

Dirk von Suchodoletz



## Die Emulationsstrategie in der Langzeitarchivierung: Vom digitalen Artefakt zu seiner Darstellung

*Emulation ist eine in der digitalen Langzeitarchivierung zunehmend ernst genommene Strategie. Sie kann die bisher hier dominierende Migration insbesondere im Umgang mit dynamischen Objekten ergänzen. Jedoch bedarf es der Beachtung einer Reihe von Bedingungen einer später erfolgreichen Anwendung in der Objektwiederherstellung. Die Formalisierung durch View-Path zur Ermittlung notwendiger Metadaten und zusätzlicher Softwarekomponenten schafft Vorschläge für mögliche Erweiterungen des OAIS-Referenzmodells.*

Emulation strategy in long term digital archiving. From the digital artefact to its presentation

*Emulation is a strategy getting more noticed in the long-term digital archiving community. It can complement the up to now dominant migration strategy and has significant advantages regarding dynamic digital objects. Nevertheless a range of success conditions are to be met for a successful recreation of objects in the future. The formalization of it using view path to identify necessary metadata and additional software components may propose extensions of the OAIS reference model.*

La stratégie d'émulation pour l'archivage à long terme. De l'artefact digital à son présentation

*L'émulation est une technique qui prend une importance grandissante dans le domaine de l'archivage à long terme des données électroniques. Elle permet de compléter les techniques actuelles et très répandues de migration en particulier en ce qui concerne les objets digitaux dynamiques. Toutefois il convient de respecter tout un ensemble de conditions pour permettre une reconstitution ultérieure réussie d'objets. En utilisant un contexte formel à l'aide de view path pour identifier les metadonnées nécessaires ainsi que les composants logiciel additionnels, il est possible de dégager des hypothèses pour étendre le modèle de référence OAIS.*

### Inhaltsübersicht

0	Einleitung .....	11
1	Statische und dynamische digitale Objekte.....	12
2	Digitale Archive als neuartige Aufgabenstellung .....	13
3	Migration und Emulation .....	14
3.1	Migration .....	14
3.2	Emulation .....	16
4	Emulatoren – Brücken zwischen digitalem Gestern und Morgen.....	17
4.1	Metadaten und Auswahlkriterien.....	17
4.2	Langzeitverfügbarkeit von Emulatoren .....	18
4.3	Beispiele von verfügbaren Emulatoren.....	18
5	Vom Objekt zu seiner Darstellung .....	19
5.1	Digitale Archäologie.....	20
5.2	Darstellungswege formalisieren.....	20
5.3	Metriken und Aufwandsabschätzungen.....	21
5.4	View-Paths im technologischen Wandel.....	23
5.5	Archivtypen im Betrieb.....	23
6	Softwarearchiv der Sekundärobjekte.....	23
7	Fazit .....	24

### 0 Einleitung

Die Langzeitverfügbarkeit von Wissen in digitaler Form stellt die Menschheit vor neuartige Herausforderungen. Digitale Objekte erweisen sich, anders als traditionelle Medien des Bibliothekswesens wie Papier oder Microfiche, als nicht mehr trivial ohne ihren technischen Erstellungskontext zugreifbar. Diese Kontexte verändern sich

schnell, und wenn Informationen nicht verloren gehen sollen, müssen diese entweder mit der technischen Entwicklung mitgeführt werden oder ihre ursprüngliche Arbeitsumgebung muss erhalten bleiben. Gedächtnisorganisationen wie Bibliotheken oder Archive sehen sich seit einigen Jahren mit dem Auftrag konfrontiert, verhältnismäßig neuartige, nämlich digitale Artefakte für lange Zeit aufbewahren und zugänglich halten zu sollen.

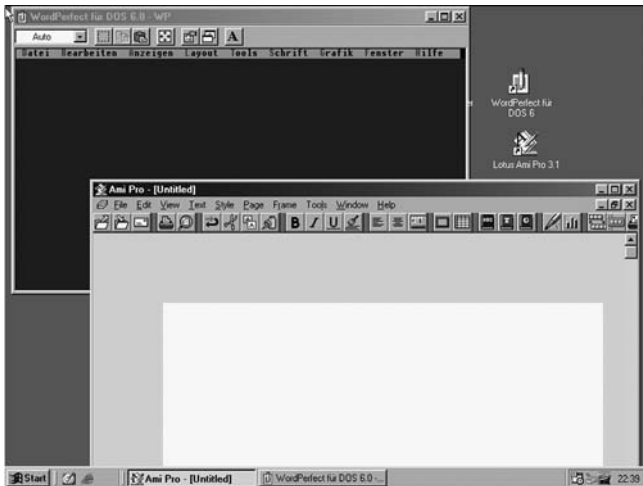


Abb. 1: Textdokumente, die mit den einst populären Textverarbeitungen Ami Pro oder Word Perfect erstellt wurden, lassen sich, obwohl noch nicht einmal älter als 15 Jahre, nicht mehr trivial in aktuellen Applikationen öffnen. Hier kann die Emulation einer alten Arbeitsumgebung, beispielsweise eines Windows 98, helfen.

Die den meisten Gedächtnisorganisationen vertrauteste Methode zur Langzeitarchivierung digitaler Objekte ist die Migration. Sie führt schrittweise ein digitales Objekt durch die sich mit der Zeit wandelnden digitalen Umgebungen, bestehend aus spezifischen Hardware-Software-Konfigurationen, mit. Am Objekt und seiner Struktur nimmt sie dafür notwendige Änderungen vor. Erst diese Anpassungen ermöglichen es, in den jeweils aktuellen rechnergestützten Umgebungen des Archivnutzers digitale Objekte zu betrachten. Diese Herangehensweise schränkt jedoch die Menge der archivierbaren Objekttypen unnötig ein, da sie sich nur mit Aufwand auf dynamische digitale Objekte anwenden lässt. Ein darüber hinaus nicht zu vernachlässigendes Problem ergibt sich aus den zwangsläufigen Veränderungen des Objekts. Diese werfen Fragen wie die nach der Authentizität auf. Letztere stellt jedoch ein zentrales Moment vertrauenswürdiger Langzeitarchive dar.

Die in Betracht gezogenen Strategien der funktionalen Langzeitarchivierung sollten deshalb dahingehend ergänzt werden, dass sie jederzeit einen Zugriff auf das Objekt in seinem Originalzustand erlauben und Ergänzungen implementieren, die überhaupt erst eine spätere Darstellung bestimmter Objekttypen gewährleisten.

Emulation, eine digitale Langzeitarchivierungsstrategie, die verstärkt wahrgenommen wird, verwendet einen anderen Ansatz als Migration. Sie operiert nicht auf dem Objekt selbst, sondern setzt an seiner ursprünglichen digitalen Umgebung an und versucht diese über die technologische Weiterentwicklung hinweg zu erhalten. Emulation verhindert auf diese Weise, dass die Primärobjekte eines digitalen Langzeitarchivs – die eigentlichen Objekte der Bewahrung – verändert werden müssen. Jede Schicht einer Software-Hardware-Konfiguration kann als Ansatzpunkt für Emulation dienen, wobei eine Reihe von Erfolgsbedingungen zu berücksichtigen ist. Hierzu zählt die Formalisierung von Zugriffsabläufen, die vom interessierenden Objekt startend auf die jeweils gültige Arbeitsumgebung des Archivnutzers zielen. Die Verfolgung des Emulationsansatzes bedarf einer ganzen Reihe von

Überlegungen: So ist bereits beim Einstellen digitaler Archivalien in ein Langzeitarchiv festzustellen, welche Art von Arbeitsumgebung und welche Zusatzinformationen für spätere Zugriffe benötigt werden. Diese Informationen sollten sich anhand der Objektmetadaten beispielsweise unter Zuhilfenahme bereits bestehender Format-Registries ermitteln lassen.

Was generell für digitale Objekte gilt, ist speziell auch für die Strategie der Emulation richtig: Auf digitale Objekte kann nicht ohne technische Hilfsmittel zugegriffen werden. So wie für das Abspielen von Schallplatten der Plattenspieler aufbewahrt werden muss, benötigt man ein Archiv an Werkzeugen. Ebenso erforderlich sind geeignete Referenzumgebungen, fest definierte Software-Hardware-Konfigurationen, als Bezugspunkte zur Wiedergabe digitaler Objekte. Während sich bereits die sinnvolle physische Erhaltung des technisch recht trivialen Gerätes Plattenspieler als nicht unproblematisch erweist, gilt dies erst recht für digitales Equipment. Bei geeigneter Wahl der Mittel kann jedoch der Fall einfacher liegen – man muss kein Hardwaremuseum aufbauen und über lange Zeit pflegen. Stattdessen kann dieses einem speziellen Softwarearchiv mit Hardwarenachbildungen, den Emulatoren, übertragen werden.

Trotzdem müssen Emulatoren, welchen die Aufgabe der Brückenfunktion zwischen dem technischen Gestern und Morgen zukommt, als Software, die wie alle digitalen Objekte von Veraltung betroffen ist, geeignet ausgewählt und behandelt werden.

## 1 Statische und dynamische digitale Objekte

Mit sich verändernden Technologien, Anwendungen und Kommunikationsformen wandelt sich zwangsläufig die Art der digitalen Objekte. Im Laufe der Zeit kommen neue Objekttypen hinzu und andere werden mit der Zeit abgelöst. Mit Daten des klassischen Typs seien digitale Objekte in Form virtueller Dokumente gemeint. Das sind Objekte, die nicht mehr primär in Papierform erstellt werden, aber für eine eventuelle Wiedergabe auf Papier gedacht sind. Zu dieser Klasse rechnet man alles, was in Behörden, Unternehmen und Organisationen an Korrespondenz, Berichten, Abrechnungen oder Strategiepapieren generiert wird. Für die meisten dieser Daten genügt es häufig sicherzustellen, dass sie innerhalb ihrer gesetzlichen Aufbewahrungsfrist eingesehen werden können. Digitale Artefakte von gesellschaftlicher oder geschichtlicher Relevanz sollten diese Fristen überdauern können.

Ein Merkmal charakterisiert die genannten Objekte: Sie lassen sich zumindest theoretisch zu jedem Zeitpunkt auf einen extradigitalen Datenträger, wie Papier oder Mikrofiche, umkopieren. Audio- und Videostücke können auf analoge Datenträger überspielt werden. Komplexer wird es beispielsweise bei Konstruktionsplänen langlebiger Industriegüter. Sie lassen sich ausdrucken, jedoch gehen typischerweise Informationsebenen und Zusammenhänge verloren.

Die vorgenannten „traditionellen Datenträger“ benötigen geringen oder gar keinen technischen Aufwand, damit Endbenutzer die enthaltenen Informationen erfassen und auswerten können. Das stimmt nicht mehr uneingeschränkt für dynamische Daten und interaktive Objekte. Sie finden in der aktuellen Langzeitarchivierungsdebatte bisher noch geringe Beachtung. Bei diesen Objekten han-

delt es sich um ausführbare Programme, Komponenten oder Softwarebibliotheken. Zu ihnen zählen: Interaktive Medien und Lehrsoftware, Datenbanken oder Betriebssysteme und Applikationen. Dynamische Daten sind dadurch gekennzeichnet, dass sie außerhalb eines festgelegten digitalen Kontexts nicht sinnvoll interpretiert und genutzt werden können.

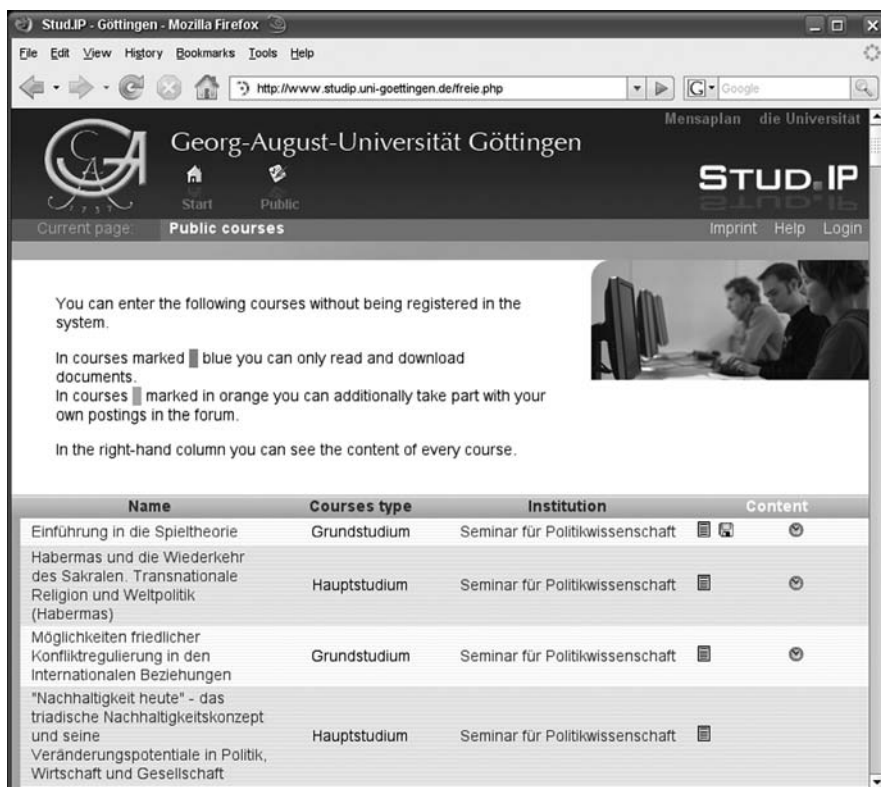


Abb. 2: Zunehmend setzen sich an Universitäten dynamische Lehrplattformen, wie beispielsweise StudIP, durch. Diese fassen einerseits elektronische Lehrmaterialien verschiedenster Formen zusammen, strukturieren Kurse und bieten andererseits dynamische Interaktion zwischen Teilnehmern und Dozenten.

Mit der breiten Durchsetzung des Computers und der begleitenden Revolution auf dem Gebiet der Datennetze kamen neue Bereiche hinzu, in denen digitale Objekte geschaffen wurden. Während in den Anfangszeiten des Internets oder des für die meisten Menschen sichtbaren Teils World Wide Web (WWW) noch statische Inhalte dominierten, die mit klassischen digitalen Dokumenten und Bildern vieles gemein hatten, so wurde dieses Medium im Laufe der Zeit deutlich interaktiver.

Dynamische Web-Auftritte, unterstützt von Multimedia-Formaten wie Flash oder großen Datenbanken, dominieren die großen Sites im Internet. Viele Web-Anwendungen führen den Hauptteil der Interaktion auf dem Server des jeweiligen Inhaltanbieters aus. Viele Anwendungen funktionieren nur im Zusammenspiel der Server mit speziellen Clients. Stark erweitert wird dieser Bereich, gerade im Umfeld von Universitäten, durch die Nutzung des Mediums als Lernplattform. Auch hier gilt: Die Abläufe von E-Learning-Systemen einfach abzufilmern, erweist sich in den meisten Fällen als ziemlich unbefriedigend. Aus interaktiven Applikationen mit einer Vielzahl möglicher Navigationspfade wird so eine lineare Darstellung eingeschränkter Komplexität und Aussagekraft. Mit den neu hinzugekommenen Aufgaben der Web-Archi-

vierung vieler Nationalbibliotheken gewinnt das Thema WWW an Brisanz.

Datenbanken zählen ebenfalls zu den sehr frühen Anwendungen von Rechnern. Die Bewegung, Durchsuchung und Verknüpfung großer Datenbestände gehört zu den großen Stärken und deshalb zu den Haupteinsatzgebieten von Computern. Diese elektronischen Datenbestände stellen oft die Grundlage für abgeleitete Objekte dar. Zur Klasse der datenbankbasierten Anwendungen zählen Planungs- und Buchhaltungssysteme, wie SAP-Anwendungen, elektronische Fahrpläne diverser Verkehrsträger bis hin zu Content Management Systemen (CMS) moderner Internetauftritte von Firmen und Organisationen. Fahrpläne von Eisenbahngesellschaften sind ein gutes Beispiel großer Datenbanken: Wo früher in Form von Kursbüchern Jahresfahrpläne archiviert wurden, ergeben sich an dieser Stelle komplett neue Herausforderungen. So wird beispielsweise seit 2008 kein Gesamtfahrplan von British Rail mehr gedruckt, einem durchaus häufig nachgefragten Objekt der British Library.

2 **Digitale Archive als neuartige Aufgabenstellung**

## 2 Digitale Archive als neuartige Aufgabenstellung

Bibliotheken besitzen nur bedingten Einfluss auf erstellte und eingelierte Formate. Sie sind angehalten, die meisten digitalen Objekte für den permanenten Zugriff seitens ihrer Nutzer bereitzuhalten. Sie müssen daher ihren oft technisch wenig versierten Besuchern eine geeignete

Plattform bereitstellen, die die Darstellung der gängigen Formate erlaubt. Da sich nicht alle Objekte für den Zugriff in die jeweils aktuelle Form migrieren lassen, muss diese Plattform zudem mit einer großen Menge alter Formate umgehen können.

Digitale Archive stellen neue Anforderungen an ihre Betreiber. Diese Erkenntnis schlägt sich beispielsweise in der Verabschiedung von Standards wie OAIS, dem Open Archival Information System, nieder. Dieser von der ISO aufgelegte Standard beschäftigt sich mit der Aufnahme, der Verwaltung und der Ausgabe digitaler Objekte. In der bisherigen Formulierung bietet OAIS lediglich eine grobe Vorgabe zum Aufbau und zur Organisation. Deshalb sind die aktuellen Forschungsprojekte und Initiativen mit seiner Verfeinerung und Konkretisierung befasst<sup>1</sup>.

<sup>1</sup> Siehe beispielsweise Holdsworth und Sergeant: A Blueprint for Representation Information in the OAIS Model, NASA Goddard Conference on Mass Storage Systems and Technologies 2000. Diverse Langzeitarchivierungsprojekte, wie PLANETS <<http://www.planets-project.eu>> oder Caspar <<http://www.casparpreserves.eu>>, befassen sich in Teilprojekten mit OAIS-Erweiterungen.

OAIS ist nicht der einzige Standard, dem Aufmerksamkeit gebührt: Neben Standards für digitale Objekte selbst, beispielsweise Datenformate, sind Metadatenstandards zur Klassifizierung und Beschreibung von Bedeutung. Für alle Standards gilt, dass diese optimalerweise offengelegt und frei von proprietären Komponenten sind. Sie erleichtern und ermöglichen in Zukunft die Langzeitarchivierung, setzen sich jedoch mit einer gewissen Verzögerung durch und wirken nicht in die Vergangenheit. Zudem lassen sich nicht für alle Objekte und Prozesse Standards definieren oder festlegen.

Unabhängig vom konkreten Objekttyp erfolgt bei der Einstellung in ein OAIS-konformes Archiv eine Anreicherung mit zusätzlichen Informationen, den Metadaten. Inzwischen wurde eine ganze Reihe von Metadaten-Schemata definiert, von denen Dublin Core oder PREMIS als flexible und erweiterbare Modelle herausragen und sich im Bereich der digitalen Langzeitarchivierung als mögliche Standards herauskristallisieren.

### 3 Migration und Emulation

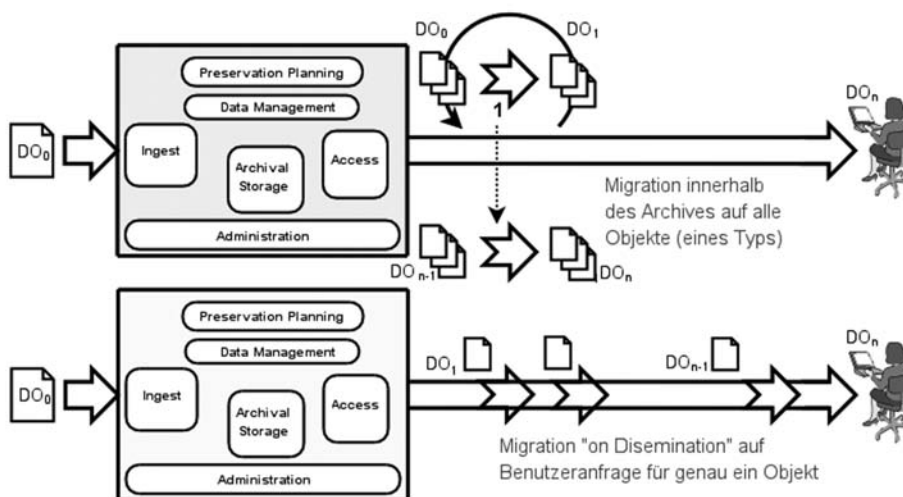


Abb. 3: Die Wahl einer Langzeitarchivierungsstrategie entscheidet nicht nur über den Managementaufwand im Archiv, sondern auch über die Anzahl anzufassender Objekte und das, was ein Benutzer unter welchem Aufwand am Ende zu sehen bekommt.

Während Digitalisierung und Standardisierung im Vorfeld der Archivaufnahme ansetzen, dienen Migration und Emulation dem späteren Zugriff auf digitale Objekte. Die Strategien arbeiten gegensätzlich: Migration hat zur Aufgabe, ein Objekt mit der technischen Entwicklung mitzuführen und für die jeweils aktuellen rechnergestützten Arbeitsumgebungen zugreifbar zu halten. Demgegenüber setzt Emulation an der Umgebung für das jeweilige Objekt an. Migration verändert das Objekt selbst mit allen Vor- und Nachteilen, Emulation hingegen nicht. Der Aufwand für das Management des Langzeitarchivs fällt je nach Strategie zu verschiedenen Zeitpunkten an. Migration hat jedoch einige Nachteile, die eine Anwendung bei bestimmten Objekttypen weitgehend verhindern. Jedoch hängt auch die Emulation von einigen Erfolgsbedingungen ab<sup>2</sup>.

### 3.1 Migration

Unter Migration im Kontext von Computern wird eine Umwandlung eines digitalen Objekts in ein abgeleitetes unter Beibehaltung einer Reihe von Eigenschaften und unter Anwendung bestimmter Kriterien verstanden. Üblicherweise handelt es sich hierbei um die Überführung von Dateien in die jeweils gerade übliche und mit aktuellen Programmen betracht- oder ausführbare Repräsentation. Dieser oft auch als Datenformatkonvertierung bezeichnete Vorgang soll den optimalen Zugang zu den Objekten zum jetzigen und möglichst auch für zukünftige Zeitpunkte sichern. Das Konzept der Migration von Computerdaten beschränkt sich nicht allein auf digitale Langzeitarchivierung. Allein die Vielzahl der Datenformate, die aufgrund einer großen Menge an Applikationen zu einem gegebenen Zeitpunkt erzeugt werden können, zwingt Benutzer permanent zu Migration, wenn sie ihre Daten untereinander austauschen wollen. Deshalb sind viele Benutzer mit Migration oder Konvertierung in der täglichen Computernutzung durchaus vertraut.

Migration als Formatumwandlung kann dabei sowohl horizontal als auch vertikal – bezogen auf eine gegebene Zeitebene – stattfinden. Horizontale Migration meint das Nachvollziehen der regelmäßigen Updates seitens der Softwareproduzenten. Da mit jeder neuen Version üblicherweise der Software neue Fähigkeiten hinzugefügt werden, hat dieses fast zwangsläufig Auswirkungen auf die Datenformate, um die neuen Eigenschaften abbilden zu können. Sollen dann ältere Dateien, die mit Vorgängerversionen erstellt wurden, geöffnet werden, findet quasi per Importfunktion ein Übergang auf die neue Software statt.

Vor ähnlichen Problemen stehen Nutzer verschiedener Softwareprodukte, die den gleichen Zweck erfüllen. Selten steht nur ein Produkt zur Verfügung. Anwendungen mit ähnlicher Funktionalität lassen sich nach ihrem Einsatzzweck klassifizieren, wobei sich Produkte in ihrer Funktionalität und in ihrem Bedienungsmodell durchaus erheblich unterscheiden können. Ein Beispiel wären die verschiedenen Applikationen zur Erstellung von Texten, wie Microsoft Word, StarOffice Writer oder historische Vertreter wie Ami Pro, Wordstar und WordPerfect. Der Im- und Export zwischen diesen bezeichnet vertikale Migration. Es sollte betont werden, dass es nicht eine einzige Migrationsstrategie gibt, sondern Migration eher als generelles Konzept, welches einen Korb von Strategien um-

<sup>2</sup> Eine Diskussion der verschiedenen Strategien, ihrer Vor- und Nachteile findet sich in Granger, Stewart: Emulation as a Digital Preservation Strategy. In: D-Lib Magazine 10 (2000) oder Hedstrom, Margaret und Clifford Lampe: Emulation vs. Migration: Do Users Care? In: DigiNews 2 (2001).

fasst, verstanden werden sollte. Gerade die Vielfalt der beteiligten Datenformate fordert einen sehr unterschiedlichen Umgang mit den verschiedenen Daten. Migration erfordert die Beschäftigung mit der Auswahl von Datenformaten und Standards, der Transformation zwischen Datenformaten und der Wahl von und Migration zwischen Datenträgern.

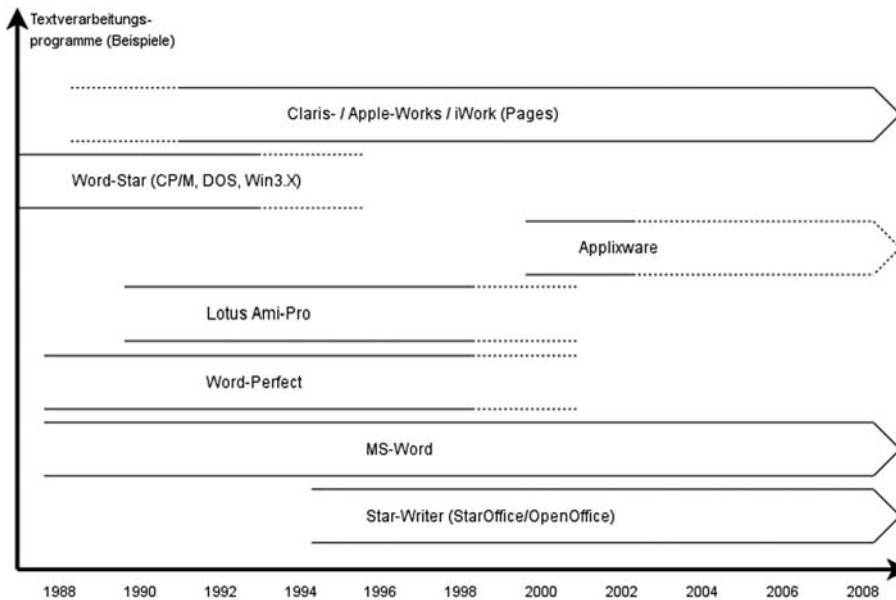


Abb. 4: Horizontale und vertikale Migration am Beispiel einiger Textverarbeitungsprogramme im Zeitablauf. Eine Reihe von Applikationen ist inzwischen ohne Nachfolger eingestellt worden und die herausgebenden Softwarefirmen sind vom Markt verschwunden.

Generell sollte man festhalten, dass Migrationsprozesse, die Datentransformationen enthalten, fast grundsätzlich unter dem Verdacht der Verlustbehaftung stehen. Finden mehrere Transformationsprozesse nacheinander statt, können sich Informationsverluste summieren<sup>3</sup>.

Ein nicht zu unterschätzendes Problem bei der Transformation der logischen Struktur ist die Frage nach der Authentizität des digitalen Objekts. Diese Fragestellung tritt bei fast allen Migrationsvorgängen, wie dem Upgrade innerhalb eines Softwareprodukts, dem Wechsel von einem Hersteller zu einem anderen oder dem nachfolgerlosen Ende einer Applikation, auf. Mit folgenden Herausforderungen ist zu rechnen:

- Es müsste bei jedem Migrationsschritt sichergestellt werden, dass der Vorgang, umkehrbar gemacht, das identische Ausgangsobjekt liefert.
- Zudem müssten alle Ausgangsdaten, auch die aller Zwischenschritte, aufgehoben werden, um spätere Vergleiche zu erlauben. Dieses empfiehlt sich jedoch generell.
- Die erstgenannten Punkte setzen aber die Verfügbarkeit der beteiligten Applikationen voraus. Das Problem der Abkündigung einer Software, dem Konkurs eines Unternehmens oder das beständige Upgrade lassen dieses jedoch bezweifeln.
- Es müsste die schrittweise Migration mit allen Objekten vollzogen werden. Viele Softwareprodukte erzwingen genau dieses, weil ihre „Format-Historie“ oft nicht sehr weit zurückreicht. Dieser Ansatz ist jedoch aufwändig. Es müsste eine Kontrolle auf die Qualität des Migrationsergebnisses erfolgen, die sich nicht in allen Aspekten automatisieren lässt. Generell muss in

Frage gestellt werden, ob eine Authentizitätsprüfung überhaupt leistbar ist, da in den meisten Fällen der reine Augenschein eines oberflächlichen Vergleichs nicht ausreicht. Zudem nimmt die Datenmenge mit jedem Migrationsschritt zu, da ja laufend neue Objekte erzeugt werden.

- Verlustbehaftete Kompressionsverfahren scheiden in Bezug auf die Erhaltung der Authentizität aus. Es wäre alternativ denkbar, spezielle Objektkopien zu erzeugen, die einen Bandbreiten sparenden Transport zum Endanwender erlauben.

In Anbetracht der beständig anwachsenden Datenmenge könnte eine Reduktion des Gesamtdatenvolumens angeraten erscheinen. Ein Aussortieren scheinbar obsoleter digitaler Objekte sollte jedoch weitgehend vermieden werden, da Vorhersagen, welches Objekt später noch einmal benötigt wird, sehr schwer zu treffen sind. Zudem könnten sich die Interessenschwerpunkte nachfolgender Generationen komplett ändern. Dieses kann ebenso nur schlecht extrapoliert werden.

Der Aufwand von Transformationen ist nicht zu unterschätzen: Unabhängig von der aktuell vermuteten Relevanz bestimmter Objekte ist immer der komplette Datenbestand zum Stichzeitpunkt zu migrieren. Dabei dürfen begleitende Maßnahmen der Qualitätssicherung und Kontrolle

nicht vernachlässigt werden, da sonst Datenverluste die Folge sind. Migrations- und Verifikationsprozesse können nicht unwesentliche Rechenkapazitäten beanspruchen, die sich für eine große Zahl von Objekten auf lange Zeiträume erstrecken. Planungen solcher Prozesse waren Bestandteil von Kopal<sup>4</sup>.

Ein generelles Problem ist der beständige Wandel der Informationstechnologien. Diesem Wandel sind die verschiedenen Softwareprodukte und -hersteller unterworfen, so dass Vorhersagen zur langfristigen Beständigkeit bestimmter Produkte oder Produktgruppen nur schwer zu treffen sind. Damit besteht für fast alle digitalen Objekte, die nicht in übergreifenden Standards abgelegt sind, die Gefahr, dass die Kette der möglichen Migrationsschritte irgendwann einmal abreißt. Am Ende des Lebenszyklus können dann wichtige Eigenschaften verloren gehen<sup>5</sup>.

<sup>3</sup> Eine gute Beschreibung dieser Problematik und einen möglichen Lösungsvorschlag liefern Mellor, Phil; Wheatley, Paul und Derek Sergeant: Migration on Request – A practical technique for digital preservation <<http://www.si.umich.edu/CAMILEON/reports/migreq.pdf>>.

<sup>4</sup> Kopal – Co-operative Development of a Long-Term Digital Information Archive <<http://kopal.langzeitarchivierung.de>> – setzte sich unter anderem mit den Herausforderungen der Bitstream-Preservation auseinander.

<sup>5</sup> Einen Überblick für Nichttechniker bietet Rohde-Enslin, Stefan: Nicht von Dauer – Kleiner Ratgeber für die Bewahrung digitaler Daten in Museen. Berlin 2004.

### 3.2 Emulation

Gegeben durch den Charakter der meisten in Bibliotheken vorkommenden digitalen Objekte oder Digitalisate dominiert die Migration. Sie versucht, Objekte mit der technologischen Entwicklung „mitzuführen“ und damit in den jeweils aktuellen Arbeitsumgebungen zugänglich zu machen. Diese Herangehensweise hat einige Schwächen und ignoriert komplett dynamische digitale Objekte. Eine weitere oft unbefriedigend beantwortete Frage ist die nach der Authentizität migrierter, also im Laufe der Zeit zwangsläufig veränderter Objekte. Wie bei alten Originalausgaben von wichtigen Werken auch oder bei juristischen Dokumenten ist die später nachprüfbare Echtheit von digitalen Objekten nicht zu vernachlässigen.

Emulation setzt dabei nicht am digitalen Objekt selbst an, sondern befasst sich mit der Umgebung, die zur Erstellung dieses Objekts 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. Einer der ersten, der auf diesen Ansatz hinwies, war J. Rothenberg<sup>6</sup>. Emulatoren dienen der Erhaltung beziehungsweise Wiederherstellung alter Arbeitsumgebungen. Dadurch erfüllen sie eine Verbindungsaufgabe zwischen der jeweils aktuellen Rechnerplattform oder einer Zwischenstufe und der zum Entstehungsmoment des Objekts gültigen Erstellungsumgebung. Emulation heißt zuerst einfach erst einmal nur die Erschaffung einer virtuellen Umgebung in einer gegebenen Arbeitsumgebung. Das kann bedeuten, dass Software durch eine andere Software nachgebildet wird, ebenso wie Hardware in Software. Ursprünglich für diese Arbeitsumgebung erstellte Objekte sollten dann weitgehend so arbeiten oder aussehen wie in der Originalplattform oder -applikation. Emulatoren bilden die Schnittstelle zwischen dem jeweils aktuellen Stand der Technik und einer längst nicht mehr verfügbaren Technologie. Deshalb müssen sich Emulatoren um die geeignete Umsetzung der Ein- und Ausgabesteuerung bemühen.

Generell lässt sich feststellen, dass für aktuelle Computerplattformen üblicherweise drei Ebenen für den Ansatz von Emulation identifiziert werden können<sup>7</sup>:

- Emulation oder Ersatz von Applikationen durch andere, um so bestimmte Datenformate darzustellen oder auszuführen: Für viele Anwender sicherlich am nahe liegendsten ist die Emulation der Eigenschaften älterer Applikationen in aktuellen Anwendungsprogrammen. Die Interpretation einer Applikation nicht-eigener aktueller Datenformate fällt ebenso in diesen Bereich. Auf diese Weise wird ein Datenaustausch zwischen verschiedenen Programmen unterschiedlicher Hersteller überhaupt erst möglich. Aus Sicht der Langzeitarchivierung genügt üblicherweise ein Betrachter, der in der jeweils aktuellen Arbeitsumgebung zur Sichtbarmachung des gewünschten Objekts ausgeführt werden kann. Hierbei ist festzuhalten, dass nicht unbedingt der komplette Funktionsumfang einer Software für das Rendering und die Betrachtung erforderlich wird. Für eine ganze Reihe generischer Datenformate wie einfache ASCII-, Bild- oder Audiodateien existiert oft eine Reihe verschiedener Betrachter für die unterschiedlichsten aktuell verfügbaren Plattformen. Je spezieller oder komplexer die Datenformate werden, desto unwahrscheinlicher wird die Existenz eines geeigneten Betrachters, der in der

Lage ist, alle gewünschten Objekteigenschaften in der gewünschten Qualität wiederzugeben.

- Emulation von Betriebssystemen, um so auf einem gegebenen Betriebssystem Applikationen auszuführen, die die Schnittstellen (API) eines anderen Betriebssystems erwarten: Die Emulation der Betriebssystem-APIs scheint einen generelleren Ansatz als die Nachbildung von Applikationen zu versprechen: Theoretisch kann auch sehr alte Software auf modernen Betriebssystemen installiert werden, solange alle aufgerufenen Funktionen in der von der Applikation erwarteten Form verfügbar sind. Bei unvollständiger Nachbildung der notwendigen APIs ist dieser Weg jedoch nicht mehr realisierbar. Da gerade im Bereich kommerzieller, proprietärer Software viele Schnittstellen nicht oder nur wenig offen gelegt sind, ist die Wahrscheinlichkeit einer Inkompatibilität groß.
- Nachbildung eines kompletten Rechners einer bestimmten Architektur: Anders als bei den Programmierschnittstellen der Betriebssysteme, den Funktionen einzelner Applikationen oder den Beschreibungen von Dateiformaten liegen die Schnittstellen der meisten Hardwareplattformen offen. Keine Applikation und kein Betriebssystem muss neu geschrieben oder übersetzt werden, um auf viele Tausend digitaler Objekte wieder zugreifen zu können. Das Funktionsset einer Hardwareplattform ist überschaubar und oft deutlich geringer als das eines Betriebssystems oder einer Applikation. Es existieren zudem deutlich weniger Hardwareplattformen als Betriebssysteme.

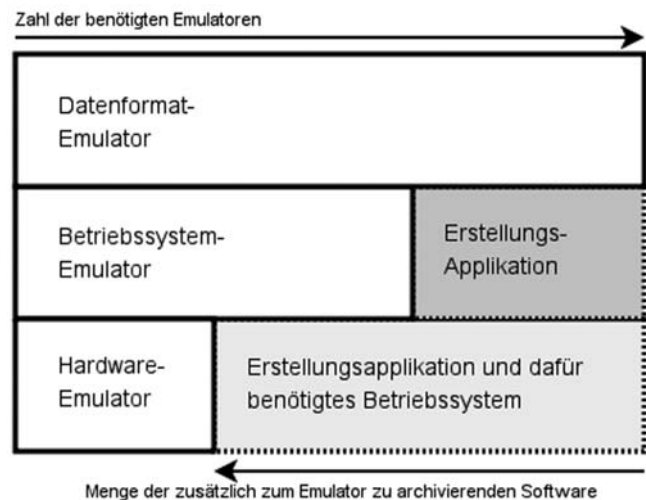


Abb. 5: Betrachtung der Anzahl der zusätzlich zu den Primärobjekten zu archivierenden zusätzlichen Softwarekomponenten: Die Zahl steigt mit der zunehmenden Allgemeinheit der Emulationsschicht.

<sup>6</sup> Vergleiche die Aufsätze von Rothenberg, Jeff: Ensuring the Longevity of Digital Information. New York 1995 und <<http://www.clir.org/pubs/archives/ensuring.pdf>> von 1999. Kompakte Überblicke geben zudem Coy, Wolfgang: Perspektiven der Langzeitarchivierung multimedialer Objekte <<http://edoc.hu-berlin.de/series/nestor-materialien/5/PDF/5.pdf>> und Borghoff; Röding; Scheffczyk und Schmitz: Langzeitarchivierung: Methoden zur Erhaltung digitaler Dokumente. Heidelberg 2003.

<sup>7</sup> Einen Überblick hierzu geben Verdegem, Remco und Jeffrey van der Hoeven: Emulation – To be or not to be, IS&T Conference on Archiving. Ottawa 2006.

#### 4 Emulatoren – Brücken zwischen digitalem Gestern und Morgen

Hardwareemulatoren können zu einer wichtigen Säule der Langzeitarchivierung digitaler Objekte werden. Möchte man die Klasse der zugreifbaren Objekte um dynamische erweitern, führt kein Weg am Erhalt der geeigneten Arbeitsumgebungen für diese Objekte vorbei. Wie bereits dargestellt, eröffnet ein Hardwaremuseum keine wirklich zuverlässige Langfristperspektive, denn die Archivierung von alten Computern ist aus einer Reihe von Gründen kostenintensiv und zunehmend risikoreich<sup>8</sup>. Deshalb bietet eine virtuelle Sammlung vergangener und sich im Prozess der Ablösung befindender Rechnerarchitekturen eine sinnvolle Alternative mit etlichen Vorteilen.

Emulatoren für die unterschiedlichen Computersysteme werden daher zu einem zentralen Bestandteil eines zu schaffenden Softwarearchivs. Dieses Softwarearchiv kann seinerseits wieder einen Bestandteil eines OAIS-basierten Langzeitarchivs bilden. Emulatoren sind ebenso wie die eigentlich interessierenden Primärobjekte von Veraltung betroffen und teilen mit diesen identische Probleme.

Die tatsächliche Erstellung der relevanten Auswahlkriterien hängt von den jeweiligen Archivierungszielen der Gedächtnisorganisationen ab. Zur Kategorisierung und späteren Sichtung spielen Metadaten, die Informationen zum Emulator oder Projekt liefern, eine Rolle. Hierzu zählen:

- die Bedienung der als wichtig erachteten Klassen digitaler Objekte,
- institutionelle Rahmenbedingungen sowie die Identifizierung relevanter Benutzergruppen und ihrer (vermuteten) Bedürfnisse,
- technische und nicht-technische Charakterisierungen und die Abschätzung der Vollständigkeit der Nachbildung,
- Anhaltspunkte zur ökonomischen Bewertung und die Abschätzung der Langfristorientierung.

##### 4.1 Metadaten und Auswahlkriterien

Die geeignete Auswahl von Werkzeugen zur Langzeitarchivierung gehört zu den wesentlichen Erfolgsmerkmalen der Emulationsstrategie. Dabei fällt es jedoch schwer, eine abschließende Liste an Auswahlkriterien aufzustellen, da diese sehr stark von der Benutzergruppe, der Art der Primärobjekte und der geplanten Präsentation dem Endbenutzer gegenüber abhängen. So gilt weiterhin, dass einerseits eine möglichst weit gefasste Auswahl die Wahrscheinlichkeit des korrekten Erhalts einer bestimmten Arbeitsumgebung steigert, andererseits dies die Kosten des Archivbetriebs in die Höhe treibt.

Metadaten und die Auswahlkriterien von Emulatoren hängen eng zusammen. Metadaten sind Informationen, die zu den meisten digitalen Objekten in der einen oder anderen Form schon erhoben wurden oder zur Einstellung in ein Softwarearchiv sinnvollerweise erhoben werden müssen. Sie unterscheiden sich damit nicht wesentlich von Primärobjekten, jedoch werden neben den klassischen Metadaten weitere Informationen zu ihrer Nutzung abgelegt werden müssen. Deshalb soll die Liste der Basisinformationen um einige Gruppen spezifischer Auswahlkriterien erweitert werden. Diese Auswahlkriterien können durchaus eine komplexere Matrix ergeben. Dabei wird es vorkommen, dass die hohe Bewertung eines Kriteriums die

Erfüllung anderer Kriterien erschwert. Da es jedoch nicht „die eine Archivierungsstrategie“ geben sollte, wird man die Auswahl der Werkzeuge eher breiter treffen. Auf diese Weise lässt sich eine höhere Redundanz der „View-Paths“, die in einem späteren Abschnitt eingeführt werden, erreichen.

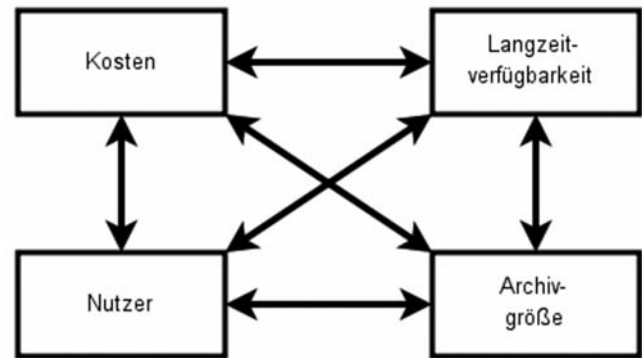


Abb. 6: Beispiele für die Interdependenzen von Auswahlkriterien: So steigen mit den Ansprüchen der Nutzer beispielsweise die zu planenden Kosten des Archivs ebenso wie mit steigender Absicherung der Langzeitverfügbarkeit.

Die Liste der Auswahlkriterien lässt sich in Kategorien gruppieren:

- allgemeine Metadaten wie Name, Hersteller oder Projekt, Homepage im Internet und Historie der letzten Änderungen, Abhängigkeiten von der Host-Plattform, die Software-Hardware-Umgebung, auf der ein Emulator ausgeführt wird, in ihrer konkreten Ausführung auch als Referenz-Workstation bezeichnet.
- Benutzergruppen – Universitäts-, Landes- oder Nationalbibliotheken haben unterschiedliche Ansprüche an Fristen der Objektarchivierung. Darstellung einzelner Objekte, Fähigkeiten im Umgang mit Objekten, Frequenzen des Zugriffs auf Objekte in ihren Langzeitarchiven.
- Technische Kriterien bestimmen die Fähigkeiten und Einsatzbereiche von Emulatoren, machen Aussagen zur Performance, Steuerbarkeit und der eventuell vorhandenen Möglichkeit der nichtinteraktiven Nutzung. Ebenso relevant sind die Anforderungen von Emulatoren an ihre technische Umgebung. Für die langfristige Bewertung und Nutzung spielen Fragen wie Programmiersprache und Dokumentation eine wichtige Rolle. Die technischen Kriterien werden sich je nach Nutzungsumgebung unterscheiden. So sind die Hardwareparameter der Emulation eines Amiga 500 deutlich beschränkter als die große Varianz der X86-PCs.

<sup>8</sup> Vergleiche die Anmerkungen zur Hardwaresammlung des Computer Museum Universität van Amsterdam <<http://www.science.uva.nl/museum>> oder Dooijes, E.H.: Old computers, now and in the future <[http://www.science.uva.nl/museum/pdfs/oldcomputers\\_dec2000.pdf](http://www.science.uva.nl/museum/pdfs/oldcomputers_dec2000.pdf)> Amsterdam 2000, oder auch Reliance on computer museums in Granger, Stewart: Emulation as a Digital Preservation Strategy. In: D-Lib Magazine 10 (2000).

- Nichttechnische Kriterien beinhalten die ökonomische Bewertung einzelner Tools wie Lizenzkosten und Support. Hierzu zählt ebenso die Produkt- oder Projektgeschichte, die Marktdurchdringung, die Existenz und das Wissen einer Anwender-Community.

Während bisher kein Emulator primär als Instrument zur Langzeitarchivierung digitaler Objekte programmiert wurde, ändert sich dieser Zustand im Laufe der aktuellen Forschungen. So kommt zu den bereits etablierten Emulatoren in jüngerer Zeit mit Dioscuri ein X86-Emulator hinzu, der primär unter dem Gesichtspunkt der Langzeitarchivierung entwickelt wird. Dieser besitzt jedoch noch nicht den Funktionsumfang der anderen Programme. Von einiger Bedeutung ist zudem die Gruppe der Virtualisierungsprogramme der X86-Architektur. Typischerweise für ganze andere Zwecke eingesetzt, eignen sie sich jedoch für eine mittelfristige Perspektive und befördern die Idee und das Wissen um Hardwareemulation erheblich.

#### 4.2 Langzeitverfügbarkeit von Emulatoren

Emulatoren sind selbst nichts anderes als dynamische digitale Objekte. Für sie gelten die analogen Problemstellungen wie für Primärobjekte, die ein Archivbenutzer zu betrachten wünscht. Deshalb sind Überlegungen zur Perpetuierung der Emulatoren für zukünftige Benutzung eine zentrale Komponente. Emulatoren müssen daher immer wieder für die jeweils aktuellen Hardware- und Betriebssystemkombinationen angepasst werden.

Die Möglichkeit der *Softwaremigration* scheint erst einmal im Widerspruch zu den eingangs vorgenommenen Überlegungen zu stehen: So war die Migration dynamischer digitaler Objekte wegen des möglichen sehr hohen Aufwands oder der Unmöglichkeit der Anwendung auf bestimmte Objekte als für die Langzeitarchivierung ungeeignet klassifiziert worden. Dieses Problem lässt sich jedoch durch geeignete Emulatorwahl reduzieren. Wenn der Emulator als Open-Source-Paket vorliegt, kann dafür gesorgt werden, dass eine rechtzeitige Anpassung an die jeweils aufkommende neue Rechnerplattform erfolgt. Vielfach wird es dabei um die Einbindung des Emulators in das aktuelle User-Interface dieser Plattform gehen. Langlebige Programmiersprachen wie beispielsweise C sollten eine Übersetzung mit dem jeweils aktuellen Compiler erlauben. Beides setzt jedoch eine gewisse Stabilität der Paradigmen, sowohl der Host-Betriebssysteme und User-Interfaces als auch der Programmiersprachen voraus. Der Vorteil dieses Ansatzes liegt in der Verwendung nur einer Emulationsschicht. Dies bedeutet, dass die Interaktion und der Datenaustausch mit dem Zielsystem direkt erfolgen kann.

Besteht keine Möglichkeit, den Emulator auf eine neue Host-Plattform zu portieren, kann wiederum die nun veraltete Host-Plattform, für die der Emulator erstellt wurde, selbst emuliert werden. Dann spricht man von *geschachtelter Emulation*. Ein wesentlicher Vorteil besteht in der Vermeidung jeden Migrationsaufwands.

Im Zuge der Forschungsbemühungen in der digitalen Langzeitarchivierung werden weitergehende Ansätze wie die UVC<sup>9</sup> oder modulare Emulation untersucht<sup>10</sup>.

#### 4.3 Beispiele von verfügbaren Emulatoren

Emulatoren für die unterschiedlichen Computersysteme werden daher zu einem zentralen Bestandteil eines

zu schaffenden Softwarearchivs. Dieses Softwarearchiv kann seinerseits wieder einen Bestandteil eines OAIS-basierten Langzeitarchivs bilden. Die Verwendung von Emulatoren ist nicht neu: Im Laufe der letzten 30 Jahre entstand eine Vielzahl, von denen hier drei exemplarisch für zwei Hardwareklassen – den X86-PC und die schon länger nicht mehr verfügbare Klasse der Home-Computer – kurz vorgestellt werden sollen.

Zu den derzeit zweifellos interessantesten Emulatoren muss *QEMU* gerechnet werden<sup>11</sup>. Inzwischen hat sich eine große Benutzer- und Entwicklergemeinschaft gebildet. *QEMU* wurde primär als Emulator für verschiedene Rechnerarchitekturen angelegt. Das Projekt ist Open Source und steht unter der GPL. Die Quelldaten sind damit frei verfügbar. *QEMU* ist eine Applikation ohne grafisches Benutzer-Interface. Es stellt eine Reihe von Kommandozeilenoptionen zur initialen Konfiguration und eine Monitorschnittstelle zur Verfügung. Letztere kann dazu genutzt werden, zur Laufzeit des Emulators Wechsellaufwerke einzubinden und auszuhängen. Auf diesem Weg können verschiedene Arten der Steuerung implementiert werden. Hierzu zählen das von außen angestoßene Wechseln von Disketten oder das „Einlegen“ von virtuellen optischen Datenträgern sowie das Verbinden von USB-Geräten.

Zusätzlich lassen sich Tastatureingaben und Mauseaktionen mittels Monitor-Interface triggern. So können bestimmte Tastenkombinationen übermittelt werden, die typischerweise von der Referenzumgebung abgefangen werden. Auf diese Weise wären ganze Programmabläufe automatisierbar, die sonst eine direkte Benutzerkommunikation erfordern.

Die Niederländische Nationalbibliothek, das Nationalarchiv sowie die Technologieberatung Tessella entwickelten in Kooperation eine zuerst 16bittige Version eines X86-Emulators in Java – *Dioscuri*. Der Name des Projekts bezieht sich auf die griechische Sage der beiden Zwillinge Castor und Pollux, von denen einer sterblich und der andere unsterblich war. Mit diesem Gleichnis soll der Charakter der Emulation ausgedrückt werden: „Endlichen“ digitalen Objekten sollen ihre unsterblichen Äquivalente zur Seite gestellt werden.

Java bietet als plattformunabhängige Programmiersprache die Chance, Programme zu erstellen, die in unterschiedlichen Referenzumgebungen ausführbar sind. Solche Applikationen vermeiden zu starke Abhängigkeiten von bestimmten Kombinationen aus Rechnerarchitektur und Betriebssystem. Der Emulator wurde als modulares Werkzeug angelegt: Jede Hardwarekomponente existiert als eigenständiges Modul, und aus der Kombination von Komponenten, unter Berücksichtigung eines

<sup>9</sup> Universal Virtual Computer als ein einfacher „General Purpose Computer“ im Zuge des DIAS implementiert. Vergleiche hierzu van Diessen, Raymond und Raymond A. Lorie: Long-Term Preservation of Complex Processes, Proceedings at the IS&T Archiving Conference. Washington 2005.

<sup>10</sup> Siehe hierzu die Ausführungen von van der Hoeven, Jeffrey; van Diessen, Raymond und K. van der Meer: Development of a Universal Virtual Computer (UVC) for long-term preservation of digital objects. In: Journal of Information Science 3 (2005).

<sup>11</sup> Betreut und weiterentwickelt wird das Projekt von Fabrice Bellard, vergleiche <<http://fabrice.bellard.free.fr/qemu>>.



gewissen Basis-Sets, wird ein vollständiger Hardwareemulator generiert.

Der überwiegende Anteil der Home-Computer basierte auf den Prozessoren der 68000er Serie von Motorola oder den Zilog Z80-Prozessoren. Zu dieser Basis kamen Komponenten wie Ein- und Ausgabebausteine hinzu, die sich zwischen den verschiedenen Maschinen nicht wesentlich unterscheiden. Deshalb verfolgt MESS einen modularen Ansatz, der eine universelle Basismaschine aufsetzt, die für jedes System, sei es Spielkonsole, Taschenrechner, Handheld oder Bürocomputer, eine Beschreibung erhält. Ausgehend von dieser greift es auf vorgefertigte Emulationscodeblöcke zurück.

Das Akronym MESS steht für Multiple Emulator Super System<sup>12</sup>. MESS ist ebenfalls Open Source und unterliegt der GPL-Lizenz. Es wird von einer größeren Entwicklergemeinschaft getragen und aktiv gepflegt. Als Referenzplattform kommen in erster Linie die Windows-Varianten, dann die weniger komfortabel unterstützten Systeme wie Linux oder Mac-OS X in Frage. Da die Firmwares beziehungsweise Betriebssysteme der einzelnen Geräte unter proprietären Lizenzen stehen, sind sie nicht Bestandteil des Emulators. Sie sind separat zu beschaffen, zu installieren und geeignet in einem Softwarearchiv abzulegen.

### 5 Vom Objekt zu seiner Darstellung

Digitale Objekte können nicht aus sich alleine heraus genutzt werden, sondern bedürfen eines geeigneten Kontextes, der bereits mehrfach angesprochenen Arbeitsumgebung, damit auf sie zugegriffen werden kann. Dieser Kontext, Arbeits- oder Nutzungsumgebung genannt, muss geeignete Hardware- und Softwarekomponenten so kom-

benötigt das entnommene Objekt einen Kontext, der dem Nachfrager eine Interpretation erlaubt. Mit zunehmendem Zeitabstand zur Erstellung des Objekts sinkt die Wahrscheinlichkeit, dass dieser Kontext in der digitalen Arbeitsumgebung des Nutzers bereits vorliegt.

Daraus ergibt sich eine zentrale Aufgabe des Archivbetreibers: Seine Zuständigkeit endet nicht mit der Bitstream-Preservation, der erfolgreichen Bewahrung und Fortschreibung der Archivobjekte, sondern er muss zudem den Zugriff auf alle im Archiv enthaltenen Objekte zu jedem Zeitpunkt sicherstellen.

Hierzu wird eine geeignete Umgebung erwartet, in der ein Archivnutzer das Objekt betrachten oder ausführen kann. Die Benutzergruppen einzelner digitaler Archive und Sammlungen werden sich im Grad ihrer Kenntnisse unterscheiden. Da bei einem durchschnittlichen Nutzer nicht zwingend von einem erfahrenen Computeranwender auszugehen ist, sind Überlegungen anzustellen, wie dieser geeignet an die notwendige Software und ihre Schnittstellen herangeführt werden kann, um mit ihr umgehen zu können. Die Arbeitsabläufe für den Zugriff auf verschiedene Objekttypen können sich unterscheiden:

- Migration-on-Dissemination. Das Primärobjekt wird – je nach Typ auch in mehreren hintereinander geschalteten Stufen – in eine für den Nutzer betrachtbare Form gebracht. Dieser Vorgang kann durchaus Emulationsprozesse involvieren, um ältere Migrationssoftware auszuführen. Er kann je nach Tiefe und Aufwand der einzelnen Schritte längere Zeit in Anspruch nehmen und unter Umständen direkte Nutzerinteraktionen erfordern. Migration-on-Dissemination eignet sich nur für statische Objekte. Das Ziel der Migration ist ein Format, welches von einem in der jeweiligen Referenzumgebung

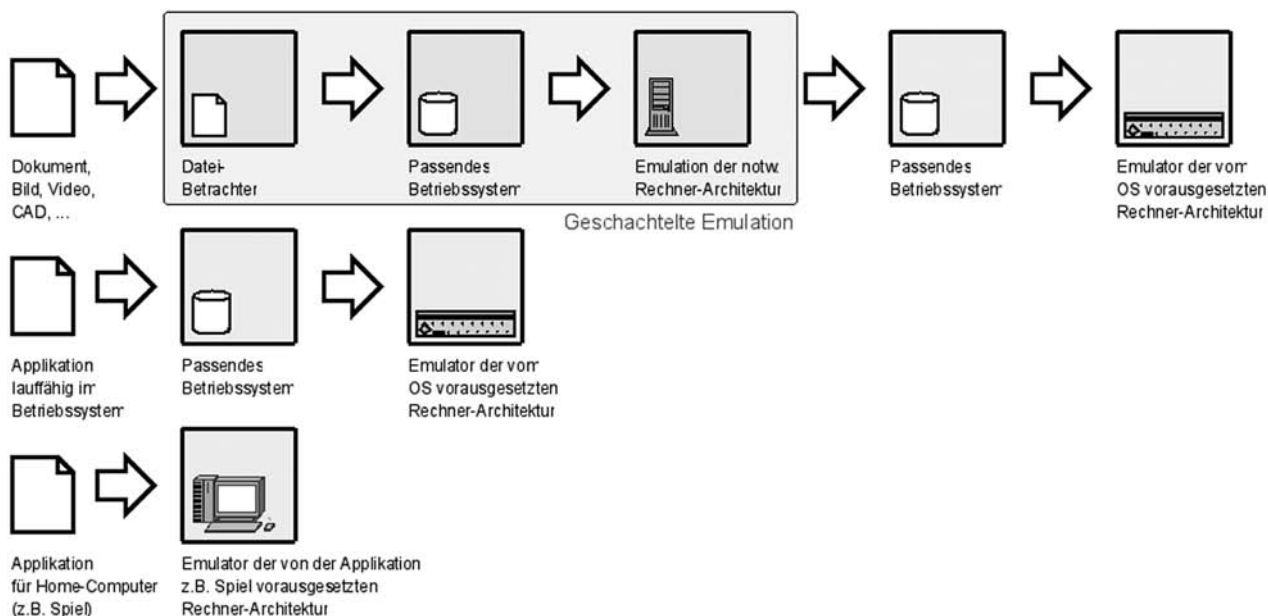


Abb. 7: Die Länge eines View-Paths ist nicht fixiert und hängt vom Typ des Objekts, auf das zugegriffen werden soll, und der Langzeitstrategie zur Erhaltung des jeweiligen Emulators ab.

binieren, dass je nach Typ des Primärobjekts seine Erstellungsumgebung oder ein geeignetes Äquivalent erzeugt wird. View-Paths sind deshalb eines der zentralen Konzepte, die in diesem Aufsatz diskutiert werden. Zu einem gegebenen Zeitpunkt wird ein Primärobjekt von einem Archivbenutzer nachgefragt werden. In jedem Fall

<sup>12</sup> Das Multi Emulator Super System <<http://www.mess.org>> wird von einer größeren Entwicklergemeinschaft getragen und ist auf einer Reihe verschiedener (Referenz-) Plattformen ablauffähig.

verfügbaren Viewer angezeigt oder abgespielt werden kann. Das Objekt wird in dieser Prozedur zwangsläufig verändert und kann damit unter Umständen bestimmten Ansprüchen nicht mehr genügen.

- Wiederherstellung einer Nutzungsumgebung. Für alle anderen vorgestellten Objekttypen verbleibt als Darstellungsoption lediglich die Wiederherstellung der ursprünglichen oder einer äquivalenten Objektumgebung. Dieser Vorgang ist wiederum durch den konkreten Objekttyp und seine spezifischen Anforderungen bestimmt. Die involvierten Prozesse werden in den meisten Fällen eine gewisse Nutzerinteraktion erfordern und zudem eine bestimmte Anzahl zusätzlicher, sekundärer Objekte benötigen. Darüber hinaus stellt sich die Frage des geeigneten Transports des Primärobjekts in die wiederhergestellte Arbeitsumgebung.

Die konkrete Gestaltung des jeweiligen Zugriffs hängt sowohl vom Archivbetreiber und seinen Richtlinien als auch von seiner Benutzergruppe ab. Dieses kann vor Ort beispielsweise durch eine Referenz-Workstation in den Räumen der Bibliothek oder durch einen geeigneten Transport über Kommunikationsnetze geschehen.

### 5.1 Digitale Archäologie

Die bisher angestellten Überlegungen gelten nicht ausschließlich im Zusammenhang mit einem Langzeitarchiv. So kann es durchaus vorkommen, dass aus unterschiedlichen Quellen, wie beispielsweise Nachlässen, digitale Objekte auftauchen, die bisher nicht in einem digitalen Langzeitarchiv gespeichert wurden. Auf diese lassen sich die eingangs vorgestellten Arbeitsabläufe der Objekttypbestimmung und der Darstellung für den Benutzer ebenfalls anwenden. Auf diese Weise könnten etablierte Gedächtnisorganisationen, wie Bibliotheken oder technische Museen, bereits vorhandene Werkzeuge für „Softwarearchäologen“ bereitstellen<sup>13</sup>. Hierzu kann es notwendig sein, geeignete Referenzhardware für das Einlesen von alten Datenträgern und Wege für das Einspielen externer Primärobjekte zur Verfügung zu stellen.

Die geschilderten Möglichkeiten sollten jedoch nicht als verlässliche Alternative eines vertrauenswürdigen Langzeitarchivs gesehen werden: Während das Archivmanagement für jeden enthaltenen Objekttyp zu jedem Zeitpunkt dessen Darstellbarkeit prüfen kann, lässt sich dieses für extern angelieferte Objekte naturgemäß nicht organisieren. So bleibt ein Unsicherheitsfaktor erhalten, ob die jeweilige Gedächtnisorganisation wirklich in der Lage ist, mit dem eingelieferten Objekt geeignet umzugehen.

### 5.2 Darstellungswege formalisieren

Die Wiederherstellung von Nutzungsumgebungen oder entsprechender Äquivalente lässt sich am besten durch so genannte „View-Paths“, Wege ausgehend vom Primärobjekt des Interesses bis zur Arbeitsumgebung des Betrachters oder Anwenders, veranschaulichen und formalisieren. Im Zuge des DIAS-Projekts an der Königlichen Bibliothek der Niederlande wurde dieses Konzept in die Diskussion eingebracht<sup>14</sup>. Die Abbildung verschiedener View-Paths zeigt einen typischen Vektor, ausgehend vom Primärobjekt über seine Erstellungsalpplikation zum

erforderlichen Betriebssystem bis hin zum daraus resultierenden Hardwareemulator.

Während der Ausgangspunkt des View-Paths durch das Primärobjekt fixiert ist, wird sich, extern determiniert durch den technologischen Fortschritt und die sukzessive Obsoleszenz vorhandener Rechnerplattformen, der Endpunkt des View-Path im Zeitablauf verschieben. Zudem sind die Längen eines View-Path vom Typ des Primärobjekts abhängig. Generell ergeben sich folgende Szenarien für View-Paths:

- Es gibt zum gegebenen Zeitpunkt einen Weg vom Primärobjekt zu seiner Darstellung oder Ausführung.
- Es existieren mehrere verschiedene View-Paths. Diese sind mit geeigneten Metriken zu versehen, um automatische Entscheidungen zu erlauben.
- Es kann Primärobjekte geben, zu denen zu bestimmten Zeitpunkten keine View-Paths konstruierbar sind.

Zur sinnvollen Bestimmung der Existenz von View-Paths sollten sie sich auf bestimmte Referenzumgebungen mit jeweils festgelegten Eigenschaften aus Hard- und Software beziehen<sup>15</sup>. Einen View-Path kann man als Entscheidungsbaum interpretieren, an dessen Wurzel das Primärobjekt steht. Ein Blatt ohne weitere Verzweigungen bildet den Abschluss des Pfades in Form einer gültigen Referenzumgebung. Zur Veranschaulichung des Aufbaus der benötigten Arbeitsumgebung lässt sich ein Schichtenmodell vorstellen, wie in der Abbildung gezeigt.

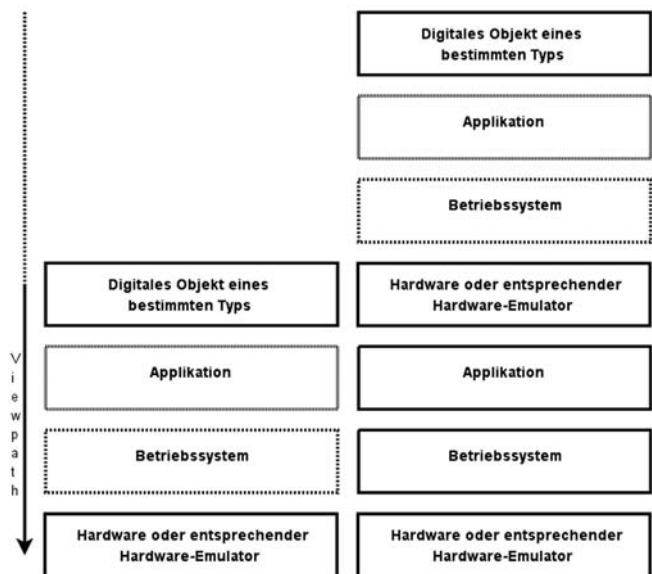


Abb. 8: View-Path in der Darstellung als Schichtenmodell. Je nach Objekt muss nicht jede Schicht durch eine Softwarekomponente repräsentiert sein.

<sup>13</sup> Derzeit wird die Einrichtung eines gemeinsamen Hardwarearchivs für das Auslesen alter Datenträger beispielsweise im Umfeld von Nestor diskutiert.

<sup>14</sup> View-Paths sind im Modell von Diessen, Raymond und Johan F. Steenbakkers: The Long-Term Preservation Study of the DNEP project – an overview of the results. Amsterdam 2002, konkrete Realisierungen des abstrakten Preservation Layer Model (PLM).

<sup>15</sup> Vergleiche dazu die Ausführungen von: van Diessen, Raymond: Preservation Requirements in a Deposit System. Amsterdam 2002.

Viele Primärobjekte lassen sich durch mehr als eine Applikation (Viewer) darstellen. Dabei können die Anzeigergebnisse in Authentizität, Komplexität oder Qualität differieren. Damit ergibt sich eine Pfadverzweigung und auf der Schicht der Applikation eine Auswahl. Ähnliches trifft für die Anforderung der Applikation nach einem Betriebssystem zu, so dass in dieser Schicht eine erneute Verzweigung auftreten kann. Die Rekursion setzt sich mit dem Betriebssystem und einer möglichen Auswahl an geeigneten Hardwareemulatoren fort. Da Referenzumgebungen weitgehend extern vorgegeben sind und nur bedingt durch den Archivbetreiber beeinflusst werden können, bestimmen sich View-Path und Referenzumgebung gegenseitig. Darüber hinaus stehen Betriebssysteme und Emulatoren über Hardwaretreiber in Abhängigkeit zueinander.

Die Modellierung des View-Paths in Schichten erfolgt nicht eng fixiert: So reduziert sich beispielsweise bei einem digitalen Primärobjekt in Form eines Programms die Zahl der Schichten. Ähnliches gilt für frühe Plattformen wie Home-Computer, deren „Betriebssysteme“ im ROM fest implementiert sind. Darüber hinaus können Schichten wiederum gestapelt sein, wenn es für einen bestimmten Emulator erforderlich wird, seinerseits eine geeignete Arbeitsumgebung herzustellen, was im rechten Teil des Schichtenmodells illustriert wird.

### 5.3 Metriken und Aufwandsabschätzungen

Eine sinnvolle Erweiterung des etwas starren Ansatzes im ursprünglichen DIAS-Preservation-Modell könnte in der Gewichtung der einzelnen View-Path-Optionen liegen, was durch eine beschreibende Metrik abgebildet werden könnte. Gerade wenn an einem Knoten mehr als eine Option zur Auswahl steht, wäre es sinnvoll:

- Präferenzen des Benutzers beispielsweise in Form der Auswahl der Applikation, des Betriebssystems oder der Referenzplattform zuzulassen,
- Gewichtungen vorzunehmen, ob beispielsweise besonderer Wert auf eine möglichst originalgetreue, authentische Darstellung oder eine besonders einfache Nutzung gelegt wird,
- Vergleiche zwischen verschiedenen Wegen zuzulassen, um die Sicherheit und Qualität der Darstellung der Primärobjekte besser abzusichern,
- den Aufwand abzuschätzen, der mit den verschiedenen View-Paths verbunden ist, um bei Bedarf eine zusätzliche ökonomische Bewertung zu erlauben.

Eine Schlussfolgerung könnten mehrdimensionale Metriken sein, die gemeinsam mit den Objektmetadaten abgelegt und durch das Archivmanagement regelmäßig aktualisiert werden. Dieses sollte in Rückkopplung mit den Archivbenutzern erfolgen, um in die Aktualisierungen Bewertungen seitens der Nutzer einfließen zu lassen, die auf dem praktischen Einsatz bestimmter View-Paths sowie ihrer Handhabbarkeit und Vollständigkeit beruhen.

Ein interessantes Problem, nicht so sehr aus der Perspektive der Primärprojekte als aus der der Archivbenutzer, ergibt sich aus unterschiedlichen Lokalisierungen. Damit ist die Anpassung der Software an bestimmte Sprachräume und Einheiten gemeint. Hierzu zählen verschiedene Währungen und ihre Zeichen, Maße oder das Format und Aussehen der Datumsangabe, die Berechnung und Anzahl von religiösen Feiertagen.

Lange Zeit wurde von kommerziellen Softwareanbietern die Landessprache in Menüführung und Benutzerdialogen als Distinktionsmerkmal verschiedener Märkte genutzt. Deshalb liegt eine triviale Form der Verzweigung im View-Path vor, wenn verschieden lokalisierte Varianten eines Programms oder Betriebssystems vorhanden sind und angeboten werden können.

Wie erörtert, kann für bestimmte Objekttypen mehr als ein View-Path vorliegen. Das erhöht zwar einerseits die Wahrscheinlichkeit des langfristig erfolgreichen Zugriffs, dieses aber bei potenziell höheren Kosten. Ausgehend vom Objekttyp und der eventuell notwendigen Applikation ergeben sich schnell weitere View-Paths, die gleichfalls realisiert werden. Ein einfaches Beispiel wären die so genannten Office-Pakete, Zusammenstellungen aus verschiedenen Applikationen: Sie können mit einer Vielfalt von Formaten umgehen – nicht nur mit den der erhaltenen Teilkomponenten, sondern darüber hinaus durch Importfilter mit einer Reihe weiterer Dateiformate.

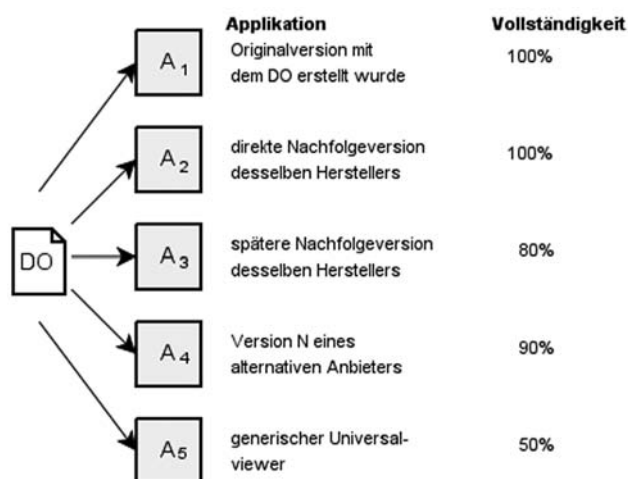


Abb. 9: Objekte eines bestimmten Typs können in vielen Fällen mit verschiedenen Betrachtern geöffnet und dargestellt werden. Dabei können jedoch Qualität und Vollständigkeit unterschiedlich ausfallen, was durch eine Gütefunktion zum Ausdruck gebracht werden könnte.

Diese Betrachtungen können helfen, eine zusätzliche Referenzplattform zu identifizieren, die nur für ein bestimmtes Objekt vorgehalten wird, das aber auf alternativen Wegen ebenfalls darstellbar ist. So muss beispielsweise zur Betrachtung eines PDFs nicht eine besonders seltene Plattform genutzt werden, wenn ein gleichwertiger Viewer auf einer mehrfach genutzten anderen Plattform ebenfalls ausführbar ist.

Schwieriger wird jedoch die Aggregation, wenn von den alternativen View-Paths nicht bekannt ist, ob sie ebenfalls zu einhundert Prozent das gefragte Objekt rekonstruieren. Ein typisches Beispiel ist der Import von Word-Dokumenten in einer anderen als der Erstellungsapplikation. An diesem Punkt könnten ebenfalls die im vorangegangenen Abschnitt vorgenommenen Überlegungen zu Benutzerrückmeldungen in Betracht gezogen werden.

Mit jedem View-Path sind bestimmte Kosten verbunden, die sich für jede Stufe abschätzen lassen. Übersteigen sie eine gewisse Schwelle, könnten ökonomische Erwägungen an die Aufnahme von Primärobjekten in das Archiv, beispielsweise die Ablehnung bestimmter Datenformate, weil deren

spätere Rekonstruktion zu hohe Kosten erwarten lässt, und das Archivmanagement geknüpft werden.

So könnte je nach Situation oder Archiv hinterfragt werden, einen View-Path aufrecht zu erhalten, wenn gute Alternativen existieren. Liegt beispielsweise ein Primärobjekt nach dem PDF 1.0 Standard vor, welches mit einem Werkzeug in einer Mac-OS 7 Nutzungsumgebung erzeugt wurde, muss deshalb diese Umgebung nicht zwingend erhalten werden. Unter folgenden Bedingungen kann ein Verzicht auf einen View-Path in Betracht gezogen werden:

- Es existiert eine ausreichende Anzahl weiterer, gleichwertiger und dabei einfacherer View-Paths.
- Wegen der guten und vollständigen Beschreibung des Formats ist es weniger aufwändig, einen Viewer für die jeweils aktuellen Host-Systeme zu migrieren als alte Nutzungsumgebungen durch Emulatoren zu sichern.
- Dieser Objekttyp ist der einzige, der eine Mac-OS 7 Umgebung verlangt.
- Es gibt kein spezielles Interesse, diese Nutzungsumgebung aufrechtzuerhalten, da das nicht im Fokus der Bibliothek liegt.

Dieses Vorgehen ließe sich auf andere View-Paths ausdehnen, die für eine Reihe von Datenformaten und Applikationen Apple- oder alternativ Microsoft-Betriebssysteme voraussetzen, wenn beispielsweise kein gesondertes Bedürfnis speziell nach dedizierten Apple-Nutzungsumgebungen existiert, weil die Art der Benutzerinteraktion, das Aussehen der grafischen Oberfläche und der Applikation in dieser von eigenständigem Interesse sind. Besonders gut lässt sich das am OpenOffice veranschaulichen, welches für etliche kommerzielle und freie Unixes wie BSD, Solaris oder Linux, Mac-OS X und selbstredend für die Windows-Betriebssysteme übersetzt wird, existiert. Hier müssen nicht alle denkbaren Zweige im Archiv vorgehalten werden. Ähnlich liegt der Fall in der trivialen Vervielfachung der View-Paths durch unterschiedliche Lokalisierungen. Durchaus anders kann der Fall liegen, wenn an anderer Stelle sekundäre Archivobjekte selbst ins Zentrum des Interesses rücken. So sind Computerkunst oder -spiele ganz eigene Zeugen ihrer Epoche.

Einfacher und damit oft günstiger zu pflegenden View-Paths kann der Vorrang vor anderen eingeräumt werden. Jedoch sind auch hier Voraussagen schwierig, und es kann die Gefahr bestehen, dass sich mit dem Wandel der Referenzumgebungen die Kostenstrukturen erneut ändern. Andererseits ließen sich Vorkehrungen treffen, dass seltenere View-Paths zumindest an spezialisierten Institutionen mit besonderem Sammelauftrag und entsprechender Finanzierung weiter betreut werden.

Die verschiedenen Strategien der Langzeitarchivierung der Emulatoren generieren verschiedenartige Kosten auf Seiten des Archivmanagements:

- Die geschachtelte Emulation sorgt für eher längere View-Paths bei geringem Migrationsbedarf. Der Aufwand entsteht beim zunehmend komplexer werdenden Zugriff auf das Primärobjekt.
- Die Migration von Emulatoren, UVMs und modularen Ansätzen erzeugt eher kurze View-Paths bei einfacherem Zugriff auf das Primärobjekt. Jedoch liegt der Aufwand im regelmäßigen Update aller benötigten Emulatoren oder ihrer Teilkomponenten.

Die erstgenannte Strategie eignet sich daher eher für Objekte mit seltenem Zugriff oder Institutionen mit kleinen,

speziell ausgebildeten Nutzerkreisen wie Archive oder Nationalbibliotheken. Die zweite Strategie ist sicherlich für vielgenutzte Objekttypen und größere Anwenderkreise vorzuziehen.

Das Zusammenfassen bestimmter View-Paths in eine gemeinsame Umgebung könnte einerseits für bestimmte Forschungsthemen und andererseits für eine Vereinfachung der View-Path-Erstellung für den Benutzerbetrieb sinnvoll sein. Für die Erforschung von Computerspielen einer bestimmten Epoche, eines Herstellers oder eines bestimmtem Genres lassen sich mehrere Applikationen in einer einzigen Containerdatei eines bestimmten Emulators aggregieren. Dieses erleichtert den Austausch, da oftmals lediglich der Container, die Konfigurationsdatei und Startanleitung des Emulators verteilt werden müssen. Das verhindert, dass sich der Endnutzer mit ihm ungewohnten Installationsvorgängen auseinandersetzen muss. Er kann sich im Sinne einer vernünftigen Arbeitsteilung auf sein wesentliches Anliegen konzentrieren.

Etwas anders gelagerte Zusammenfassungen sind ebenfalls denkbar, um auf Datensätze der öffentlichen Administration zu späteren Zeitpunkten zugreifen zu können<sup>16</sup>. Solche typischerweise in einem komprimierten Archiv zusammengefassten Daten von Erhebungen, Statistiken oder Auswertungen umfassen eine Reihe typischer Objektformate einer gewissen Epoche. Diese könnten in einem gemeinsamen Container untergebracht sein und Hilfsprogramme wie den Dokumentenausdruck nach Postscript oder PDF enthalten. Dabei können virtuelle Druckertreiber fehlende Exportfunktionen der alten Applikationen ersetzen.

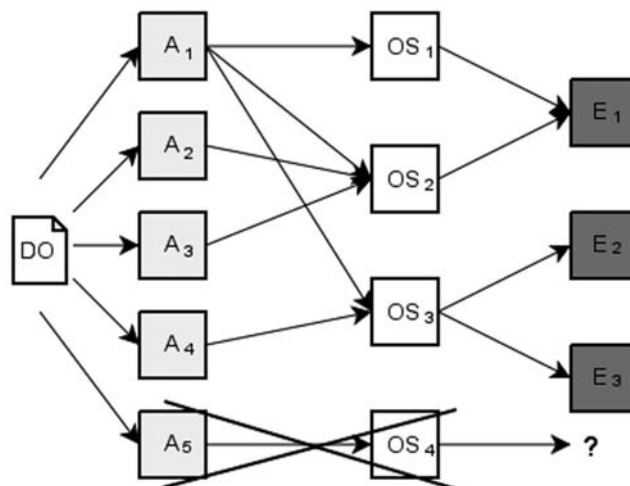


Abb. 10: Redundante View-Paths zum Rendering/Ablauflassen eines digitalen Objekts (DO). Dieses kann im Beispiel von fünf verschiedenen Applikationen (A) geöffnet werden, die auf vier verschiedenen Betriebssystemen (OS) laufen können, wobei für ein OS kein geeigneter Emulator (E) bereitsteht.

<sup>16</sup> Vorgeschlagen für die typischen Datenpakete der amerikanischen Administration: Reichherzer, Thomas und Geoffrey Brown: Quantifying software requirements for supporting archived office documents using emulation, Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries. JCDL 2006.

#### 5.4 View-Paths im technologischen Wandel

Im OAIS-Referenzmodell übernimmt das Management des digitalen Langzeitarchivs eine Reihe von Aufgaben, die sich mit dem Lifecycle-Management der Primärobjekte beschäftigen. Hierfür wurden besonders die Arbeitsprozesse für die Objektauf- und -ausgabe betrachtet. Für das langfristige Management wird insbesondere die Aufgabe des Preservation Planning interessant.

Ein zentrales Moment ist die regelmäßige Kontrolle der View-Paths mit jedem Wechsel der Referenzumgebung als Bezugsgröße. Jeder Plattformwechsel stellt neue Anforderungen für die Wiederherstellung von Umgebungen. Bei dieser Überprüfung handelt es sich um einen iterativen Prozess, der über alle registrierten Objekttypen des Archivs abläuft. Hierfür ist jeweils eine geeignete Strategie für den Übergang von einer Referenzumgebung auf eine neue zu suchen. Zudem gilt: Neu hinzukommende Dateiformate erfordern neue View-Paths.

Im Laufe der Zeit nimmt die Zahl der historischen Rechnerarchitekturen zwangsläufig zu, so dass mit jedem Wechsel der aktuellen Arbeitsumgebungen ein hoher Migrationsaufwand für die verwendeten Emulatoren anstehen kann. Diese Randbedingungen beeinflussen den Aufwand, der seitens der jeweiligen Bibliothek oder Einrichtung geleistet werden muss.

Im Zuge der zunehmenden Wahrnehmung der digitalen Langzeitarchivierung als neue Herausforderung könnten einige neue Ansätze proaktiv verfolgt werden. So wäre es beispielsweise vorstellbar, dass die Hersteller von Betriebssystemen bereits bei deren Entwicklung bestimmte Hardwaretreiber für Meta-Devices berücksichtigen. Das könnten virtuelle Geräte für die Grafik- oder Audioausgabe oder die Netzwerkschnittstelle sein. An diese Stelle könnten sogar gesetzliche Verpflichtungen treten, solche Schnittstellen oder Treiber vorzuhalten, die ähnlich wie die Pflichtabgabe von Publikationen an die Deutsche Nationalbibliothek funktionieren.

Die geschilderten Herausforderungen haben klare ökonomische Implikationen für die mit digitalen Objekten befassten Gedächtnisorganisationen. Sie werden einen Teil der notwendigen Entwicklungen der Emulatoren und Viewer leisten müssen oder diese Leistung am Markt einkaufen. Es existieren bereits einige erfolgversprechende Initiativen, die mit kooperativen Ansätzen arbeiten. Für Zusammenschlüsse wie Nedlib oder InterPARES sollte es kein (finanzielles) Problem sein, solche Herausforderungen anzunehmen. Kommen bei den angestrebten Entwicklungen offene Standards und Werkzeuge wie PRONOM oder Dioscuri<sup>17</sup> heraus, können sich einerseits die Belastungen der Einzelinstitutionen in Grenzen halten und andererseits können verbindliche Verfahren entwickelt werden, die von einer breiten Community unterstützt werden.

#### 5.5 Archivtypen im Betrieb

Je nach gefordertem View-Path werden weitere digitale Objekte benötigt. Wegen des je nach Objekttyp relativ hohen Aufwands sind Überlegungen zu treffen, wie ein View-Path realisiert wird. Gerade für häufiger angeforderte Pfade ist es denkbar, mit vorbereiteten Umgebungen zu arbeiten.

Zusammen betrachtet ergeben sich verschiedene Anforderungen an den Archivbetrieb:

- Erstellung eines Hintergrundarchivs: In diesem werden die einzelnen Elemente des View-Path dauerhaft abgelegt. Sie werden dann genauso behandelt wie Primärobjekte. An dieser Stelle kann überlegt werden, ob bestimmte Einzelobjekte, wie Emulatoren, spezifische Hilfsprogramme und Beschreibungen, in einem Paket gebündelt oder einzeln abgelegt werden.
- Betrieb eines Online-Archivs für den Direktzugriff: Für häufig nachgefragte Sekundärobjekte kann es sinnvoll sein, diese zusätzlich zum Langzeitarchiv in einem speziellen Archiv, wie einem speziellen Dateisystem einer Referenzumgebung, vorzuhalten. Das kann einerseits das Langzeitarchiv entlasten und andererseits zu einem beschleunigten Ablauf der View-Path-Generierung führen.
- Anlage eines View-Path-Caches: Für oft nachgefragte und aufwändiger zu rekonstruierende View-Paths kann die Vorhaltung vorbereiteter Arbeitsumgebungen den Aufwand für Nutzer und Archivbetreiber reduzieren. Diese Caches könnten als Teil des Online-Archivs oder direkt auf der Referenzplattform abgelegt sein.

#### 6 Softwarearchiv der Sekundärobjekte

Generell kann ein Archivnutzer nicht auf einer nackten Computerhardware ein PDF-Dokument öffnen. Er braucht hierfür mindestens eine Software zur Betrachtung, die ihrerseits nicht direkt auf der Hardware ausgeführt wird, und deren Schnittstellen für den Datei- und Bildschirmzugriff sowie die Abfrage der Benutzereingaben direkt programmiert werden. Diese Software setzt wiederum ein Betriebssystem als Intermediär voraus, welche die Ansteuerung der Ein- und Ausgabeschnittstellen der Hardware übernimmt und die Basisbenutzerinteraktion realisiert.

Je nach gewählter Ebene der Emulation werden zusätzliche Softwarekomponenten benötigt. Während bei der Emulation von Applikationen für den alternativen Zugriff auf ein bestimmtes Datenformat der Bedarf an weiterer Software entfällt, steigt dieser mit jeder tieferen Ebene im Software-Hardware-Stack. Optimalerweise laufen die emulierten Applikationen auf einer aktuellen Plattform und erlauben aus dieser heraus den direkten Zugriff auf die digitalen Objekte des entsprechenden Formats. 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. Vorstellbar wäre dieses Verfahren für statische Dateitypen wie die verschiedenen offenen und wohl dokumentierten Bildformate.

<sup>17</sup> Vergleiche die verschiedenen Projekte: Steenbakkers, Johan F.: *Setting up a Deposit System for Electronic Publications. The NEDLIB Guidelines*. Amsterdam 2000. The technical registry PRONOM <<http://www.nationalarchives.gov.uk/pronom>> und DIOSCURI Java X86 Emulator <<http://dioscuri.sourceforge.net>>.

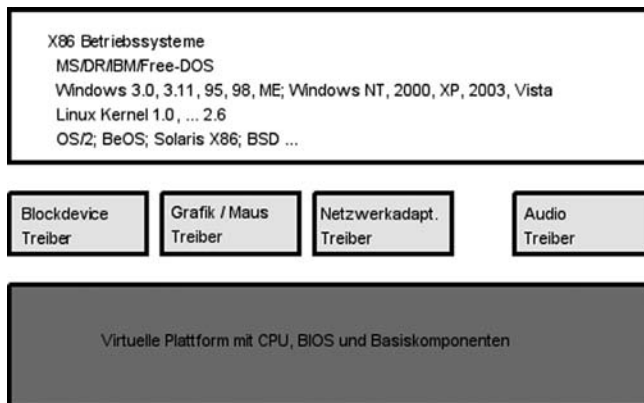


Abb. 11: Für die spätere Nutzung von X86-Applikationen ist mindestens eine Reihe von Sekundärobjekten, wie passende Betriebssysteme und ihre Hardware in einem Softwarearchiv zu hinterlegen.

Die Emulation eines Betriebssystems oder dessen Schnittstellen erlaubt theoretisch mit einigen Einschränkungen, 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. Bei der Portierung des Betriebssystememulators muss darauf geachtet werden, dass sämtliche im Softwarearchiv eingestellten Applikationen weiterhin funktionieren.

Die Hardwareemulation setzt auf einer weit unten liegenden Ebene an. Das bedeutet zwar einen sehr allgemeinen Ansatz, erfordert im Gegenzug jedoch eine ganze Reihe weiterer Komponenten. Um ein gegebenes statisches digitales Objekt tatsächlich betrachten zu können oder ein dynamisches Objekt in der Ausführung zu sehen, müssen je nach Architektur die Ebenen zwischen der emulierten Hardware und dem Objekt selbst „überbrückt“ werden. Die Brücke wird durch die weiter vorne vorgestellten View-Paths realisiert. Diese sehen je nach gewählter Strategie, „Migration-on-Dissemination“ oder Wiederherstellung einer Nutzungsumgebung, unterschiedlich aus.

Bei Nachbildung einer kompletten Hardware durch Emulation benötigt man in jedem Fall mindestens ein oder je nach Bedarf mehrere Betriebssysteme, die sich als Grundlage der darauf aufsetzenden Applikationen ausführen lassen. Das bedeutet für ein Softwarearchiv, dass neben dem Emulator für eine Plattform auch die darauf ausführbaren Betriebssysteme aufgehoben werden müssen. Das gilt ebenfalls für die darauf basierenden Applikationen, die zur Darstellung der verschiedenen Datenformate erforderlich sind. Erfolgt eine Portierung, also Migration des Hardwareemulators, muss anschließend überprüft werden, dass die gewünschten Betriebssysteme weiterhin funktionsfähig bleiben. Da die meisten Applikationen nach demselben Prinzip wie die Betriebssysteme arbeiten, sollte deren korrekte Ausführung direkt aus derjenigen der Betriebssysteme folgen.

Am besten geeignet im Sinne einer wirklichen Langzeitarchivierung sind quelloffene Implementierungen von Emulatoren. 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 nachgebildet 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.

## 7 Fazit

Der Einsatz der Langzeitarchivierungsstrategie *Emulation* erreicht, fast beliebige digitale Objekte über lange Zeiträume zu bewahren. Am weit reichendsten und Erfolg versprechendsten ist nach derzeitigem Stand die Emulation von Hardware, die es erlaubt, alte Nutzungsumgebungen wiederzubeleben, in denen digitale Primärobjekte in ihrer unveränderten Form angesehen werden oder ablaufen können. Dabei geht es nur in wenigen Anwendungsfällen darum, historische Umgebungen als reguläre Arbeitsplattform zu erhalten, in denen zukünftige Benutzer sich ebenso bewegen wie die Anwender der jeweiligen Periode. Der Hauptfokus liegt klar auf der „Sichtbarmachung“ von Primärobjekten und allen dazu notwendigen Werkzeugen und Informationen. Damit dieses langfristig gelingen kann, ist eine Reihe von Erfolgsbedingungen zu berücksichtigen – Emulation zählt zu den aufwändigeren Strategien, die nicht umsonst zu haben ist.

So muss für einen erfolgreichen Zugriff auf ein bestimmtes digitales Objekt mindestens ein View-Path konstruierbar sein. Dieser beginnt normalerweise bei der Erstellungs-, alternativ einer funktional vergleichbaren Applikation oder der Ablaufumgebung des Objekts und endet auf der aktuellen Plattform, mit der ein Archivnutzer interagiert. Das hat im Gegensatz zur Migration den Vorteil, dass das digitale Objekt selbst nicht verändert werden muss und damit sowohl integer als auch authentisch bleibt. Hieraus folgen unmittelbar zwei weitere Einsatzgebiete: Mehrfach in die jeweils aktuelle Referenzumgebung migrierte Objekte lassen sich durch einen Vergleich mit ihrem in einer emulierten Nutzungsumgebung ablaufenden Original bei Bedarf verifizieren. Zudem bekommen „Softwarearchäologen“ einen digitalen Werkzeugkasten an die Hand, mit dem sie außerhalb von modernen digitalen Langzeitarchiven aufbewahrte Objekte analysieren können. Zur Erhaltung der View-Paths wird ein Softwarearchiv erforderlich, das neben der Software eine Reihe wichtiger Metadaten wie Handbücher, Bedienungsanleitungen und Lizenzschlüssel vorhalten muss.

Trotzdem kann die Emulation die Migration nicht ersetzen, sondern sie stattdessen sinnvoll ergänzen: Während sich Migrationsprozesse etlicher statischer Objekttypen weitgehend automatisiert durchführen lassen, erfordert Emulation typischerweise einen hohen Anteil an Benutzerinteraktion. Darüber hinaus besteht in der Emulatormigration eine denkbare Strategie der langfristigen Nutzbarkeit dieser Applikationen.

Das um dynamische digitale Objekte erweiterte Objektverständnis bedeutet nicht, dass digitale Langzeitarchive neu erfunden werden müssen, sondern dass eigentlich interessierende Primärobjekte bestimmte Sekundärobjekte erforderlich werden lassen. Für diese Bestimmungprozesse spielen einerseits geeignete Format-Registries und andererseits Metadatenschemata zur Aufnahme der zu-

sätzlich notwendigen Informationen zur Behandlung des jeweiligen Objekts eine Rolle.

Inzwischen existieren einige Metadatenstandards, die sich anbieten, zusätzlich zu traditionellen Archivalien dynamische digitale Objekte zu klassifizieren. Es kann jedoch erforderlich werden, Standards um ausführliche Beschreibungen oder Installationsanleitungen zu erweitern.

Vertrauenswürdige digitale Langzeitarchive schaffen neuartige Anforderungen für Gedächtnisorganisationen. Die Tätigkeiten und Fähigkeiten digitaler Archivare weichen zwangsläufig von traditionellen Aufgabenbeschreibungen ab: „New Skills Call for New Jobs“<sup>18</sup>. Ähnliches betrifft den Archivbetrieb selbst: Digitale Langzeitarchive basieren optimalerweise auf offenen Verfahren, die auf gemeinsam definierten und entwickelten Standards beruhen. Die Erfahrung zeigt, dass lokale und nationale Lösungen der Größe der Aufgabe nicht gewachsen sind. Bezogen auf historische digitale Objekte muss man heute schon feststellen, dass für proprietäre Entwicklungen teilweise ein hoher Preis in Form hohen Aufwands beim Zugriff auf oder des Verlusts von Informationen entrichtet werden muss. Große Probleme sowohl der Primär- als auch Sekundärobjekte kann das im Augenblick vielfach diskutierte Digital Rights Management schaffen. Dieses dient in der jeweils „aktiven“ Phase eines

Objekts dem Schutz vor unauthorisiertem Zugriff oder der Wahrung der Rechte des ursprünglichen Objekterstellers. Dieses Rechtemanagement oder dieser Kopierschutz steht einem Archivierungsanliegen entgegen: Digitale Langzeitarchive sind darauf angewiesen, Objekte kopieren zu können. Zudem sollte der Zugriff auf Objekte für spätere Nutzer nicht an Systeme gebunden sein, die zu diesem Zeitpunkt mit hoher Wahrscheinlichkeit nicht mehr verfügbar sind. An dieser Stelle ist der Gesetzgeber gefordert, geeignete Rahmenbedingungen zu schaffen.

**Anschrift des Autors:**

Dirk von Suchodoletz  
Rechenzentrum der Universität  
H.-Herder-Str. 10  
D-79104 Freiburg i.Br.  
E-Mail: dirk.von.suchodoletz@rz.uni-freiburg.de

<sup>18</sup> van der Werf, Titia: Experience of the National Library of the Netherlands. The State of Digital Preservation. An International Perspective, Conference Proceedings. Documentation Abstracts, Inc. Washington 2002.