, e.g. by using services like PRONOM (The National Archives 2007). Then, depending on the application, the operating system or necessary level of hardware-emulation, a number of additional software, so called secondary objects, like applications, helper programs and drivers need to be taken into account. There might exist different View-Paths for each object type and the number probably increases with the number of object types.

Because certain object types have a relatively high complexity, some issues need to be considered about how a View-Path is computed. Especially for frequently requested View-Paths it could be conceivable to work with prepared, cached environments. Altogether, this leads to the following requirements for managing the archive:

Creation of a Background Archive: In this case every single object needed to create a certain View-Path is permanently stored within the archive. These additional objects are to be kept like the primary objects of interest. At this point it could be considered whether the View-Path objects like the emulators, operating systems, specific helper software and description are bundled together in a single package or stored individually.

Operation of an Online Archive for Direct Access: For frequently requested secondary objects it is more efficient to store these in a special archive, additional to the long-term archive – both to reduce the load on the long-term archive and to improve the process of generating arbitrary View-Paths.

Setup of a View-Path Cache: For often requested and more complex View-Paths, the use of a prepared working environment can reduce the work for users and archive operators. This cache either is part of the online archive or is directly available on the reference platform. A major challenge archivists face is the interactive character of most of the software components involved. Hence, some steps need to be executed through direct machine interaction. Once the View-Path is created for rendering primary objects of a certain type, it could be reused on objects of the same type.

Ad-hoc View-Path Generation Versus Prefabricated View-Path Archive

In the moment of digesting a primary object from the archive for object rendering, it needs to be processed and then executed, automatically or with the interaction of the archive user or archivist. These work steps imply not only copying and reproducing the object's bit stream, but actually providing access to the object in a sensible way. At this point emulation and migration strategies do not differ much: The procedures for the object reproduction could be both described by the aforementioned View-Paths. Additionally, independent View-Paths e.g. one for the migrated object and one for the original one and its emulated environment, might help ensuring its authenticity.

For the actual deployment of View-Paths mainly two different approaches could be identified:

1. Generating the software stack from scratch directly from the components stored in the software archive.
2. Using prefabricated setups generated once by an archivist and then stored as an emulator container file in the View-Path cache of the archive.

Of course any variant in between would be possible too. While the first option would require lots of repeated manual work or automation (e.g. using the ideas presented in this paper) it would reduce intermediary items to be stored and updated if the emulation environment changes. The latter reduces the delivery time and complexity of often requested View-Paths.

Each item of the View-Path cache needs to be described in the software object repository to be chosen accordingly and compared to the multi-item on demand View-Path setup.

In any case the archive user needs a certain set of tools to access the object. These are often additional software which has to be kept in the archive too or in a specialized software repository.

Object Exchange Challenges

The loading of primary objects is a major part of any automated processing setup: The objects need to be passed on into the emulated environment. This is typically not a trivial task and depends on the feature-set of the original environment.

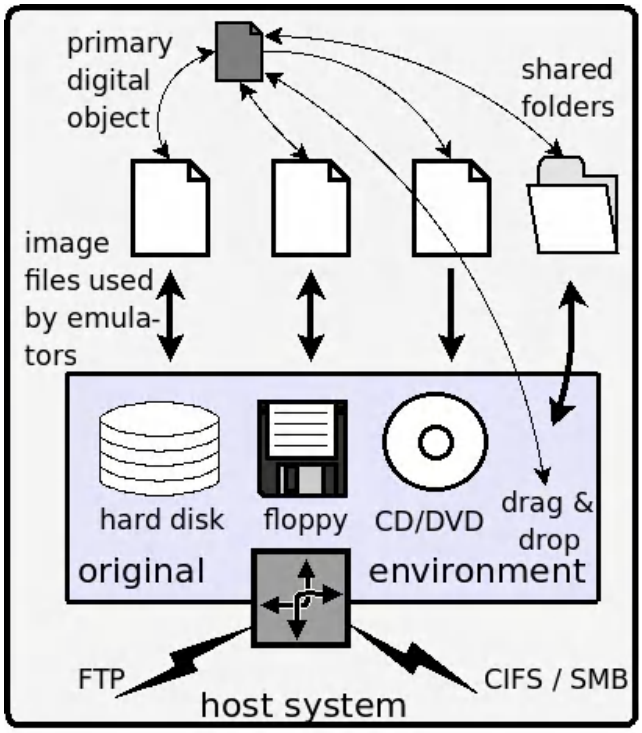

Figure 2: Depending on the recreated environment (and type of emulation) several options for the transport of the digital objects exist.

Network-Transport

Many of the newer and advanced operating systems offer low level access to standard network interfaces. Such interfaces were widely integrated in the early eighties but first into the high-end commercial systems. Mid of the nineties a network interface became standard of every desktop machine. In order to use these interfaces it often required the installation of additional software like hardware drivers and network protocol stacks. In the beginning, there was a wider range of higher level network protocols in use, but in the nineties the Internet Protocol (IP) became more dominant and soon a quasi-standard. Support for the various protocols were usually not part of the standard features of ancient operating systems. Thus, additional software had to be purchased and installed separately. These software packages have to be conserved in the software archive too.

The main advantage of the network-transport is the synchronous operation: While running the reference environment any exchange of objects is possible in both directions. The size of files is only bound to limitations of the deployed ancient operating system.

The FTP is one of the oldest protocols using TCP/IP. It is around for more than 30 years and has not changed very much. In the beginning, there were only simple command line tools with small footprint. Now, most of the modern operating systems or additional tools implement

comfortable front ends to this protocol. The NCSA telnet package contains an FTP client as every Windows version from 98 on. It was absolutely standard for the different Unixes from nearly the beginning.

The SMB (Server Message Block) protocol, and its successor CIFS (Common Internet File System) are not a simple file transport protocols but also network file systems. Thus, they offer more convenient transportation of files back and forth and furthermore these features are directly integrated into the standard file system of the running operating systems. It was originally invented by IBM with the aim of turning the DOS Interrupt 33 (21h) local file-access into a networked file-system, later on Microsoft made considerable changes to it. SMB is available in Windows since the Workstation release of 3.11 and Windows NT. It is also implemented for the different Unixes and similar systems and known as the Samba software package1. The existence of many implementations of SMB/CIFS, especially the Open Source Samba package[1], should offer a long time support for that protocol well after its demise.

Container Files

Emulators usually use so called container files as virtual hard-disk images. Therefore, they offer an option to transport a digital object into the emulated environment by embedding it in the container file, or by creating a secondary one, which is then attached as an additional virtual hard-disk.

However, for producing or modifying such containers, exact knowledge of the internal format and the used file system is required. Furthermore, a modified version of the digital object is only available after the emulator is stopped. Otherwise any changes to the block-layer might corrupt the container. Thus, no online access to digital objects like network connections is possible.

ISO and Floppy Disk Images

Both devices are typically removable in contrast to virtual hard-disks discussed in the aforementioned section and thus offer a data exchange option while the emulator or virtualization tool is running.

The emulator has to support virtual media loading and ejecting functionality, otherwise media changes might not get noticed. Not all hardware platforms and operating systems support optical drives (e.g. ISO-9660 images), but most of them support floppy disks.

Interactive Preservation Workflows

The idea is to interactively record a particular workflow once, such as loading an old Lotus AMI Pro document in its original environment and converting it to a PostScript by printing with an suitable printer driver into a file. Such a recording can serve as base for a deeper analysis and generation of a machine script for then completely automated repetition.

We define an interactive migration workflow as ordered list of interactive events which are passed on to the emulated environment through a defined interface. These events can for example be mouse or keyboard events, but are however not limited to just these. Each of these events is linked with a precondition and an expected outcome which can be observed as a state of the emulated environment. Before this effect is not seen, the next event cannot occur. To link events with special preconditions and outcome is necessary since the workflow is depending on the level of capacity of the emulation environment: programs will take different amounts of time to run depending on the load of the hosting machine. In the interactive case, this occurs e.g. through optical control of the user. For an automated run, the definition of expected states and a reliable verification is indispensable.

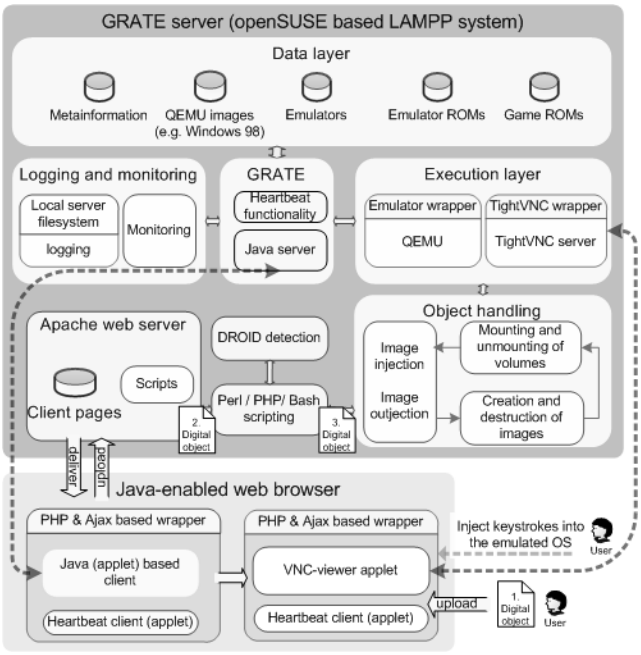

Figure 3: GRATE architecture.

Unattended Interactive Migration

In the scope of the PLANETS (Farquhar and Hockx-Yu 2007) project a prototype was developed using GRATE which allows the wrapping of various software environments within a single application. It provides the archive user an abstract interface independent of digital objects (Welte 2009; Becker et al. 2009). GRATE uses VNC to reach an abstraction of a very wide range of different hardware architectures (cf. Fig. 3).

Vital parts of the GRATE functionality can be realized with the modular Open Source emulator QEMU. It offers emulation of a wide range of hardware architectures, VNC

[1] The Samba Project, http://www.samba.org [9/8/2009]

support as well as hardware-monitor interfaces. Screen output and input via mouse or keyboard—which up till now are still the most used methods of human-computer interaction—can be well simulated and observed using the open VNC protocol. Thus, this setting is making available an abstract interface which can be used for computer interaction.

Synchronization and Machine Monitoring

In order to create interactive preservation workflows, a technique producing and checking pre- and post-conditions in a reliable way is necessary. Best suited for the recording and verification of the effects of interactive events are various technical approaches and/or combinations of these. To observe the behaviour of the emulated hardware, depending on the system, the screen output, the state of the emulated processor registers or a fingerprint of its main memory can be used to draw conclusions.

For our experiments we have found the approach of VNCPlay (Zeldovich and Chandra 2005) useful which produces and compares snapshots of a small area around the mouse cursor for synchronization. This method was originally developed for the platform independent latency evaluation of graphical user interfaces.

Moreover, in our setting we make use of an emulated environment which not only allows the observation of external events of the running machine (e.g. screen), but also internal states of the emulated machine can be observed. But most important is the ability to monitor and alter the state of peripheral devices in an automated way.

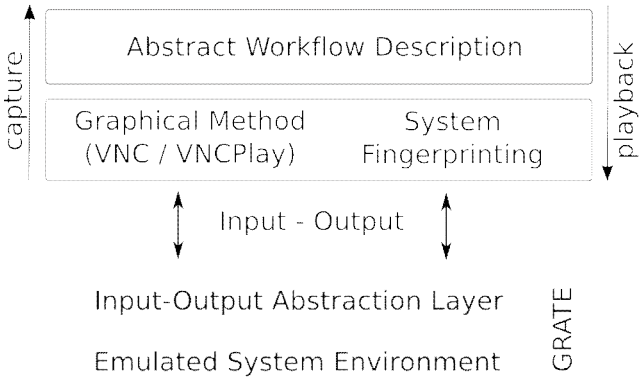

Figure 4: System setup for unattended interaction workflows.

Experiments and Example Workflows

Given a certain primary object an archivist creates or resolves a View-Path by choosing the closest match of existing pathways in the software archive. In the worst case the process starts with including the operation system. In the best case only the appropriate application has to be installed, if necessary at all. Every transition between two dependencies in the View-Path consists of recorded installation and configuration sessions within an emulated environment possibly with some auxiliary data sources attached.

For example for installing the Lotus AmiPro (Fig. 5) application a runtime environment is necessary, consisting of a supported operation system and the emulation of necessary hardware requirements. For this example QEMU the emulation of a 386 compatible PC running Windows 3.11 on basis of MS-DOS 6.20 with CD-ROM support was used. In order to produce PostScript documents an appropriate printer driver has to be installed additionally.

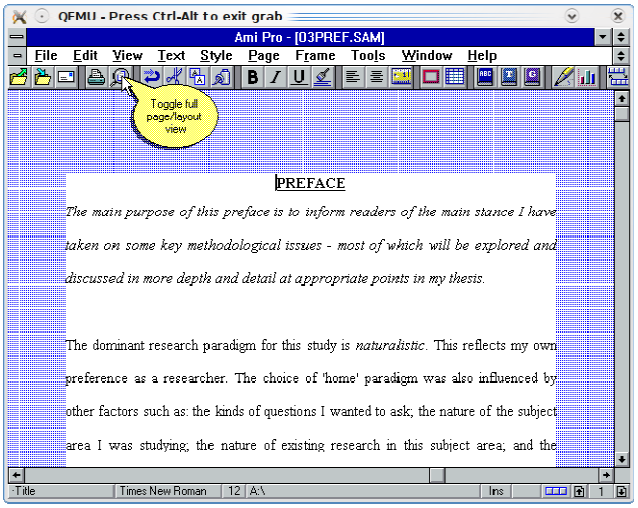

Figure 5: AMI Pro in an emulated Win3.11 environment.

We do not record the installation process of operating systems, since these runnable images serve as end-points of every single View-Path. Only transitions from these endpoints need to be captured. For this example we recorded the installation of a PostScript printer driver and the AmiPro application as two distinct steps. For each process we rebooted the emulator with an on-the-fly generated ISO-9660 disk-image containing all necessary tools and installation files.

The result of both recordings is the full view-path for the designated file format of the primary object and thus is ready for a view- or migration session. For both session types the primary object has to be prepared for transportation into the emulated environment. For this example we have created a floppy image-file containing the AmiPro (SAM) document.

For the view-session only a few automated actions might be recorded due to the simplicity of this example. For more complex applications preparing a ready-made view-session supporting users on older GUI environments may increase productivity. Migration-sessions are designed to run in an unattended manner. Once the session is recorded it should be replayable on any document within the given specifications. The recorded workflow in this example was opening a SAM file with AmiPro and printing it with a PS-enabled printer driver directly to the

attached floppy disk. We were able to successfully run a batch job doing a migration of several AmiPro documents serially. For each document the image with the full View-Path was freshly booted and a floppy image with the original file was created and the prerecorded workflow was executed. For each step the resulting PS-file was saved on the floppy and was available for further processing.

Results

We have evaluated the playback under different environments and conditions (cf. Tab. 1). For example, the replay described in scenario 1 was conducted under the same environment the recording took place. The elapsed time is comparable to the recording time. The small deviation is due to the extra time added after synchronization points to allow the operation system to be ready to handle the next input-event. In scenario 2 the playback environment was under heavy IO- and CPU-load and therefore the playback took three times longer than the recording. Not only the total time increased (denoted as real) but also the CPU-time accounted to the process (denoted as user) increased significantly. This is due to more screen-fingerprint comparisons, while waiting for the matching synchronization point.

	Scenario 1	**Scenario 2**
real	4 min 28.804 sec	14 min 03.495 sec
user	0 min 04.652 sec	3 min 53.219 sec
sys	0 min 01.420 sec	0 min 05.588 sec

Table 1: Playback within different environments. Numbers created by the UNIX time command. Recording took 4m 9.949 s

However, we have observed several failures while experimenting with different workflows and environmental settings. For example creating the above described View-Path in the opposite order (first installing AmiPro and then the PostScript driver) the playback of the installation procedure failed at the very last step. The screen's fingerprint differed more than the allowed threshold of 5% of mismatched pixels. This was mainly due to the change of the desktop background, as the icon of the previously installed printer application was missing. Other failures were due to bad click- or event-timing. But with some training and more experience we were able to create working recordings with high success rates. Additionally, some extra measures can be taken to further improve the reliability:

The environment must always entered and left in a predictable way. There are two possibilities entering a session. Either the environment is booted or some actions, like prerecorded tasks, have already been executed. The system should look the same at any time, even if the session starts on top of different but compatible View-Paths. Most importantly, all windows should be closed and if possible no background or minimized tasks should run. Whenever possible keyboard-shortcuts should be used. They are in particular helpful if new windows are opened at a random position. With the appropriate key-combination the window can be maximized and therefore the playback can always match the synchronization point. Furthermore, relying on a single screen-snapshot close to the mouse-pointer is not always sufficient. In contrast to the original purpose of the VNCPlay techniques, for creating reliable and sustaining interactive workflows, some extra effort can be imposed on the recording user. We are currently developing a toolbox to alter and improve recordings by defining extra synchronization points and altering the designated area for fingerprinting but also choosing fallback strategies e.g. if a mouse click was not recognized by the system properly.

Conclusion and Outlook

With the described techniques, tools and experiments we have shown that using the taken approach is a promising and suitable way constructing workflows for aggregating View-Paths, preparing view sessions and executing interactive migrations, all in an unattended way.

The advantages of automated wrapping of interactive environments are twofold: It is possible to run large batch jobs e.g. for migration tasks in an unattended mode. Also untrained people like average archive users of the future could get access to ancient environments without knowing how exactly they have to be handled. But nevertheless, emulation is a strategy with certain complexities requiring specially trained archivists: "New Skills Call for New Jobs". Aside from the emulated computer environment, documentation and skills are needed to understand how to operate an old computer environment. As of today many of us still remember older environments such as MS-DOS and early Windows versions, but soon even that information will disappear as much of the mainframe operation knowledge is already. Therefore, manuals, tutorials and practical how-to's need to be available as well. Taking the approach presented into account the emulation software archive needs to contain not only the emulators and all components for the emulated environments but has to be supplemented with a new section—the interactive workflows and its View-Path pendants.

However, for unattended interactive migration tasks some important challenges remain. Most importantly, such recorded workflows should be described on an abstract level and should therefore be editable. Hence, such abstract workflows can be treated as digital objects with possible migration paths and facilitate future compatibility.

Just as well, the description format should support branches to allow different behavior in different environments. If, for example, an action fails, in some

cases a rollback and/or retry from or to a defined checkpoint should be possible. Thus, a higher probability of successfully applying a prerecorded workflow in a slightly different environment could be achieved.

Usability and reliability can also be improved by providing appropriate tools. Editing recordings allows the user to improve recordings, alter or set synchronization points manually and to select a strategy for failed actions. Also, enriching recordings with descriptive meta-data will help future users understanding the recorded workflows.

Acknowledgements

Part of this work was supported by the European Union in the 6th Framework Program, IST, through the PLANETS project, contract 033789.

References

Becker, C.; Kulovits, H.; Kraxner, M.; Gottardi, R.; Rauber, A.; andWelte, R. 2009. Adding quality-awareness to evaluate migration web-services and remote emulation for digital preservation. In To appear in the Proceedings of the 13th European Conference on Digital Libraries (ECDL 2009).

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

Oltmans, E.; van Diessen, R.; and van Wijngaarden, H. 2004. Preservation functionality in a digital archive. In JCDL '04: Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries, 279–286. New York, NY, USA: ACM Press.

The National Archives. 2007. The technical registry pronom. WWW-Dokument, http://www.nationalarchives.gov.uk/pronom [Online; letzter Aufruf 10.01.2008].

van der Hoeven, J.; van Diessen, R.; and van der Meer, K. 2005. Development of a universal virtual computer (uvc) for long-term preservation of digital objects. Journal of Information Science 31(3):196–208.

von Suchodoletz, D., and van der Hoeven, J. 2008. Emulation: From digital artefact to remotely rendered environments. In iPRES 2008: Proceedings of the Fifth International Conference on Preservation of Digital Objects, 93–98. The British Library, St. Pancras, London: The British Library.

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

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