H. Neuroth, A. Oßwald, R. Scheffel, S. Strathmann, K. Huth (Hrsg.)

nestor Handbuch

Eine kleine Enzyklopädie der digitalen Langzeitarchivierung

Version 2.3

Kapitel 17.8

Interaktive digitale Objekte

nestor

nestor Handbuch: Eine kleine Enzyklopädie der digitalen Langzeitarchivierung
hg. v. H. Neuroth, A. Oßwald, R. Scheffel, S. Strathmann, K. Huth
im Rahmen des Projektes: nestor – Kompetenznetzwerk Langzeitarchivierung und Langzeitverfügbarkeit digitaler Ressourcen für Deutschland
nestor – Network of Expertise in Long-Term Storage of Digital Resources
http://www.langzeitarchivierung.de/

Kontakt: editors@langzeitarchivierung.de
c/o Niedersächsische Staats- und Universitätsbibliothek Göttingen,
Dr. Heike Neuroth, Forschung und Entwicklung, Papendiek 14, 37073 Göttingen

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet unter http://www.d-nb.de/ abrufbar.

Neben der Online Version 2.3 ist eine Printversion 2.0 beim Verlag Werner Hülsbusch, Boizenburg erschienen.

URL für Kapitel 17.8 „Interaktive digitale Objekte“ (Version 2.3):
urn:nbn:de:0008-20100617343
http://nbn-resolving.de/urn/resolver.pl?urn:nbn:de:0008-20100617343

Gewidmet der Erinnerung an Hans Liegmann (†), der als Mitinitiator und früherer Herausgeber des Handbuchs ganz wesentlich an dessen Entstehung beteiligt war.

17.8 Interaktive digitale Objekte

Dirk von Suchodoletz

Interaktive Applikationen sind spezielle digitale Objekte, die sich aufgrund ihres dynamischen, nichtlinearen Charakters schlecht mit traditionellen Archivierungsstrategien wie der Migration bewahren lassen. Sie treten dem Archivbetreiber typischerweise in zwei Ausprägungen entgegen. Entweder sie sind aus sich heraus von Bedeutung und primärem Interesse: In diese Klasse zählen Datenbankprogramme oder Computerspiele oder auch besondere technische Umgebungen, die als solche bewahrt werden sollen. Diese Objekte kann man auch als Primärobjekte fassen. Zusätzlich benötigt werden weitere dynamische Objekte, wie Applikationen zum Anzeigen oder Abspielen bestimmter Datenformate.
Sie werden als Hilfsmittel für die eigentlich interessierenden Objekte gebraucht und könnten daher als Sekundärobjekte bezeichnet werden. Allen dynamischen Objekten ist gemein, dass sie sich nur durch die Rekonstruktion ihrer Nutzungsumgebungen, einer bestimmten Zusammenstellung aus Software und / oder Hardware bewahren lassen. Diese Umgebungen lassen sich mithilfe der Emulationsstrategie langzeitbewahren. Oft genügt das interaktive Objekt alleine nicht: Je nach Primärobjekt oder Komplexität sind weitere Komponenten wie Schriftarten, Codecs oder Hilfsprogramme erforderlich, die in einem Softwarearchiv zusätzlich aufgehoben werden müssen.

Einführung

Mit der Durchdringung fast aller Bereiche des täglichen Lebens mit Computern änderten sich die Verfahren zur Speicherung, Verbreitung und Vervielfältigung von Informationen. Die elektronische Universalmaschine übernimmt eine dominierende Rolle: Eine zunehmende Zahl traditioneller Objekte wie Texte, Bilder, Musik und Film sind nicht mehr an analoge Medien gebunden, sondern können effizient digital bearbeitet, kopiert und verbreitet werden. Wissenschaftler fast aller Disziplinen erheben ihre Daten immer seltener ohne elektronische Maschinen, erfasste Informationen nehmen ohne Umweg über Papier und Stift den Weg zur Verarbeitung, Nutzung, Auswertung und Archivierung.

Die erzeugten digitalen Objekte können, anders als klassische Medien wie Papier oder Leinwände, nicht aus sich alleine heraus betrachtet werden. Die Erstellung und der Zugriff auf sie ist nur mithilfe eines nicht unerheblichen technischen Apparates möglich und gerade dieser Apparat unterliegt einer rasanten Fortentwicklung. Digitale Objekte erfordern eine bestimmte technische Umgebung, die sich aus Soft- und Hardwarekomponenten zusammensetzt. Diese Umgebung, hier als Erstellungs- oder Nutzungsumgebung bezeichnet, sorgt

dafür, dass ein Benutzer ein digitales Objekt je nach Typ betrachten, anhören oder ausführen kann.

Bei Computerprogrammen und Betriebssystemen, allgemein unter dem Begriff Software zusammengefasst, handelt es sich um dynamische, interaktive digitale Objekte. Vielfach wurden sie für die direkte Benutzerinteraktion mit dem Computer geschrieben, damit Menschen überhaupt erst sinnvoll Rechnerhardware nutzen können. Interaktive Objekte zeichnen sich durch einen nicht-linearen Aufbau und Ablauf aus. Erst die Interaktion mit dem Computeranwender bestimmt, wie sich das Objekt verhält. Jede Sitzung verläuft anders, deshalb ist die Zahl der möglichen Handlungswege typischerweise unbeschränkt.

Solche interaktiven Objekte treten dem Archivbetreiber und -nutzer in zwei Ausprägungen gegenüber. Einerseits handelt es sich um *Primärobjekte*. Diese sind als archivwürdig eingestufte Objekte, an denen ein direktes Interesse besteht. Hierzu zählen beispielsweise Computerspiele, interaktive Medien, Unterhaltung oder digitale Kunst, deren Archivierung in eigenen Abschnitten dargestellt wird. *Sekundärobjekte* meint alle digitalen Objekte, die zur Darstellung oder zum Ablaufenlassen von Primärobjekten erforderlich sind.[54]

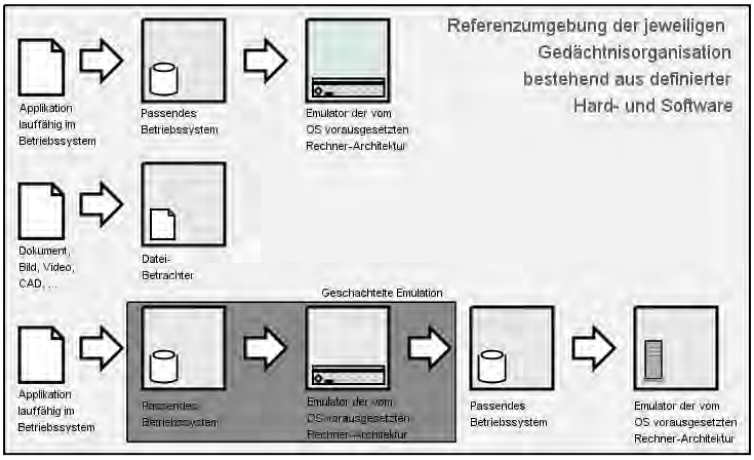

Abbildung 1: Je nach Art des Primärobjekts sind unterschiedliche Schritte zu seiner Darstellung oder Ausführung notwendig.

54 Sekundärobjekte werden je nach Vorhandensein bestimmter Primärobjekte nachgefragt und besitzen keine eigene Archivwürdigkeit aus sich heraus.

Diese Zusammenhänge lassen sich durch View-Paths (Kapitel 9.3) formalisieren. Darunter versteht man Darstellungswege, die vom darzustellenden oder auszuführenden digitalen Primärobjekt starten und bis in die tatsächliche Arbeitsumgebung des Archivnutzers reichen (Abbildung 1). Sie unterscheiden sich in ihrer Länge je nach Objekttyp und eingesetzter Archivierungsstrategie. Ein klassisches statisches Objekt wie ein Textdokument, Bild oder eine Videosequenz (linearer Ablauf ohne beliebige Verzweigungen) benötigen eine Darstellungsapplikation. Diese kann im Fall einer Migration direkt in der Arbeitsumgebung des Archivnutzers (Abbildung 1, Mitte) ablaufen. Damit fällt der View-Path kurz aus. Ist diese Applikation nicht mehr auf aktuellen Systemen installierbar und lauffähig, so sind weitergehende Maßnahmen erforderlich. Dann muss die Nutzungsumgebung auf geeignete Weise nachgebildet werden, was durch Emulation realisiert werden kann. Damit verlängert sich der View-Path und die Menge der benötigten Sekundärobjekte vervielfacht sich mindestens um den Emulator und ein zur Applikation passendes Betriebssystem.

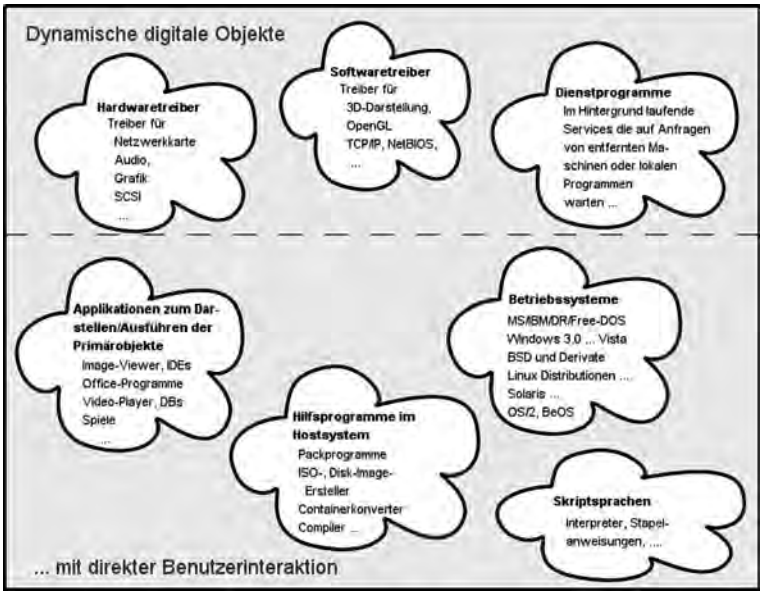

Abbildung 2: Die Klasse der dynamischen digitalen Objekte umfasst die interaktiven Objekte, die für eine direkte Benutzerinteraktion erstellt wurden. Ihre Einordnung muss nicht immer eindeutig ausfallen.

Typen interaktiver Objekte und ihre Bedeutung

Die Ausdifferenzierung und technologische Entwicklung auf dem Gebiet der elektronischen Datenverarbeitung hat inzwischen einen hohen Grad der Differenzierung erreicht, der sich in einer Reihe spezialisierter Komponenten niederschlägt. Erst das Zusammenspiel von ausführbaren Programmen, Betriebssystemen, Softwarebibliotheken und weiteren Komponenten (Abbildung 2) erlaubt die Erschaffung der typischen Objekte, mit denen sich die digitale Langzeitarchivierung befasst.

Die Liste wichtiger Sekundärobjekte umfasst:

- Betriebssysteme - sind die Grundkomponenten eines jeden Rechners neben seiner (physikalischen) Hardware (Abbildung 3). Sie übernehmen die Steuerung der Hardware, kümmern sich um Ressourcenverwaltung und -zuteilung und erlauben die Interaktion mit dem Endanwender. Sie sind im kompilierten Zustand[55] deshalb nur auf einer bestimmten Architektur ablauffähig und damit angepasst an bestimmte Prozessoren, die Art der Speicheraufteilung und bestimmte Peripheriegeräte zur Ein- und Ausgabe. Da eine Reihe von Funktionen von verschiedenen Programmen benötigt werden, sind diese oft in sogenannte Bibliotheken ausgelagert. Programme, die nicht alle Funktionen enthalten, laden benötigte Komponenten aus den Bibliotheken zur Laufzeit nach. Bibliotheken und Programme hängen dementsprechend eng miteinander zusammen.

- Anwendungsprogramme - oberhalb der Betriebssystemebene[56] befinden sich die Anwendungen (Abbildung 3). Dieses sind Programme, die für bestimmte, spezialisierte Aufgaben erstellt wurden. Mit diesen Programmen generierten und bearbeiteten Endanwender Daten der verschiedensten Formate. Der Programmcode wird im Kontext des Betriebssystems ausgeführt. Er kümmert sich um die Darstellung gegenüber dem Benutzer und legt fest, wie beispielsweise die Speicherung von

55 Computerprogramme werden in verschiedenen Programmiersprachen erstellt. Dieses sind typischerweise sogenannte Hochsprachen, die nicht direkt von einem Prozessor interpretiert werden können und erst in Maschinensprache übersetzt werden müssen. Während die Hochsprache relativ abstrakt von der konkreten Hardware ist, kann Maschinensprache immer nur auf einer bestimmten Computerarchitektur ausgeführt werden.

56 Für die schematische Darstellung der Arbeit eines Computers wird oft ein Schichtenmodell gewählt.

Objekten in einer Datei organisiert ist und wie der Anwender mit dem Computer interagiert. Das Betriebssystem übernimmt die Speicherung von Dateien auf Datenträgern üblicherweise angeordnet in Verzeichnissen. Zur Ausführung auf einer bestimmten Rechnerarchitektur werden Betriebssysteme und Applikationen aus dem sogenannten Quellcode in den passenden Binärcode übersetzt. Deshalb können Programme und Bibliotheken nicht beliebig zwischen verschiedenen Betriebssystemen verschoben werden.

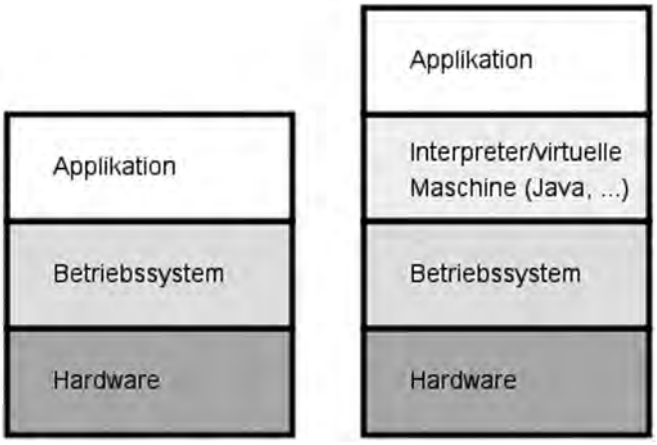

Abbildung 3: Eine typische Ablaufumgebung neuerer Rechnerarchitekturen für digitale Objekte bestehend aus Hard- und Software als Schichtenmodell. Interpreter und abstrakte Programmiersprachen wie Java können eine weitere Schicht einführen (rechts).

Ein wichtiges Beispiel digitaler Objekte primären Interesses sind Datenbanken. Die Bewegung, Durchsuchung und Verknüpfung großer Datenbestände gehört zu den großen Stärken von Computern. Zur Klasse der datenbankbasierten Anwendungen zählen Planungs- und Buchhaltungssysteme, wie SAP, elektronische Fahrpläne diverser Verkehrsträger bis hin zu Content Management Systemen (CMS) heutiger Internet-Auftritte von Firmen und Organisationen. Wenn von einer Datenbank sehr verschiedene Ansichten ad-hoc erzeugt werden können, ist sehr schwer abzusehen, welche dieser Ansichten zu einem späteren Zeitpunkt noch einmal benötigt werden könnten. Unter Umständen hat man sich dann auf Teilmengen festgelegt, die von nachfolgenden Betrachtern als unzureichend oder irrelevant eingestuft werden könnten. Gerade bei Datensammlungen wichtiger langlebiger Erzeugnisse wie Flugzeugen besteht großes allgemeines Interesse eines zeitlich unbeschränkten Zugriffs.

Alle genannten dynamischen Objekttypen zeichnen sich dadurch aus, dass sie außerhalb ihres festgelegten digitalen Kontextes heraus nicht sinnvoll interpretiert und genutzt werden können. Zudem ist ihre Migration nicht trivial:[57] Es fehlt typischerweise die gesamte Entwicklungsumgebung und der Zugriff auf den Quellcode, die eine Anpassung und Übersetzung auf aktuelle Systeme erlauben würden. Zudem wäre der personelle Aufwand immens und stünde häufig nicht in einem sinnvollen Verhältnis zum Wert des Objekts. Ebenso scheidet eine Überführung in ein analoges Medium in den meisten Fällen aus: Der Ausdruck des Programmcodes auf Papier oder Mikrofilm oder die Aufnahme einer Programmsitzung auf Video sind derart „verlustbehaftete" Speicherverfahren, dass sie im Sinne der Archivierung vielfach die gestellten Anforderungen nicht erfüllen.

Emulation – Erhalt von Nutzungsumgebungen

Emulation heißt zuerst einfach erstmal nur die Schaffung einer virtuellen Umgebung in einer gegebenen Nutzungsumgebung, üblicherweise dem zum Zeitpunkt des Aufrufs üblichen Computersystem. Das kann bedeuten, dass Software durch eine andere Software nachgebildet wird, ebenso wie Hardware in Software.

Emulation setzt dabei nicht am digitalen Objekt selbst an, sondern beschäftigt sich mit der Umgebung, die zur Erstellung dieses Objektes vorlag. Das bedeutet beispielsweise die Nachbildung von Software durch andere Software, so dass es für ein betrachtetes digitales Objekt im besten Fall keinen Unterschied macht, ob es durch die emulierte oder durch die Originalumgebung behandelt wird. Dem Prinzip folgend kann Computerhardware durch Software nachgebildet werden, auch wenn dieses erstmal deutlich komplexer erscheint. Einen ausführlichen Überblick zur Emulation als Langzeitarchivierungsstrategie gibt Kapitel 8.4, das entsprechende Kapitel in Borghoff (2003) oder auch Holdsworth (2001). Das Grundlagenwerk zur Emulation in der Langzeitarchivierung stammt von Jeff Rothenberg (1999) und (2000). Eine erste praktische Einbindung von Emulationsansätzen erfolgt derzeit im EU-geförderten PLANETS Project.[58]

---

57 Maschinencode kann nicht trivial von einer Rechnerarchitektur auf eine andere übersetzt werden so wie dieses für wohldefinierte Datenformate statischer Objekte möglich ist.

58 Es wird unter dem „Information Society Technologies (IST) Programme" des Framework 6 anteilig finanziert (Project IST-033789) und beschäftigt sich mit der prototypischen Erstellung von Tools

Emulatoren sind spezielle Software-Applikationen, die in der Lage sind Nutzungsumgebungen, wie Hardware-Plattformen, in derart geeigneter Weise in Software nachzubilden, dass ursprünglich für diese Nutzungsumgebung erstellte Applikationen weitgehend so arbeiten wie auf der Originalplattform. Emulatoren bilden damit die Schnittstelle, eine Art Brückenfunktion, zwischen dem jeweils aktuellen Stand der Technik und einer längst nicht mehr verfügbaren Technologie. Dabei müssen sich Emulatoren um die geeignete Umsetzung der Ein- und Ausgabesteuerung und der Peripherienachbildung bemühen.

Die Hardwareemulation setzt auf einer weit unten liegenden Schicht an (Abbildung 3). Das bedeutet auf der einen Seite zwar einen sehr allgemeinen Ansatz, erfordert umgekehrt jedoch eine ganze Reihe weiterer Komponenten: Um ein gegebenes statisches digitales Objekt tatsächlich betrachten zu können oder ein dynamisches Objekt ablaufen zu sehen, müssen je nach Architektur die Ebenen zwischen der emulierten Hardware und dem Objekt selbst „überbrückt" werden. So kann ein Betrachter nicht auf einer nackten X86-Maschine ein PDF-Dokument öffnen (Abbildung 4). Er braucht hierfür mindestens ein Programm zur Betrachtung, welches seinerseits nicht direkt auf der Hardware ausgeführt wird und deren Schnittstellen direkt programmiert. Dieses Programm setzt seinerseits ein Betriebssystem als intermediär voraus, welches sich um die Ansteuerung der Ein- und Ausgabeschnittstellen der Hardware kümmert.

Die Auswahl der inzwischen kommerziell erhältlichen oder als Open-Source-Software verfügbaren Emulatoren oder Virtualisierer (Abbildung 4) ist inzwischen recht umfangreich geworden, so dass häufig sogar mehr als ein Emulator für eine bestimmte Rechnerarchitektur zur Verfügung steht. Der überwiegende Anteil von Emulatoren und Virtualisierern wurde oftmals aus ganz anderen als Langzeitarchivierungsgründen erstellt. Sie sind heutzutage Standardwerkzeuge in der Software-Entwicklung. Nichtsdestotrotz eignen sich viele der im folgenden vorgestellten Werkzeuge für eine Teilmenge möglicher Langzeitarchivierungsaufgaben. Institutionen und privaten Nutzern reichen in vielen Fällen derzeitig verfügbare Programme aus. Jedoch eignet sich nicht jeder Emulator gleichermaßen für die Zwecke des Langzeitzugriffs, weil sich die nachgebildete Hardware ebenso wie die reale weiterentwickelt. Wird alte Hardware nicht mehr unterstützt, kann es passieren, dass ein bestimmtes Betriebssystem nicht mehr auf das Netzwerk zugreifen oder Audio abspielen kann.

Das Schichtenmodell (Abbildung 3) lässt sich nicht auf alle Rechnerarchitekturen anwenden. So existiert für frühe Architekturen keine deutliche Unterscheidung zwischen Betriebssystem und Applikation. Frühe Modelle von Home-Computern verfügten über eine jeweils recht fest definierte Hardware,

die zusammen mit einer Art Firmware ausgeliefert wurde. Diese Firmware enthält typischerweise eine einfache Kommandozeile und einen Basic-Interpreter. Nicht alle für den Betrieb von Emulatoren benötigten Komponenten, wie beispielsweise die genannte Home-Computer-Firmware ist frei verfügbar und muss deshalb mit geeigneten Rechten abgesichert sein. Ohne die Archivierung dieser Bestandteile ist ein Emulator für die Plattform wertlos und das Ausführen entsprechender archivierter Primärobjekte unmöglich.

Abbildung 4: Der VMware-Server 2 ist ein X86 Virtualisierer, der einen Zugriff auf alte Nutzungsumgebungen, wie Windows98 über das Netz erlaubt.

Softwarearchiv

Nach den eher theoretisch angelegten Vorbetrachtungen zum Ansatzpunkt der Emulation und erforderlicher Zusatzkomponenten steht nun das Softwarearchiv als ein zentrales Hilfsmittel der Emulationsstrategie im Mittelpunkt. Zu den Erfolgsbedingungen für den Erhalt möglichst originalgetreuer Nutzungsumgebungen für die verschiedensten Typen digitaler Objekte zählen nicht nur

die Primärwerkzeuge – die Emulatoren. Das Archiv muss eine ganze Reihe verschiedener Softwarekomponenten umfassen:

- Geeignete Emulatoren sind zu speichern, so dass mit ihrer Hilfe die Wiederherstellung einer Rechnerarchitektur für bestimmte Nutzungsumgebungen erfolgen kann. Hierzu kann es nötig sein Firmware-Komponenten,[59] wie Home-Computer-ROMs oder X86-BIOS, ebenfalls zu speichern.
- Betriebssysteme, die je nach Rechnerplattform einen Teil der Nutzungsumgebung ausmachen, sind im Softwarearchiv abzulegen.
- Treiber der Betriebssysteme müssen zusätzlich gespeichert werden, da sie den Betriebssystemen erst ermöglichen mit einer bestimmten Hardware umzugehen.
- Alle Applikationen, mit denen die verschiedenen digitalen Objekte erstellt wurden, sind zu archivieren. Diese Anwendungsprogramme sind ebenfalls Bestandteil der Nutzungsumgebung des Objektes. Sie sind in vielen Fällen auf die vorgenannten Betriebssysteme angewiesen.
- Unter Umständen notwendige Erweiterungen einer Applikationsumgebung, wie bestimmte Funktionsbibliotheken, Codecs[60] oder Schriftartenpakete zur Darstellung, sind aufzubewahren.
- Hilfsprogramme, welche den Betrieb der Emulatoren vereinfachen oder überhaupt erst ermöglichen, sind zu sammeln. Hierzu zählen beispielsweise Programme, die direkt mit dem jeweiligen Containerformat eines Emulators umgehen können.
- Je nach Primärobjekt oder gewünschter Nutzungsumgebung sind mehrere Varianten derselben Software zu archivieren, um beispielsweise die Anpassung an den deutschen oder englischsprachigen Raum zu erreichen. Das betrifft einerseits die Verfügbarkeit verschiedensprachiger Menüs in den Applikationen aber auch geeignete Schriftarten für die Darstellung von Sonderzeichen oder Umlauten.

59 Basissoftware, die direkt mit der Hardware verknüpft vom Hersteller ausgeliefert wird.

60 Codecs sind in Software gegossene Verfahren zur Digitalisierung und Komprimierung analoger Medien, wie Audio, Filme.

Mittels View-Paths lässt sich der Vorgang zur Bestimmung der benötigten Softwarekomponenten formalisieren. Ausgehend von den Metadaten des Primärobjekts über die Metadaten der benötigten Applikation zur Betrachtung oder zum Abspielen bis hin zu den Metadaten des Betriebssystems werden die Komponenten ermittelt. Hierzu müssten bereits bestehende Format-Registries, wie beispielsweise PRONOM[61] erweitert werden.

Datenaustausch von Objekten

Ein weiteres nicht zu unterschätzendes Problem liegt im Datentransport zwischen der im Emulator ablaufenden Software und der Software auf dem Computersystem des Archivnutzers. Diese Fragestellung unterscheidet sich nicht wesentlich vom Problem des Datenaustauschs zwischen verschiedenen Rechnern und Plattformen. Mit fortschreitender technischer Entwicklung ergibt sich unter Umständen ein größer werdender Spalt zwischen dem technologischen Stand des stehenbleibenden emulierten Systems und dem des Host-Systems , das die Emulation ausführt. Zum Teil halten die verfügbaren Emulatoren bereits Werkzeuge oder Konzepte vor, um die Brücke zu schlagen.

Nach der Rekonstruktion einer bestimmten Nutzungsumgebung möchte man in dieser die gewünschten Daten ansehen, ablaufen lassen oder in selteneren Fällen bearbeiten. In der Zwischenzeit haben sich mit einiger Wahrscheinlichkeit die Konzepte des Datenaustausches verändert. Hier ist nun dafür zu sorgen, dass die interessierenden Objekte geeignet in die (emulierte) Nutzungsumgebung gebracht werden können, dass die Betrachtung für den Archivnutzer in sinnvoller Form möglich ist und dass eventuell Bearbeitungsergebnisse aus der Nutzungsumgebung in die aktuelle Umgebung transportiert werden können. Vielfach wird sich je nach Erstellungsdatum des Objektes die damalige Erstellungs- oder Nutzungsumgebung dramatisch von der jeweils aktuellen unterscheiden.

Diese Unterschiede überbrückt der Emulator, indem er statt mit physischen Komponenten mit virtuellen arbeitet. Während früher ein Computerspiel von einer Datasette[62] oder eine Datenbank von einer 8“ Diskette geladen wurde, stehen diese Varianten nicht mehr zur Verfügung. Die Daten sind im Augenblick der Archivaufnahme in Abbilder der früheren Medien umgewandelt worden, die dem logischen Aufbau des originalen Datenträgers entsprechen. Die

61 „The technical registry PRONOM“, http://www.nationalarchives.gov.uk/pronom des britischen Nationalarchivs. Diese Formatregistratur kennt eine große Zahl verschiedener Datenformate. Gleichzeitig steht mit DROID ein Programm zur Ermittlung bereit.

62 Spezielles Compact-Kassenten-Gerät für die Datenspeicherung von Home-Computern, wie dem C64.

virtuellen Datenträger können sodann vom Emulator gelesen und dem nachgebildeten System wie die originalen angeboten werden.

In vielen Fällen liegen die interessierenden Primärdaten als Dateien im Host-System und noch nicht auf einem passenden virtuellen Datenträger vor. Da Emulatoren selten direkt auf Dateien im Host-System zugreifen können, müssen geeignete Wege geschaffen werden. Typischerweise lesen Computer Daten von einem Peripheriegerät wie Festplatten, optischen oder Diskettenlaufwerken. Emulierte Maschinen verwenden hierzu virtuelle Hardware. Über diese muss für einen geeigneten Transport von digitalen Objekten gesorgt werden, da der Benutzer diese normalerweise über das Host-System in die gewünschte Nutzungsumgebung einbringen wird. In einigen Fällen wird auch der umgekehrte Weg benötigt: Der Transport von Primärobjekten aus ihrer Nutzungsumgebung in die jeweils gültige Referenzumgebung.

Generell stehen eine Reihe von Varianten zur Verfügung (Abbildung 5), die jedoch von der jeweiligen Referenzplattform und ihrer technischen Ausstattung abhängen.

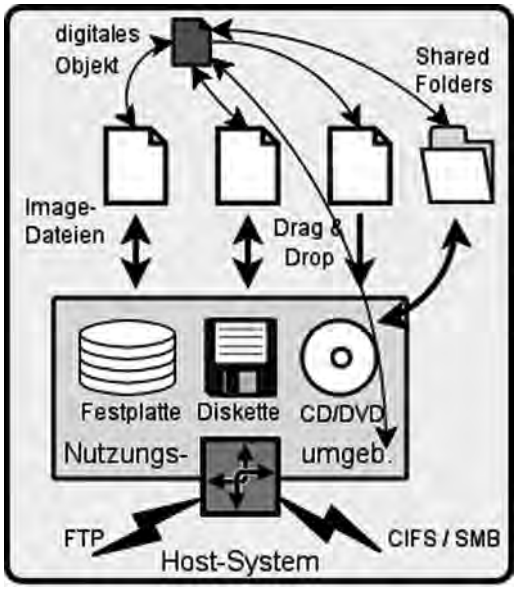

Abbildung 5: Digitale Objekte können auf verschiedene Weise zwischen der im Emulator laufenden Nutzungsumgebung und dem Host-System ausgetauscht werden.

Es lassen sich drei grundlegende Klassen von Austauschverfahren unterscheiden:

- Der klassische Weg, ohne Eingriffe in die Nutzungsumgebung und daher am universellsten einsetzbar, ist wohl der Einsatz von Datenträgerabbildern (auch als Disk-Images oder Containerdateien bezeichnet). Bei diesen handelt es sich um spezielle Dateien, welche die komplette Struktur und den vollständigen Inhalt von Datenspeichern der verschiedensten Art, wie Disketten, CD-ROMs, DVDs oder Festplatten enthalten. Der wesentliche Vorteil von Images ist der Erhalt der logischen Struktur des vormaligen Datenträgers bei Überführung in eine gut archivierbare Repräsentation.
- Im Laufe der technologischen Entwicklung begannen sich Computernetzwerke durchzusetzen. Betriebssysteme, die solche unterstützen, erlauben mittels virtueller Netzwerkverbindungen digitale Objekte zwischen Referenz- und Nutzungsumgebung auszutauschen.
- Neben diesen Verfahren bieten einige Virtualisierer und Emulatoren spezielle Wege des Datenaustauschs wie sogenannte „Shared Folders". Dieses sind spezielle Verzeichnisse im Host-System, die direkt in bestimmte Nutzungsumgebung eingeblendet werden können. Eine weitere Alternative besteht in der Nutzung der Zwischenablage[63] zum Datenaustausch.

Die notwendigen Arbeitsabläufe zum Datenaustausch können vom Erfahrungshorizont der jeweiligen Nutzer abweichen und sind deshalb auf geeignete Weise (in den Metadaten) zu dokumentieren oder zu automatisieren.

Disketten-, ISO und Festplatten-Images

Diskettenlaufwerke gehören zur Gruppe der ältesten Datenträger populärer Computersysteme seit den 1970er Jahren. Der überwiegende Anteil der Emulatoren und Virtualisierer ist geeignet, mit virtuellen Diskettenlaufwerken unterschiedlicher Kapazitäten umzugehen. Die virtuellen Laufwerke entsprechen, wie virtuelle Festplatten auch, einer Datei eines bestimmten Typs im Host-System. Das Dateiformat der virtuellen Disketten ist beispielsweise für alle virtuellen X86er identisch, so dass nicht für jeden Emulator ein eigenes Image ab-

63 Dieses entspricht einem Ausschneiden-Einfügen zwischen Host- und Nutzungsumgebung. Diese Möglichkeit muss vom Emulator unterstützt werden, da er die Brückenfunktion übernimmt. Die Art der austauschbaren (Teil-)Objekte hängen dabei vom Emulator und beiden Umgebungen ab.

gelegt werden muss und ein einfacher Austausch erfolgen kann. Die dahinter stehende einfache Technik sollte es zudem erlauben, sie auch in fernerer Zukunft zu emulieren.
Die physikalische Struktur der Diskette wird durch eine logische Blockstruktur in einer Datei ersetzt (Abbildung 6), indem die Steuerhardware des Laufwerkes und der physische Datenträger weggelassen werden. Aus Sicht des Betriebssystems im Emulator ändert sich nichts. Diskettenzugriffe werden einfach durch Zugriffe auf die Abbilddatei im Host-System umgesetzt.

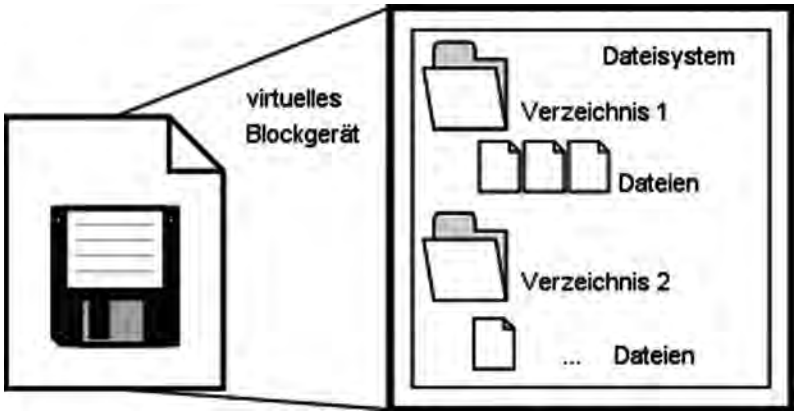

Abbildung 6: Der logische Aufbau einer Datei, die ein Disketten-Image repräsentiert. Die innere Blockstruktur kann mit einem Dateisystem versehen sein, welches Verzeichnisse und Dateien enthält.

Wegen des einfachen Aufbaus einer solchen Datei gibt es in den meisten Host-Systemen die Möglichkeit den Inhalt als virtuelles Blockgerät mit Dateisystem einzuhängen. Dadurch kann der Container so einfach wie eine normale Festplatte gelesen und beschrieben werden, wenn das Host-System das Dateisystem im Container unterstützt. Beispielsweise gelingt das Einbinden eines DOS-, VFAT- oder HFS-formatierten Disketten-Images auf Linux-Hosts ohne Schwierigkeiten, solange diese Dateisysteme und das spezielle Loop Blockdevice[64] vom Kernel[65] unterstützt werden. In Zukunft könnten langzeitrelevante

64 Das sogenannte Loop Device erlaubt Dateien innerhalb eines Dateisystems dem Betriebssystem als virtuellen Datenträger anzubieten. Auf diese Weise wird es möglich auf einzelne Komponenten innerhalb einer Containerdatei zuzugreifen.

65 Oder Kern, ist das eigentliche Betriebssystem, die zentrale Software zur Hardware- und Prozesssteuerung.

Dateisysteme mittels FUSE[66] realisiert und gepflegt werden. Für den Zugriff auf Disketten-Images in einer Referenzumgebung sind unter Umständen spezielle Hilfsprogramme notwendig, die zusätzlich archiviert werden sollten.

Disketten-Images können entweder beim Start des Emulators bereits verbunden sein. Fast alle Emulatoren unterstützen zudem das Ein- und Aushängen zur Laufzeit, welches normalerweise vom Benutzer getriggert[67] werden muss. Dieser sollte zudem darauf achten, dass eine Image-Datei nicht mehrfach eingebunden ist, da innerhalb des Containers Blockoperationen ausgeführt werden und kein Schutz vor konkurrierenden und potenziell zerstörenden Zugriffen stattfindet.

Hardwareemulatoren von Rechnerarchitekturen, die mit CD- oder DVD-Laufwerken umgehen können, verfügen über geeignete Implementierungen, um dem Gastsystem angeschlossene optische Wechseldatenträger anbieten zu können. Bei CDs und DVDs handelt es sich wie bei Disketten und Festplatten um blockorientierte Datenträger. Es besteht eine wesentliche Beschränkung: Der Datenaustausch kann nur in eine Richtung erfolgen, da einmal erstellte Datenträger ein typischerweise nur lesbares Dateisystem (ISO 9660 mit eventuellen Erweiterungen) enthalten.

CD-ROMs als Datenträger für den Einsatz im Computer gibt es erst seit Mitte der 1990er Jahre. Während Disketten quasi nativ in den meisten Plattformen unterstützt sind, muss für die Verwendung von CD-ROMs üblicherweise noch ein Treiber geladen werden, der ebenfalls im Softwareachiv abgelegt sein sollte.

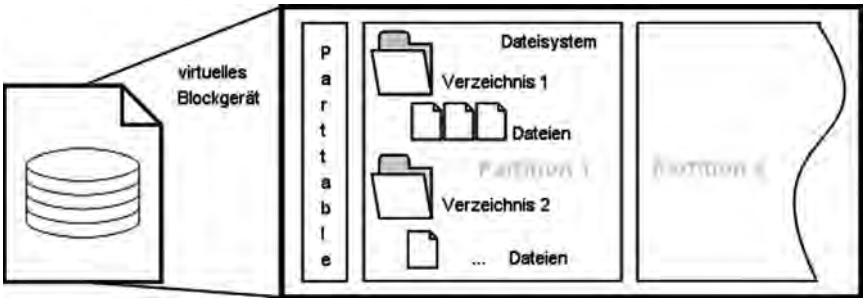

Abbildung 7: Der logische Aufbau einer Datei, die ein Festplatten-Image enthält. Dieses kann durch eine Partitionstabelle und mehrere Partitionen mit eventuell unterschiedlichen Dateisystemen strukturiert sein.

66 Filesystem im Userspace sind Dateisystemtreiber, die nicht im Betriebssystemkern implementiert sind.

67 Dieses entspricht dem traditionellen Einlegen und Auswerfen von Disketten mittels Softwarefunktion des Emulators.

Genauso wie bei Disketten, optischen Datenträgern wie CD-ROM, DVD oder Blu-ray-Disk, handelt es sich bei Festplatten um standardisierte blockorientierte Geräte. Dabei sorgen gerätespezifische Firmware und die jeweilige Hardwareplattform dafür, dass ein Betriebssystem in abstrakter Form auf das Laufwerk zugreifen kann. Spezielle Treiber, wie für die Einbindung optischer Wechseldatenträger, sind primär nicht notwendig.

Seit ungefähr 30 Jahren haben sich blockorientierte Festspeichergeräte als dauerhafter Speicher durchgesetzt. Der Inhalt dieser Speichermedien wird als virtuelles Blockgerät in Form von Dateien im Host-System repräsentiert. Ist der Aufbau dieser Dateien bekannt, so können diese ohne laufenden Emulator erzeugt und verändert werden.

Gegenüber Disketten ist bei Festplatten eine Partitionierung, eine logische Aufteilung der Gesamtkapazität, durchaus üblich.
Wegen des komplexeren inneren Aufbaus von Containerdateien wird ein einfaches direktes Einbinden im Referenzsystem fehlschlagen, wie es für Disketten- und ISO-Images vorgenommen werden kann. Stattdessen braucht man spezielle Programme, die mit Partitionen geeignet umgeben können.

Netzwerke zum Datenaustausch

Neuere Betriebssysteme verfügen über die Fähigkeit, Daten über Netzwerke auszutauschen. Viele Emulatoren enthalten deshalb virtuelle Netzwerkhardware und können zwischen Host- und Nutzungsumgebung virtuelle Datenverbindungen einrichten. Während Ende der 1980er Jahre noch proprietäre Protokolle wie Appletalk, NetBIOS und IPX/SPX neben TCP/IP existierten, verdrängte letzteres inzwischen weitestgehend die anderen. Hierbei ist bemerkenswert, dass der jetzige IPv4-Standard, welcher in seiner ersten Fassung am Ende der 1970er Jahre festgelegt wurde, im Laufe der Zeit erstaunlich stabil geblieben ist. Gleichzeitig offenbart sich der Vorteil freier, offener und damit langlebiger Standards.

Für den Austausch von digitalen Objekten bieten sich in erster Linie Protokolle an, die bereits seit vielen Jahren standardisiert sind. An prominenter Stelle stehen das seit den 1980er Jahren definierte File Transfer Protocol (FTP) und das Server Message Block Protocol (SMB). Handelt es sich bei diesen Softwarekomponenten nicht bereits um einen Teil des Betriebssystems, müssen sie bei Bedarf separat im Softwarearchiv abgelegt werden.

Bewahrung interaktiver, dynamischer Objekte

Unabhängig vom Charakter des Objekts, ob Primär- oder Sekundärobjekt, muss seine Nutzungsumgebung wiederherstellbar sein und damit kommt nur die Emulation als Strategie in Betracht. Eine zentrale Erfolgsbedingung für den Einsatz von Emulationsstrategien besteht deshalb im Aufbau und Betriebs eines Softwarearchivs. Je nach gewähltem Ansatzpunkt der Emulation wird eine Reihe zusätzlicher Softwarekomponenten benötigt (Abbildung 8)[68].

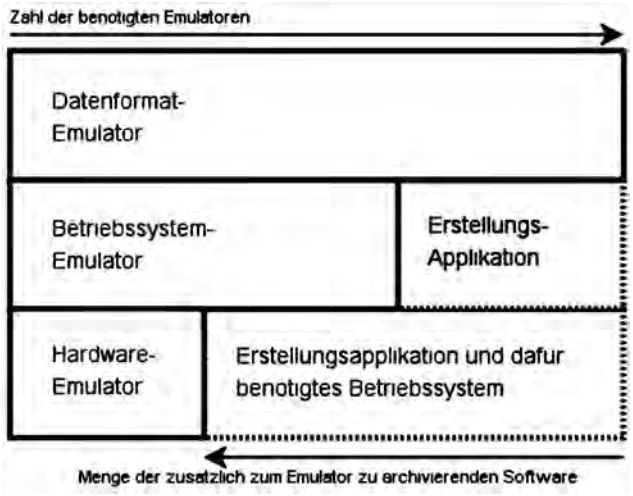

Abbildung 8: Je nach Ansatzpunkt der Emulation können eine Reihe zusätzlicher Sekundärobjekte benötigt werden.

Im Idealfall laufen die emulierten Applikationen[69] auf einer aktuellen Plattform und erlauben aus dieser heraus den direkten Zugriff auf die digitalen Objekte des entsprechenden Formates (Abbildung 1, Mitte). Solange es gelingt die entsprechende Applikation bei Plattformwechseln zu migrieren oder bei Bedarf neu zu erstellen, ist dieser Weg für die Langzeitarchivierung bestimmter Dateitypen durchaus attraktiv. Einsetzbar wäre dieses Verfahren durchaus für statische Dateitypen wie die verschiedenen offenen und dabei gut dokumentierten Bildformate.

68 Reichherzer (2006)

69 Beispielsweise ist OpenOffice ein Emulator für diverse Microsoft-Office Formate.

Die Emulation eines Betriebssystems oder dessen Schnittstellen erlaubt theoretisch alle Applikationen für dieses Betriebssystem ablaufen zu lassen. Dann müssen neben dem Emulator für das entsprechende Betriebssystem auch sämtliche auszuführende Applikationen in einem Softwarearchiv aufbewahrt werden (Abbildung 9). Bei der Portierung des Betriebssystememulators muss darauf geachtet werden, dass sämtliche in einem Softwarearchiv eingestellten Applikationen weiterhin ausgeführt werden können.

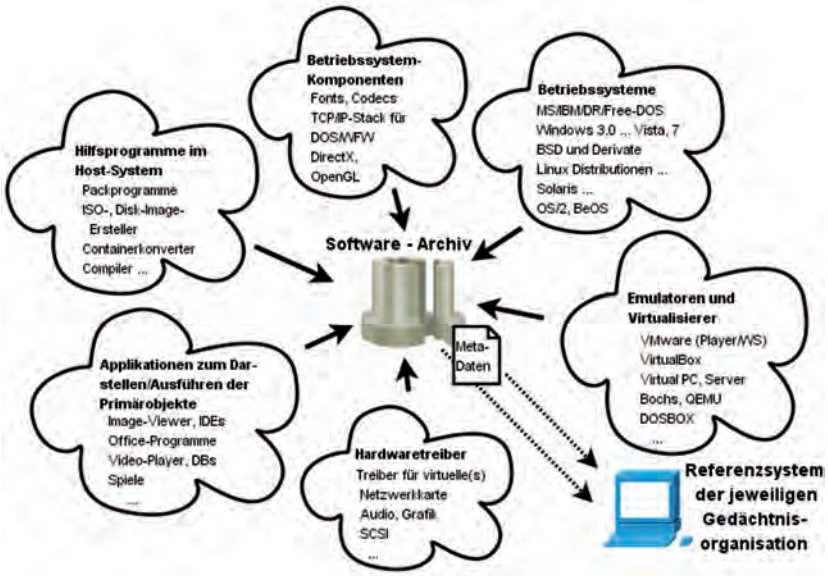

Abbildung 9: Auswahl der in einem Softwarearchiv für die X86-Architektur zu berücksichtigenden Sekundärobjekte.

Die Nachbildung einer kompletten Hardware in Software verspricht die besten Ergebnisse und verfolgt den allgemeinsten Ansatz. Dann benötigt man mindestens eines oder je nach darauf aufsetzenden Applikationen mehrere Betriebssysteme. Das bedeutet für ein Softwarearchiv, dass neben dem Emulator für eine Plattform auch die darauf ablauffähigen Betriebssysteme aufbewahrt werden müssen. Ebenso gilt dieses für die Liste der Applikationen, die zur Darstellung der verschiedenen archivierten Datentypen erforderlich sind. Erfolgt eine Portierung, also Migration des Hardwareemulators, muss anschließend überprüft werden, dass die gewünschten Betriebssysteme weiterhin ablauffähig bleiben. Da die meisten Applikationen typischerweise nicht direkt die Hardware

nutzen, sondern dafür auf die Schnittstellen der Betriebssysteme zugreifen, sollte deren Funktionsfähigkeit direkt aus der der Betriebssysteme folgen.

Virtualisierer entwickeln sich wie die nachgebildete Hardware weiter, um den technischen Entwicklungen und Marktbedürfnissen zu folgen. Damit kann es passieren, dass auch die nachgebildete Hardware für ein altes Betriebssystem zu neu ist. Dieses Problem löst beispielsweise VMware derzeit noch durch die Pflege und Bereitstellung geeigneter Treiber für jedes offiziell unterstützte Betriebssystem. Derzeit ist die Liste noch sehr lang und im Sinne der Archivierung fast vollständig. In Zukunft könnten jedoch am Ende des Feldes jedoch Betriebssysteme schrittweise herausfallen. Ein weiteres Problem von Virtualisierern ist die Art der Prozessornutzung. Viele virtuelle Maschinen reichen CPU-Befehle des Gastes direkt an die Host-CPU weiter (Beispiel VMware, Abbildung 4). Das setzt voraus, dass diese mit diesen Befehlen umgehen kann.

Geeigneter im Sinne einer wirklichen Langzeitarchivierung sind quelloffene Implementierungen.[70] Sie erlauben zum einen die Übersetzung für die jeweilige Plattform und zum anderen auch langfristige Anpassungen an völlig neue Architekturen. Zudem kann sichergestellt werden, dass auch alte Peripherie dauerhaft virtualisiert wird und nicht einem Produktzyklus zum Opfer fällt.

Ein weiterer Punkt ist die unter Umständen notwendige Anpassung der Containerdateien, in denen die Gastsysteme installiert sind. Ändert der Emulator das Datenformat, sind diese Dateien genauso wie andere digitale Objekte in ihrer Les- und Interpretierbarkeit gefährdet. Üblicherweise stellen jedoch die kommerziellen Anbieter Importfunktionen für Vorgängerversionen zur Verfügung. Bei freien, quelloffenen Emulatoren kann alternativ zur Weiterentwicklung dafür gesorgt werden, dass ein bestimmtes Dateiformat eingefroren wird. Ändert sich das Containerformat eines Emulators oder Virtualisierers müssen alle virtuellen Festplatten im Laufe der Zeit migriert werden, da sonst zu einem bestimmten Zeitpunkt keine Appliktion mehr existiert, die mit diesem umgehen kann. Damit wiederholt sich das Problem der Langzeitarchivierung – enthaltene Objekte sind nicht mehr zugreifbar und damit verloren.

---

70 Open Source Emulatoren können jederzeit von jedem Anwender mit geeigneten Programmierkenntnissen angepasst und damit für neue Plattformen übersetzt werden. Steht der Quellcode nicht zur Verfügung ist man der Produktstrategie des jeweiligen Unternehmens ausgeliefert, die häufig in deutlich kürzen Zyklen als denen der Langzeitarchivierung denkt.

Literatur

Borghoff, Uwe M. et al. (2003): *Langzeitarchivierung: Methoden zur Erhaltung digitaler Dokumente*. Heidelberg, dpunkt.verlag. ISBN 3-89864-245-3

Holdsworth, David / Wheatley, Paul (2001): *Emulation, Preservation and Abstraction*. In: RLG DigiNews, Nr. 4. Vol. 5

Rothenberg, Jeff (2000): *An Experiment in Using Emulation to Preserve Digital Publications*. http://nedlib.kb.nl/results/emulationpreservationreport.pdf

Rothenberg, Jeff (1999): *Avoiding Technological Quicksand: Finding a Viable Technical Foundation for Digital Preservation*. In: The State of Digital Preservation: An International Perspective. Conference Proceedings, Documentation Abstracts, Inc., Institutes for Information Science, Washington, D.C., April 24-25

Reichherzer, Thomas / Brown, Geoffrey (2006): *Quantifying software requirements for supporting archived office documents using emulation*. In: JCDL '06: Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries