

■ DAS SOFTWAREARCHIV – EINE ERFOLGSBEDINGUNG FÜR DIE LANGZEITARCHIVIERUNG DIGITALER OBJEKTE

von Dirk von Suchodoletz

Gedächtnisorganisationen wie Bibliotheken, Archive und Museen müssen sich in immer größerem Umfang mit dem digitalen Erbe auseinandersetzen. Die Bedeutung und der Anteil von digitalen Objekten am Gesamtumfang der jeweiligen Sammlung werden in den kommenden Jahren zunehmen. Diese neuartigen Artefakte sind, anders als traditionelle Medien wie Papier, Leinwand oder Microfiche, nicht mehr trivial ohne ihren technischen Erstellungskontext zugreif- oder nutzbar. Erhält beispielsweise eine Bibliothek den Nachlass eines wichtigen Lyrikers auf Disketten mit Word Perfect Dateien, so genügt es nicht, dass diese bitgetreu fehlerfrei ins Archiv kopiert werden können. Ohne die Erstellungssaplikation oder eine kompatible, die mit dem Format der Dateien umgehen kann, sind die digitalen Artefakte wertlos (Abb. 1). Das Vorliegen einer geeigneten Applikation muss je nach Alter der Objekte nicht genügen, da sie vielleicht nicht mehr auf heutigen Systemen installiert und ausgeführt werden kann. Hier müssen Bewahrungsstrategien ansetzen, welche die besonderen Eigenschaften der digitalen Objekte berücksichtigen. Das bedeutet, dass wenn Informationen aus digitalen Objekten nicht für immer verloren gehen sollen, sie entweder mit der technischen Entwicklung mitgeführt werden müssen oder ihr ursprünglicher Erstellungskontext oder ein geeignetes Äquivalent erhalten bleiben.

Dieser Artikel beschreibt einen aus Sicht des Autors bisher noch zu wenig beachteten Aspekt digitaler Archivierung und beschäftigt sich mit der Bedeutung von Softwarearchiven von Sekundärobjekten. Er basiert auf der Dissertation „Funktionale Langzeitarchivierung digitaler Objekte – Erfolgsbedingungen des Einsatzes von Emulationsstrategien“, die im Rahmen des PLANETS Projects (Planets 2010) an der Universität Freiburg angefertigt und 2009 mit einem Preis des VFI ausgezeichnet wurde (Suchodoletz 2009).

Strategien in der digitalen Langzeitarchivierung

Abgesehen von der reinen Medienkonservierung spielen in der digitalen Langzeitarchivierung im Wesentlichen zwei Archivierungsstrategien eine



Abbildung 1: Word Perfect war Ende der 1980er und Anfang der 1990er eine sehr populäre Textverarbeitungssoftware. Um damit erstellte Artefakte in der Originalversion zu betrachten, muss die Software und eine geeignete Umgebung mitarchiviert werden.

Rolle: Migration und Emulation (Neuroth et.al. 2009). Sie sollen den verlässlichen Zugriff auf digitale Objekte auch nach vielen Jahren sicherstellen. Die den meisten Gedächtnisorganisationen vertrautere und derzeit dominierende Methode ist die Migration. Sie transportiert schrittweise ein digitales Objekt durch die sich mit der Zeit wandelnden digitalen Arbeitsumgebungen, die aus sich permanent ändernden Software-Hardware-Konfigurationen bestehen. Migration stellt auf diese Weise sicher, dass Archivnutzer das sie interessierende Artefakt innerhalb der jeweils bekannten Computerumgebung nutzen können und die jeweils aktuellen Such- und Indizierungsverfahren funktionieren. Am Objekt und seiner Struktur neh-

men Migrationsprozesse hierfür notwendige Änderungen vor. Erst diese Anpassungen ermöglichen es, auf den jeweils aktuellen Computersystemen auf digitale Objekte zuzugreifen. Migration weist jedoch zwei gravierende Nachteile auf: Sie schränkt die Menge der später zugreifbaren Objekttypen unnötig ein, da sie sich nur mit Aufwand auf dynamische digitale Objekte, wie beispielsweise Computerspiele oder wissenschaftliche Spezialsoftware, anwenden lässt. Ein weiteres Problem entsteht aus den zwangsläufigen Veränderungen des Objekts: So wird dessen Authentizität fragwürdig und untergräbt potenziell die Vertrauenswürdigkeit eines Langzeitarchivs.

Deshalb sollte zusätzlich sichergestellt werden, dass ein Zugriff auf das Objekt in seinem Originalzustand erfolgen kann. Zudem sollte die unnötige Einschränkung der archivierbaren Objekttypen reduziert werden. An diesem Punkt gewinnt Emulation an Bedeutung, da sie einen komplett anderen Ansatz als Migration verwendet. Sie operiert nicht auf dem Objekt selbst, sondern stellt die ursprünglichen digitalen Umgebungen insoweit wieder her, um einen unveränderten Zugriff zu erlauben. Emulatoren übernehmen dabei die Aufgabe, obsolete Komponenten, wie alte Hardware, Betriebssystem-APIs oder Applikationen geeignet nachzubilden. Die langjährige Forschung in diesem Gebiet (u.a. Hoeven, Verdegem 2006) hat jedoch gezeigt, dass ein Ansatz auf der elementarsten Ebene, d.h. der Ersatz alter Computerhardware, am erfolgversprechendsten ist. Dieses impliziert, dass neben dem eigentlich interessierenden digitalen Objekt weitere Softwarekomponenten aufgehoben werden müssen, um eine spätere erfolgreiche Wiederherstellung zu leisten. Diese Aufgabe muss ein spezielles Softwarearchiv übernehmen. Zur Sicherstellung der Vollständigkeit des Archivs wird eine Formalisierung von Zugriffsabläufen, die vom interessierenden Objekt startend auf die jeweils gültige Arbeitsumgebung des Archivnutzers zielen, benötigt. Emulatoren, denen die Aufgabe der Brückenfunktion zwischen dem technischen Gestern und Morgen zukommt, sind als Software wie alle digitalen Objekte von Obsoleszenz betroffen und müssen deshalb geeignet ausgewählt und behandelt werden (Suchodoletz 2009).

Zugriff auf digitale Artefakte

Digitale Objekte bedürfen, wie eingangs erwähnt, eines geeigneten Kontextes, damit auf sie zugegriffen werden kann. So lässt sich beispielsweise eine Videodatei nicht ohne entsprechende Software, die üblicherweise ein bestimmtes Betriebssystem voraussetzt, abspielen. Das Betriebssystem muss über Schnittstellen zur Bildausgabe in passender Auflösung und

Farbtiefe sowie die Fähigkeit zur Audioausgabe verfügen. Das wiederum setzt geeignete Hardware oder deren bezogen auf die Anforderungen vollständige Nachbildung durch Emulation voraus. Neuere Betriebssysteme nutzen sogenannte Gerätetreiber, um die Hardware korrekt anzusprechen, also Bilder via Grafikkarte auf einen Monitor ausgeben zu können oder analoge Audiosignale via Soundkarte zu generieren.

Die notwendigen Software- und Hardwarekomponenten zur Sichtbarmachung oder Ausführung von verschiedenen, im Langzeitarchiv enthaltenen Objekttypen sind im Moment der Archivaufnahme der digitalen Artefakte festzulegen. An dieser Stelle werden diverse Arbeitsabläufe notwendig, die es erst erlauben, dass ein späterer Archivnutzer tatsächlich mit dem gewünschten Objekt umgehen kann. Mit zunehmendem Zeitabstand zur Erstellung des Objekts sinkt die Wahrscheinlichkeit, dass dieses direkt in der digitalen Arbeitsumgebung des Nutzers ausgeführt oder betrachtet werden kann. Daraus ergibt sich eine zentrale Aufgabe von Betreibern digitaler Langzeitarchive: Ihre Zuständigkeit endet nicht mit der erfolgreichen Bewahrung und bitgetreuen Fortschreibung der Archivobjekte, sondern sie müssen zusätzlich die Nutzbarkeit zu jedem Zeitpunkt sicherstellen. Die Arbeitsabläufe für den Zugriff auf verschiedene Objekttypen können sich unterscheiden (Abb. 2).

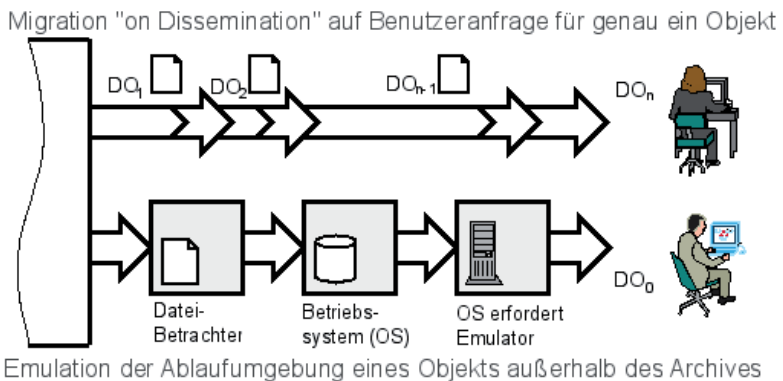


Abbildung 2: Digitale Objekte benötigen einen geeigneten Kontext zur Betrachtung oder Ausführung, der vom Archivbetreiber zur Verfügung gestellt werden sollte.

Migration-on-Dissemination – das Primärobjekt wird, je nach Typ auch in mehreren hintereinander geschalteten Stufen, durch wiederholt ablaufende Prozesse in eine für den Nutzer betrachtbare Form gebracht. Dieser

Vorgang kann durchaus Emulationsprozesse involvieren, um ältere Software zum Zwecke der Formatmigration auszuführen. Er kann je nach Tiefe und Aufwand der einzelnen Schritte längere Zeit in Anspruch nehmen und unter Umständen direkte Nutzerinteraktionen erfordern. Das Ziel der Migration ist ein Format, welches von einem in der jeweiligen Referenzumgebung verfügbaren Viewer angezeigt oder abgespielt werden kann. Das Objekt wird in dieser Prozedur zwangsläufig verändert und kann damit unter Umständen bestimmten, beispielsweise juristischen, Ansprüchen nicht mehr genügen.

Wiederherstellung einer Nutzungsumgebung – für dynamische Objekttypen (Programme, insbesondere auch mit direkter Benutzerinteraktion) verbleibt als Darstellungsoption lediglich die Wiederherstellung der ursprünglichen oder einer äquivalenten Umgebung. Dieser Vorgang hängt wiederum vom konkreten Objekttyp und seinen spezifischen Anforderungen ab. Die involvierten Prozesse werden eine bestimmte Anzahl zusätzlicher, sekundärer Objekte benötigen. Darüber hinaus stellt sich die Frage des geeigneten Transports des Objekts in die wiederhergestellte Umgebung, da emulierte Umgebungen und das Benutzersystem nicht über gemeinsame einfache Schnittstellen zum Datenaustausch verfügen müssen.

Formalisierung des Objektzugriffs

Die Wiederherstellung von Hardware-Softwareumgebungen oder entsprechender Äquivalente lässt sich am besten durch sogenannte „View-Paths“, Wege ausgehend vom Objekt des Interesses bis in die aktuelle Arbeitsumgebung des Betrachters oder Anwenders, veranschaulichen und formalisieren (Neuroth et.al. 2009). Während der Ausgangspunkt des View-Paths durch das interessierende Artefakt fixiert ist, wird sich, extern determiniert durch den technologischen Fortschritt und die sukzessive Obsoleszenz vorhandener Rechnerplattformen, der Endpunkt des View-Paths 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, Computerplattformen in der jeweiligen Gedächtnisorganisation für den Objektzugriff durch Archivare und Nutzer, beziehen. Einen View-Path kann man so als Entscheidungsbaum interpretieren, an dessen Wurzel das Artefakt steht. Ein Blatt ohne weitere Verzweigungen bildet den Abschluss des Pfades in Form einer gültigen Referenzumgebung (Abb. 3).

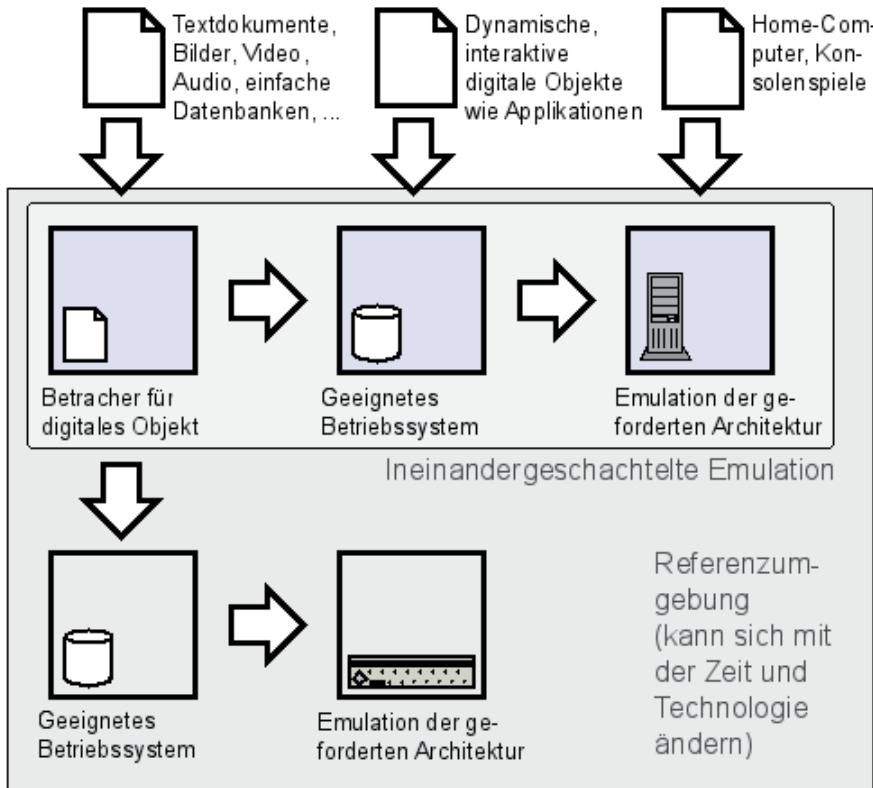


Abbildung 3: View-Paths können sich in Länge und Aufwand je nach digitalem Objekt und Langzeitstrategie für die Emulatoren in der Länge wesentlich unterscheiden.

Viele digitale Objekte lassen sich durch mehr als eine Applikation (Viewer) darstellen. Dabei können die Anzeigeergebnisse in Authentizität, Komplexität oder Qualität differieren. Das ergibt eine Pfadverzweigung

auf der Applikationsebene. Ähnliches trifft für die Anforderung der Applikation nach einem Betriebssystem zu, so dass in dieser Ebene 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. Diese Bedingungen sind für ein Softwarearchiv von essentieller Bedeutung.

Die Konsequenzen hieraus für Migrations- und Emulationsstrategien ähneln sich: So kann es für migrierte Objekte sinnvoll sein, neben dem Default-View-Path zu einem jeweils aktuellen Viewer auch den Emulations-View-Path zur regelmäßigen oder stichprobenartigen Kontrolle vorzuhalten. Um dabei den Aufwand für häufig auftretende View-Paths zu verringern, lassen sich Überlegungen zur Aggregation oder zum Caching vornehmen.

Softwarearchiv der Sekundärobjekte

Zu den Erfolgsbedingungen digitaler Langzeitarchivierungsstrategien für die verschiedensten Typen digitaler Objekte zählen nicht nur die Primärwerkzeuge, wie Emulatoren oder Migrationsprogramme. Wie eingangs motiviert sind weitere Softwarekomponenten erforderlich. Da sowohl die für den Anwender interessanten Daten und relevante Software als auch die zur Ansicht notwendigen Hilfsprogramme, Emulatoren, Betriebssysteme digitale Objekte sind, sollen sie wie folgt unterschieden werden: Ein Primärobjekt oder digitales Artefakt gehört zu den primär interessierenden Objekten eines Archivbenutzers, welches er in irgendeiner Form betrachten oder nutzen will. Für die Wiederherstellung einer bestimmten Hardware-Softwareumgebung werden zusätzliche Komponenten benötigt, die sich aus dem jeweiligen View-Path ergeben. Sie können je nach dessen Wahl variieren. Diese Software in den verschiedenen Ausformungen wird als Sekundär- oder Hilfsobjekt bezeichnet. Ein solches Objekt ist für den Archivbenutzer nicht primär von Interesse, wird aber gebraucht, um überhaupt mit dem digitalen Artefakt umgehen zu können. Es muss zusätzlich zum Primärobjekt geeignet archiviert werden.

Das Softwarearchiv selbst kann wieder als Bestandteil eines größeren Archivs nach dem OAIS-Modell (Neuroth et.al. 2009), einem abstrakten Standard für digitale Langzeitarchive, angesehen werden. Für die Aufbe-

wahrung der Emulatoren, der Betriebssysteme, Applikationen und Hilfsprogramme gelten damit weitgehend die identischen Regeln wie für die eigentlichen digitalen Zielobjekte. Trotzdem kann es von Interesse sein, diese Daten in direkt zugreifbarer Weise oder auch in speziell aufbereiteter Form vorzuhalten und über Organisationsgrenzen hinweg abzugleichen.

Ein zusätzlicher Problemkreis ergibt sich aus der Art der Sekundärobjekte. Anders als bei den meisten Zielobjekten werden im Laufe der Zeit Änderungen oder Ergänzungen notwendig, die im Archiv berücksichtigt werden sollten. Digitale Objekte werden, wie ihre analogen Pendant, häufig mit einschränkenden Nutzungsrechten versehen. Für eine langfristige Verfügbarkeit von als notwendig erachteten Sekundärobjekten sind diese häufig erst von ihrem Auslieferungs- oder Installationsdatenträger zu lösen und in eine langzeitarchivierbare Form zu überführen. Hierfür sind virtuelle Datenträger und mögliche Formen der Ablage von Sekundärobjekten zu betrachten. Ebenso spielen in diesem Bereich rechtliche Vorgaben und Rahmenbedingungen eine Rolle (siehe unten).

View-Paths und Softwarearchiv

Der Umfang an Sekundärobjekten im Softwarearchiv wird durch die View-Paths aller Primärobjekttypen bestimmt. Neben den Objekttypen hat die Art der Archivierungsstrategie und bei der Verwendung von Emulation deren Ansatzpunkt im Software-Hardware-Stack einen wesentlichen Einfluss. So kann die Emulation von Applikationen, wie beispielsweise der Einsatz von Open Office zum Lesen von MS-Word-Dokumenten einen alternativen Zugriff auf ein bestimmtes Datenformat bereitstellen, ohne bezogen auf die aktuelle Nutzerplattform weitere Softwareanforderungen zu stellen. Solange es gelingt, die entsprechenden Applikationen 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 wohldokumentierten Bildformate.

Der Aufwand steigt indes, wenn beispielsweise ein altes AmiPro in einem Windows 3.1 ausgeführt werden soll, welches seinerseits in einem X86-Emulator auf der Nutzerplattform läuft. Die Hardwareemulation setzt auf einer weit unten liegenden Ebene an. 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

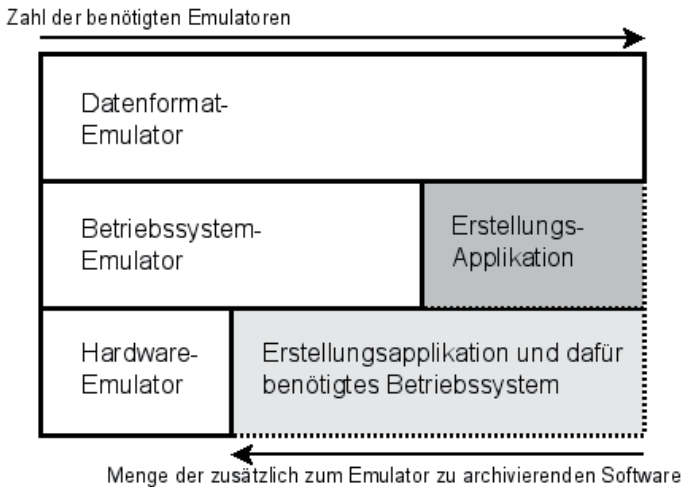


Abbildung 4: In Abhängigkeit von der genutzten Archivierungsstrategie und dem Ansatzpunkt von Emulation werden unterschiedlich viele zusätzliche Softwarekomponenten zur Betrachtung oder Ausführung eines digitalen Artefakts benötigt.

zu sehen, müssen je nach Architektur die Ebenen zwischen der emulierten Hardware und dem Objekt selbst „überbrückt“ werden (Abb. 4).

Bei Nachbildung einer kompletten Hardware durch Emulation benötigt man in jedem Fall mindestens eines 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 ablauffähig bleiben. Da die meisten Applikationen nach demselben Prinzip wie die Betriebssysteme arbeiten, sollte deren Funktionsfähigkeit direkt aus derjenigen der Betriebssysteme folgen.

Inhalte des Softwarearchivs

Wie eingangs ausgeführt, benötigt jedes digitale Objekt einen Betrachter oder eine passende Umgebung, mit dem es geöffnet oder in der es ausge-

führt werden kann. Daraus folgt, dass man zum einen diese Umgebung geeignet rekonstruieren muss, um dann in dieser die gewünschten Daten anzusehen oder in selteneren Fällen zu bearbeiten.

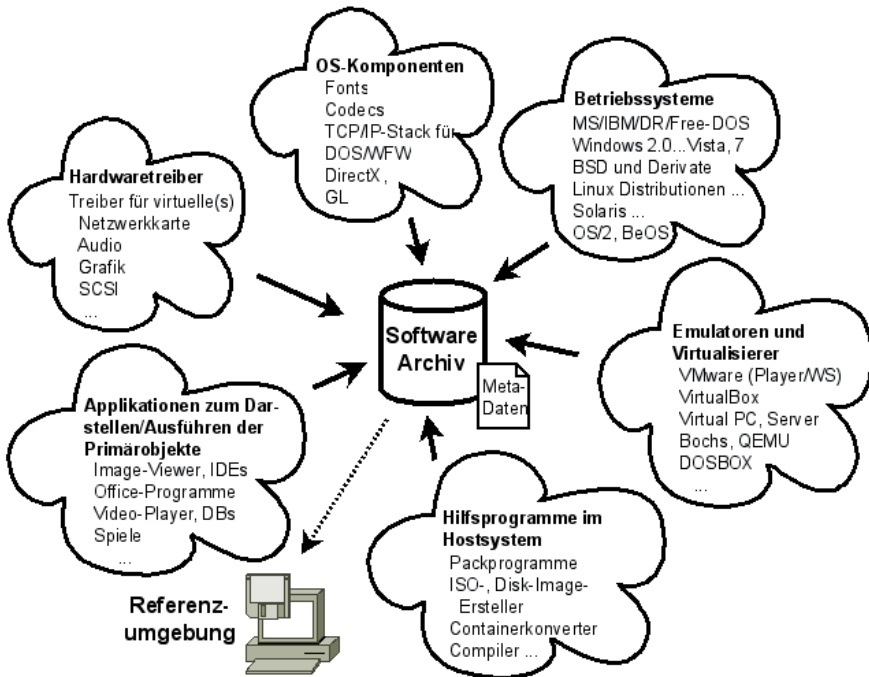


Abbildung 5: Das Softwarearchiv der Sekundärobjekte muss eine Reihe von zusätzlichen Komponenten beinhalten, um einen späteren Zugriff auf Primärobjekte zu ermöglichen.

Andererseits wird sich je nach Erstellungsdatum des Objekts die damalige Erstellungs- oder Nutzungsumgebung dramatisch von der jeweils aktuellen unterscheiden. In der Zwischenzeit haben sich mit einiger Wahrscheinlichkeit die Konzepte des Datenaustausches verändert. Hier ist nun dafür zu sorgen, dass die Primärobjekte 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.

Das Archiv (Abb. 5) muss die verschiedenen Typen von Emulatoren umfassen, mit deren Hilfe bestimmte Datenformate gelesen werden können oder, breiter angelegt, die Wiederherstellung einer Rechnerarchitektur für

bestimmte Umgebungen erfolgen kann. Zudem sind die Betriebssysteme zu archivieren, die je nach Rechnerplattform einen Teil der Nutzungsumgebung ausmachen. Sie bilden für die überwiegende Anzahl von Rechnerplattformen das vermittelnde Glied zwischen den Applikationen sowie der Hardware und stellen eine Reihe von Standardschnittstellen zum Benutzer und zu den verschiedenen Applikationen bereit. Insbesondere für modernere Architekturen werden Treiber notwendig, da sie dem Betriebssystem überhaupt erst erlauben, eine bestimmten Hardware anzusprechen, wenn kein direkter Zugriff via BIOS-API realisiert werden kann.

Ein zentrales Element des Archivs sind Applikationen, mit denen die verschiedenen digitalen Objekte erstellt wurden. Sie treten dem Benutzer entweder in direkter Form über die Benutzerschnittstellen der jeweiligen Rechnerplattform entgegen oder werden mittelbar durch Benutzerinteraktion aufgerufen. Applikationen erscheinen an einer definierten Stelle des View-Paths. Die letztlichen Anforderungen werden durch das jeweilige Primärobjekt bestimmt. Eine besondere Gruppe von Anwendungsprogrammen sind Skriptsprachen oder Runtime Environments. Sie bilden eine für den Endanwender typischerweise unsichtbare Brückenfunktion zwischen dem Betriebssystem und Applikationen, die wegen der eingeschobenen Abstraktionsschicht auf verschiedenen Computerarchitekturen ablauffähig sind. Darüberhinaus sind unter Umständen notwendige Erweiterungen einer Applikation, wie bestimmte Funktionsbibliotheken, Codecs oder Schriftartenpakete zur Darstellung zu ermitteln und bei Bedarf hinzuzufügen. Weiterhin sind alle Hilfsprogramme zu sammeln, welche den Betrieb der Emulatoren vereinfachen oder überhaupt erst ermöglichen. Hierzu zählen beispielsweise Programme, die direkt mit dem jeweiligen Containerformat eines Emulators umgehen können. Ebenfalls in diese Klasse fallen (De-)Kompressionsprogramme, die verschiedene Pack- und Archivtypen beherrschen, in denen Software ursprünglich bereitgestellt wurde oder mit denen Primärobjekte zur Platzersparnis komprimiert wurden.

Rechtliche Implikationen

Viele digitale Artefakte sind das Ergebnis eines menschlichen Schöpfungsprozesses und damit wie ihre analogen Pendanten im Augenblick der Erstellung mit bestimmten Schutzrechten ausgestattet (Beger 2005). Diese Rechte gelten üblicherweise für einen bestimmten begrenzten Zeitraum, der je nach nationalem Rechtssystem auf sehr viele Jahre angelegt ist. Sie überschreiten damit weit den intendierten Lebenszyklus eines durch-

schnittlichen digitalen Objekts. Das kann bedeuten, dass wenn bis zum Zeitpunkt des Ablaufs der Rechte gewartet wird, ein für einen Zugriff seitens späterer Nutzer keine Technik oder kein Wissen mehr existiert. Dieses Dilemma lässt sich nicht trivial auflösen: Entweder man fordert seitens des Herstellers zum Veröffentlichungszeitpunkt eine ungeschützte Kopie, die an designierte Gedächtnisorganisationen abzuliefern ist, oder man wartet bis zum Ablauf einer gewissen Frist der Primärvermarktung. Dieser Zeitpunkt ist jedoch schwer zu bestimmen und es besteht die Gefahr, dass die Herstellerfirma schon aus dem Marktgeschehen ausgeschieden ist.

Da mit der Objekterstellung vielfach kommerzielle Interessen verfolgt werden, wie die Erstellung von Anwendungsprogrammen und ihr anschließender Verkauf an Endnutzer, möchte der Hersteller seine spätere Vergütung sicherstellen. Dies geschieht typischerweise mittels Kopierschutzverfahren oder durch eine später notwendige „Freischaltung“ des Objekts. Es existieren zwei wesentliche Arten von Schutzmechanismen: Vorrichtungen auf Ebene des Datenträgers und solche am digitalen Objekt selbst.

Die aus Sicht der Objektersteller zu schützenden Objekte werden in Abhängigkeit vom jeweiligen Computersystem auf verschiedenen Datenträgern zur Verfügung gestellt. Hierbei handelt es sich gewöhnlich um Installations- oder Laufzeitdatenträger: Erstere werden üblicherweise einmal benötigt, um eine bestimmte Software auf einen Rechner aufzuspielen. Letztere werden während des Ablaufs der jeweiligen Applikation auf ihr Vorhandensein überprüft und es werden von ihnen diverse Laufzeitdaten nachgeladen. Bei beiden handelt es sich um Verteilmedien, mit denen Hersteller ihre Produkte in den Umlauf bringen. Sie machen aus der körperlosen Software ein klassisches „verkauftes“ physisches Objekt. Schutzvorrichtungen an Datenträgern sollen verhindern, dass das Medium problemlos digital kopiert werden kann. Hierbei kommen oft fragwürdige Verfahren zum Einsatz, die sich abseits von definierten Standards bewegen wie kopiergeschützte Audio-CDs, die nicht mehr dem CD-Standard entsprechen. Ähnliches versuchen Computerspielerhersteller mit scheinbar defekten oder speziellen leeren Dateien auf Datenträgern, auf deren Existenz geprüft wird. Teilweise nutzen Kopierschutzverfahren nichtstandardkonforme Spezialfunktionen der Lesehardware. Damit entsteht das Problem, dass Primärobjekte unter Umständen nicht adäquat gesichert und in ein digitales Langzeitarchiv überführt werden können. Viele Datenträger mit ihren Schutzvorrichtungen haben den Charakter relativ kurzlebiger Konsumprodukte, die teilweise bewusst nicht auf eine lange Haltbarkeit angelegt sind. Hier wäre zu klären, wie die Rechte zwischen Konsumenten, Herstellern und Gedächtnisorganisationen auszutarieren sind.

Der Schutz an digitalen Objekten ist insofern von Interesse, als dass es nur für wenige Schutzmechanismen gelingt, die notwendigen Komponenten mitzuarchivieren. Da die meisten modernen Verfahren mit online verifizierten und aktualisierten Schlüsseln arbeiten, eignen sich diese nicht für eine Langzeitarchivierung, da im Augenblick der Nichterreichbarkeit oder Abschaltung des Schlüsselservers das Artefakt unbrauchbar werden kann. Deshalb sollten Primärobjekte generell ohne einen Zugriffsschutz ins Langzeitarchiv eingestellt werden. Neben der Freiheit von allen Online-Schutzkomponenten müssen die Metadaten die notwendigen Offline-Lizenzschlüssel enthalten, um das Produkt installieren und aktivieren zu können. Hier sind die neuesten Entwicklungen zu beobachten, die mit zeitlich beschränkten Lizenz- oder Abonnementsmodellen arbeiten.

Archivmanagement

Das Softwarearchiv selbst kann wieder als Bestandteil eines größeren Archivs beispielsweise nach dem OAIS-Modell angesehen werden. Dabei liegt der Umfang der Sekundärobjekte wie oben erwähnt für die meisten Archive deutlich unter der Zahl ihrer Primärobjekte. Deshalb kann man bei Nutzung eines gemeinsamen Speichersystems die Datenmenge des Softwarearchivs in den meisten Fällen mengenmäßig vernachlässigen. Dennoch steigt die Menge der zu verwaltenden Daten und Metadaten im Zeitverlauf unaufhörlich an. So ist für jeden neuen Typ von Primärobjekt zu bestimmen, ob bereits ein gültiger View-Path vorliegt oder ob Komponenten ergänzt werden müssen.

Ein weiterer Problembereich ergibt sich aus der Art der Sekundärobjekte. Im Laufe der Zeit werden Änderungen oder Ergänzungen notwendig, die im Archiv berücksichtigt werden sollten. Emulatoren und ihre Komponenten sind je nach gewählter Strategie mit der Entwicklung der Referenzumgebung mitzuführen und entsprechend zu aktualisieren.

Anders als viele Primärobjekte werden die Sekundärobjekte häufig nachgefragter View-Paths regelmäßig abgerufen werden. Diese Zugriffscharakteristiken werden sich über ähnliche Gedächtnisorganisationen hinweg nicht stark unterscheiden. Deshalb ist es sicherlich zielführend, Archivmodelle mit kooperativen Anteilen zu diskutieren, welche sich mit der Schaffung einer hohen, auch räumlichen Redundanz der Daten befassen und eine geeignete Sicherung der Objekte und Vergabe von Zugriffsrechten zu erreichen (Abb. 6).

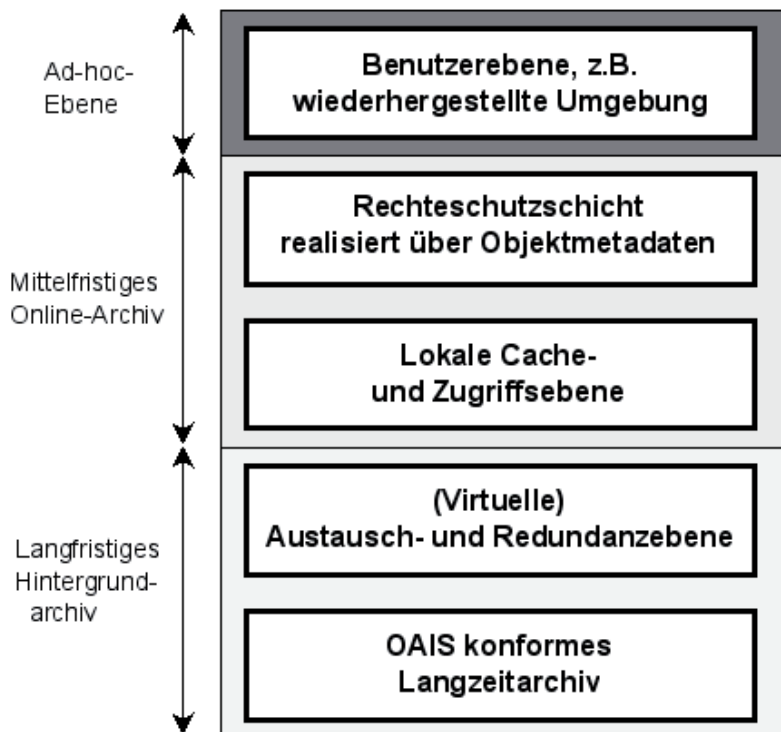


Abbildung 6: Zur Sicherstellung eines Rechtemanagements ausserhalb der enthaltenen Objekte und Ermöglichen einer geeigneten Replikation des Archivs bietet sich ein (verteiltes) Schichtenmodell an.

Ebenso sollte es einen zügigen und komfortablen Zugriff der Archivnutzer auf oft benutzte View-Paths geben, die sich durch sogenannte View-Path-Caches – vorbereitete Nutzungsumgebungen – organisieren lassen. Dabei ist eine sinnvolle und kosteneffektive Verteilung des Aufwands zur Archivpflege zu erwägen. Ein solches Archiv könnte als Schichtenmodell (Abbildung) organisiert werden und dabei verschiedene Speicher- und Rechteebenen realisieren. Das langfristige Hintergrundarchiv wird bei den meisten Institutionen das vorhandene OAIS-konforme Langzeitarchiv der Primärdaten sein. Oberhalb dieser Ebene könnte sich eine Austauschschicht befinden, über die zwischen räumlich getrennten Einheiten derselben Institution oder über Institutionen hinweg Archivobjekte verteilt abgelegt werden. Diese sollte aus Redundanzgründen möglichst nicht durch einen Rechtesschutz eingegrenzt werden. Das mittelfristige Online-Archiv

befindet sich deutlich in Benutzernähe. Über dieses greifen Archivnutzer oder das entsprechende Archivpersonal auf Sekundärobjekte zur Wiederherstellung von Umgebungen zu. Auf dieser Ebene wird ein Rechtemangel sinnvollerweise mittels Metadaten realisiert. Für häufig nachgefragte View-Paths sollte ein Cache angelegt werden. Das entlastet das Langzeitararchiv von wiederholten Anfragen. Die Ad-hoc-Ebene bleibt nur für die Dauer einer Session erhalten. Hierbei handelt es sich um die Schicht, auf der der Archivbenutzer agiert. Er hat hier die Möglichkeit, alle zum Zugriff auf die ihn interessierenden Artefakte notwendigen Schritte einzuleiten. Das erfolgt innerhalb der von der Gedächtnisorganisation vorgegebenen Referenzumgebung.

Ansätze einer Softwarearchivierung

Bisher existieren nur wenige, unkoordinierte Ansätze. So erregt insbesondere frühe Software mit nationalem Ursprung ein gewisses Interesse. So besitzen beispielsweise die British Library oder die Bibliotheque National de France Sammlungen früher Computerspiele. Oder es gab die Initiative einer Medienwissenschaftlerin wie die in Neuseeland (Swalwell 2007). Die Deutsche Nationalbibliothek bewahrt insbesondere Software, die als Beigabe zu Büchern oder Zeitschriften für ihre Sammlungen abgeliefert wurden. Diese Ansätze sind jedoch bei weitem nicht ausreichend, um dem Verschwinden der Software älterer Computerplattformen Einhalt zu gebieten. Bisher tauchen immer noch größere Mengen alter Computer und Softwarepakete auf Online- und Offline-Flohmärkten auf. Ebenso erhalten Institutionen wie das Computerspielmuseum in Berlin regelmäßig Software- und Hardware Spenden. Es nimmt jedoch die Lesbarkeit der Datenträger und die Funktionsfähigkeit der Lesegeräte immer weiter ab, so dass selbst wenn diese noch gefunden wurden, die Wahrscheinlichkeit eines Zugriffs weiter sinkt.

Populärere Artefakte lassen sich teilweise verteilt im Internet finden, obschon die Copyright-Lage oft mehr als unklar ist und in vielen Fällen das Einverständnis der ursprünglichen Rechteinhaber nicht vorliegt. Dieses lässt sich teilweise kaum trivial beschaffen, da ursprüngliche Besitzer schon längst aus dem Marktgeschehen ausgeschieden sind. Hier lassen sich private Abandonware-Initiativen beobachten, die versuchen solche Software zu sammeln und zu bewahren. Sie operieren jedoch in einer unklaren rechtlichen Situation, zudem ist die Nachhaltigkeit solcher Ansätze nicht wirklich klar.

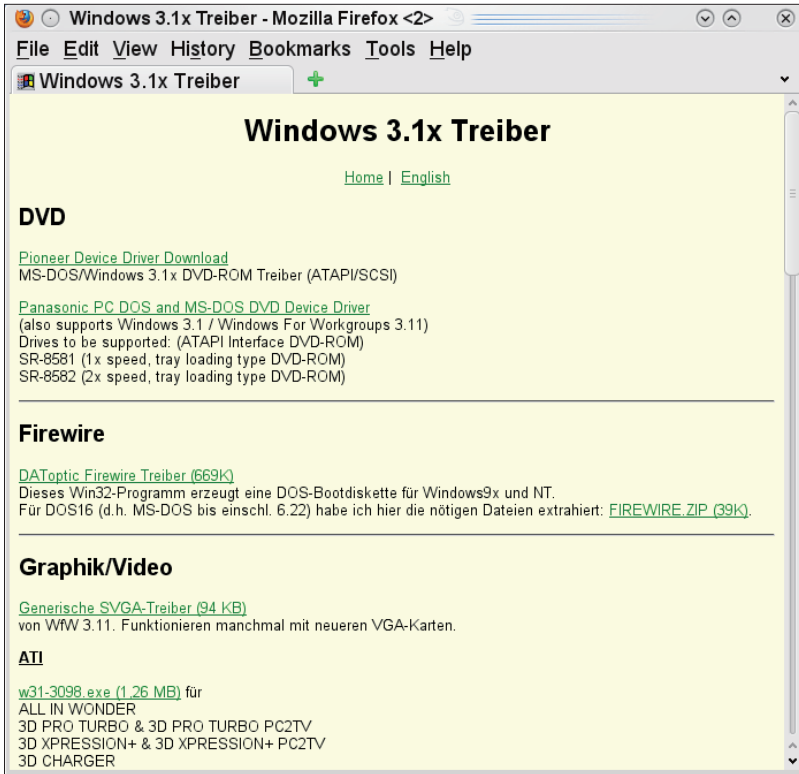


Abbildung 7: Diese Seite im Netz versammelt eine größere Auswahl an Hardwaretreibern für das inzwischen historische Windows 3.11. Einige davon können benötigt werden, um beispielsweise Audioausgaben oder Netzwerkkommunikation in einer emulierten Hardware zu ermöglichen.

Etwas anders liegt das Problem bei Hardwaretreibern, die insbesondere für die Emulation der X86-Architektur benötigt werden (Abb. 7). So wird populäre Hardware, wie diverse Soundblaster Audiokarten, Novell NE2000 oder Intel-Netzwerkkarten sowie S3 oder Cirrus-Grafikkarten in Software nachgebildet, jedoch der jeweilige Originaltreiber des ursprünglichen Herstellers benötigt.

Fazit

Für eine langfristig zuverlässige digitale Langzeitarchivierung muss die Nachhaltigkeit einer Softwarearchivierung sichergestellt werden. Diese Aufgabe sollte von einem geeigneten Konsortium, beispielsweise aus großen Gedächtnisorganisationen, übernommen werden. Insbesondere im Hinblick auf die Erhaltung des notwendigen Knowhows sollte hier ebenfalls ein verteilter Ansatz gewählt werden, in dem sich einzelne Institutionen auf einen Bereich spezialisieren, jedoch ein Austausch und gemeinsamer Zugriff möglich bleiben. Insbesondere wenn es um die Bewahrung unterschiedlich lokalisierter Varianten von Software geht, bietet sich eine Kooperation der nationalen Institutionen an.

Darüber hinaus sind für Softwarearchive geeignete rechtliche Rahmenbedingungen zu schaffen, die sicherstellen, dass auch nach dem kommerziellen Ende einer Software oder dem Ausscheiden des ursprünglichen Herstellers und seiner eventuellen Nachfolger aus dem Markt ein rechtlicher und technischer Zugriff weiterhin auf die betreffende Software möglich bleibt. Eine besondere Herausforderung der Softwarearchivierung liegt in der Bewahrung von Spezialkomponenten, wie Hardwaredrivers für die durch die Emulatoren bereitgestellten Netzwerk-, Grafik- oder Soundkarten. Hinzu kommen Codecs oder Fonts, die für bestimmte Typen von Videos, Audio oder Dokumenten benötigt werden.

Ein weiterer Gegenstand für zukünftige Forschungen und Entwicklungen besteht in der geeigneten Einbindung von Softwarearchiven in Langzeitarchivierungssysteme. Dieses sollte sicherstellen, dass in der Abhängigkeit der Eigenschaften der jeweiligen Objekte (significant properties) im Zuge der Aufnahme in das Langzeitarchiv alle notwendigen Zusatzkomponenten mitarchiviert sind oder werden. Ebenfalls sollten Prozesse, die Emulation involvieren, automatisiert werden können, um auch wenig kundigen Anwendern ohne großen Aufwand einen späteren Zugriff auf die Artefakte zu erlauben (Rechert 2010).

Dr. Dirk von Suchodoletz
Lehrstuhl für Kommunikationssysteme
Rechenzentrum Universität Freiburg
Hermann-Herder-Str. 10
D-79104 Freiburg im Breisgau
dirk.von.suchodoletz@rz.uni-freiburg.de

Referenzen

- Beger, Gabriele, 2005, „Langzeitarchivierung und Recht“ im Tagungsband zu „Kulturelles Gedächtnis im 21. Jahrhundert“, Universitätsverlag Karlsruhe
- Neuroth, Heike et.al., 2009, „Eine kleine Enzyklopädie der digitalen Langzeitarchivierung, Version 2.0 - Juni 2009“, Verlag Werner Hülsbusch
- Hoeven, Jeffrey van der, Verdegem, Remco, 2006, „Emulation: To be or not to be“, IS&T Conference on Archiving 2006, Ottawa, Canada
- Planets, 2010, EU-gefördertes Projekt zur digitalen Langzeitarchivierung, WWW-Dokument, <http://www.planets-project.eu>
- Rechert, Klaus, Suchodoletz, Dirk von; 2010, „Tackling the Problem of Complex Interaction Processes in Emulation and Migration Strategies“, ERCIM News 80
- Suchodoletz, Dirk von; 2009, „Funktionale Langzeitarchivierung digitaler Objekte - Erfolgsbedingungen des Einsatzes von Emulationsstrategien“, Cuvillier Verlag Göttingen
- Swalwell, Melanie; 2007, WWW-Dokument, <http://nztronix.org.nz> „Early New Zealand Software Database“