nestor
Kompetenznetzwerk Langzeitarchivierung

nestor Handbuch:

Eine kleine Enzyklopädie der digitalen Langzeitarchivierung

15.3.4 Interaktive Applikationen

Herausgeber

Heike Neuroth
Hans Liegmann (†)
Achim Oßwald
Regine Scheffel
Mathias Jehn
Stefan Strathmann

GEFÖRDERT VOM

Bundesministerium
für Bildung
und Forschung

Im Auftrag von

nestor – Kompetenznetzwerk Langzeitarchivierung und Langzeitverfügbarkeit digitaler Ressourcen für Deutschland
nestor – Network of Expertise in Long-Term Storage of Digital Resources
http://www.langzeitarchivierung.de

Kontakt

editors@langzeitarchivierung.de
c/o
Niedersächsische Staats- und Universitätsbibliothek Göttingen
Heike Neuroth
Forschung und Entwicklung
Papendiek 14
37073 Göttingen
Tel. +49 (0) 55 1 39 38 66

Der Inhalt steht unter folgender Creative Commons Lizenz:
http://creativecommons.org/licenses/by-nc-sa/2.0/de/

15.3.4 Interaktive Applikationen

Dirk von Suchodoletz

Die Diskussionen und Forschung zur digitalen Langzeitarchivierung von statischen digitalen Primärobjekten, wie Dokumenten, Digitalisaten oder Audio- und Videodatenströme sind bereits recht weit vorangekommen. Anders liegt der Fall für dynamische digitale Objekte: Sie kommen in der aktuellen Langzeitarchivierungsdebatte bisher fast nicht vor. Zu ihnen zählen:

- Betriebssysteme. Sie erlauben gemeinsam mit der physikalischen Hardware eines jeden Rechners überhaupt erst seinen sinnvollen Betrieb. Ihre Aufgabe besteht in der Steuerung der Hardware, der Ressourcenverwaltung und -zuteilung. Sie erlauben die Interaktion mit dem Endanwender. Sie sind im kompilierten Zustand – übersetzt aus dem Quellcode in ausführbaren Maschinencode - deshalb nur auf einer bestimmten Architektur ablauffähig. Mit der Übersetzung erfolgte die Anpassung an bestimmte Prozessoren, die Art der Speicheraufteilung und Peripheriegeräte zur Ein- und Ausgabe. Da eine Reihe von Funktionen von verschiedenen Programmen benötigt werden, sind diese oft in sogenannte Bibliotheken ausgelagert. Programme, die nicht alle Funktionen enthalten, laden benötigte Komponenten dynamisch aus den Bibliotheken zur Laufzeit nach. Bibliotheken und Programme hängen dementsprechend eng miteinander zusammen.
- Anwendungsprogramme. Sie setzen auf der Betriebssystemebene auf. Für die schematische Darstellung der Arbeit eines Computers wird oft ein Schichtenmodell gewählt, das die Anwendungen in der obersten Ebene anzeigt (Abbildung 15.3.4.1). Applikationen sind Programme, die für bestimmte, spezialisierte Aufgaben erstellt wurden. Mit diesen Programmen generierten und bearbeiteten Endanwender Daten der verschiedensten Formate. Der Programmcode wird im Kontext des Betriebssystems ausgeführt. Er kümmert sich um die Darstellung gegenüber dem Benutzer und legt fest, wie beispielsweise die Speicherung von Objekten in einer Datei organisiert ist und der Anwender mit dem Computer interagiert. Das Betriebsystem übernimmt die Speicherung von Dateien auf Datenträgern üblicherweise angeordnet in Verzeichnissen. Zur Ausführung auf einer bestimmten Rechnerarchitektur werden Betriebssysteme und Applikationen aus dem sogenannten Quellcode in den passenden Binärcode übersetzt. Deshalb können Programme und Bibliotheken nicht beliebig zwischen verschiedenen Betriebssystemen verschoben werden.

- Computerspiele (eigener Abschnitt)
- Interaktive Medien zur Lehre und Unterhaltung (eigener Abschnitt)
- Datenbanken - zählen ebenfalls zu den sehr frühen Anwendungen von Rechnern. Die Bewegung, Durchsuchung und Verknüpfung großer Datenbestände gehört zu den großen Stärken von Computern. Diese elektronischen Datenbestände stellen oft die Grundlage für abgeleitete Objekte dar. Zur Klasse der datenbankbasierten Anwendungen zählen Planungs- und Buchhaltungssysteme, wie SAP, elektronische Fahrpläne diverser Verkehrsträger bis hin zu Content Management Systemen moderner Internet-Auftritte von Firmen und Organisationen. Wenn von einer Datenbank sehr verschiedene Ansichten ad-hoc erzeugt werden können, ist sehr schwer abzusehen, welche dieser Ansichten zu einem späteren Zeitpunkt noch einmal benötigt werden könnten. Unter Umständen hat man sich dann auf Teilmengen festgelegt, die von nachfolgenden Betrachtern als unzureichend oder irrelevant eingestuft werden. Gerade bei Datensammlungen wichtiger langlebiger Erzeugnisse wie Flugzeugen, Infrastrukturen oder Gebäuden besteht großes allgemeines Interesse eines zeitlich unbeschränkten Zugriffs.

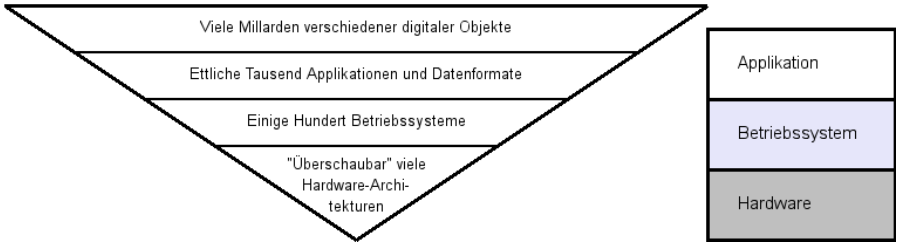

Abbildung 15.3.4.1: Sicht auf die Anzahl der nachzubildenden Objekte (links) je nach Wahl der Schicht für typische Rechnerplattformen (rechts)

Dynamische Daten zeichnen sich dadurch aus, dass sie außerhalb eines festgelegten digitalen Kontexts nicht sinnvoll interpretiert und genutzt werden können. Eine Überführung in ein analoges Medium scheidet aus: Der Ausdruck des Programmcodes auf Papier oder die Aufnahme einer Programmsitzung auf Video sind derart „verlustbehaftete" Speicherverfahren, dass sie im Sinne der Archivierung in keiner Weise die Anforderungen erfüllen.

Emulation – Erhalt von Ablaufumgebungen

Emulation setzt nicht am digitalen Objekt selbst an, sondern beschäftigt sich mit der Umgebung, die zur Erstellung dieses Objektes vorlag. Das bedeutet beispielsweise die Nachbildung von Software durch andere Software, so dass es für ein digitales Objekt im besten Fall keinen Unterschied macht, ob es durch die emulierte oder durch die Originalumgebung behandelt wird. Ebenso kann Computerhardware durch Software nachgebildet werden, auch wenn dieses erst einmal deutlich komplexer erscheint. Generell lässt sich feststellen, dass für aktuelle Computerplattformen üblicherweise drei Ebenen für den Ansatz von Emulation identifiziert werden können:

Emulation oder Ersatz von Applikationen durch andere, um so bestimmte Datenformate darzustellen oder auszuführen. In der heutigen Softwarelandschaft ist es üblich bestimmte Applikationen oder deren Funktionalität nachzuprogrammieren, um sie in eigene Produkte zu integrieren oder den Funktionsumfang bestehender Software zu erweitern. Für viele Anwender sicherlich am naheliegendsten ist die Emulation der Eigenschaften älterer Applikationen in aktuellen Anwendungsprogrammen. Die Interpretation von einer Applikation nicht-eigenen aktuellen Datenformaten fällt ebenso in diesen Bereich. Auf diese Weise wird überhaupt ein Datenaustausch zwischen verschiedenen Programmen unterschiedlicher Hersteller erst möglich. Deshalb gehören solche Funktionen oftmals zum Standardumfang einer Applikation. Ein typisches Beispiel dieser Art von „Emulation" einer Applikation ist die Benutzung von StarOffice oder OpenOffice, um sich Microsoft Word Dokumente oder Excel Tabellen in einer Linux- oder Unix-Umgebung anzusehen. Dieses trifft auf eine ganze Reihe weiterer Applikationen zu.

Emulation von Betriebssystemen, um so auf einem gegebenen Betriebssystem Applikationen auszuführen, die die Schnittstellen (API) eines anderen Betriebssystems erwarten. Bezogen auf das Schichtenmodell (Abbildung 1) setzt diese Art der Emulation auf der Betriebssystemebene an. Die Emulation von Betriebssystemen zielt darauf ab, dass sich beispielsweise Programme unter Linux/X11 ausführen lassen, die ursprünglich für Microsoft Windows geschrieben wurden. Die Motivation liegt nicht primär in der Langzeitarchivierung begründet, sondern darin dem Zwang auszuweichen, für eine bestimmte gewünschte Applikation auch ein anderes Betriebssystem verwenden zu müssen. Besonders hervorheben kann man an dieser Stelle das Wine-Projekt oder Cygwin (Posix unter Windows).

Nachbildung eines kompletten Rechners einer bestimmten Architektur. Der Ansatz die komplette Hardware einer bestimmten Rechnerplattform zu archivieren, nutzt die tiefste Schicht (Abbildung 15.3.4.1). Deshalb reicht der Ansatz die komplette Hardware einer Computerarchitektur in Software nachzubilden deutlich weiter. Dieses scheint auf den ersten Blick sehr aufwändig, hat jedoch entscheidende Vorteile: Die Schnittstellen sind offengelegt, da oft recht verschiedene Anbieter ihre Betriebssysteme für die verschiedenen Architekturen entwickelt haben.

Emulation bedeutet also zunächst nur die Erschaffung einer virtuellen Umgebung in einer gegebenen Ablaufumgebung, üblicherweise in dem zum Zeitpunkt des Aufrufs üblichen Computersystem. Emulatoren bilden somit die Schnittstelle, eine Art Brückenfunktion, zwischen dem jeweils aktuellen Stand der Technik und einer längst nicht mehr verfügbaren Technologie. Dabei müssen sich Emulatoren um die geeignete Umsetzung der Ein- und Ausgabesteuerung und der Peripherienachbildung bemühen.

Die Auswahl der inzwischen kommerziell erhältlichen oder als Open-Source-Software verfügbaren Emulatoren oder Virtualisierer ist inzwischen recht umfangreich geworden, so dass häufig sogar mehr als ein Emulator für eine bestimmte Rechnerarchitektur zur Verfügung steht. Jedoch eignet sich nicht jeder Emulator gleichermaßen für die Zwecke des Langzeitzugriffs.

So existiert für frühe Architekturen keine deutliche Unterscheidung zwischen Betriebssystem und Applikation. Home-Computer verfügten über eine jeweils recht fest definierte Hardware, die zusammen mit einer Art Firmware verbunden ausgeliefert wurde. Diese Firmware enthält typischerweise eine einfache Kommandozeile und einen Basic-Interpreter. Nicht alle für den Betrieb von Emulatoren benötigten Komponenten, wie beispielsweise genannte Home-Computer-Firmware, ist frei verfügbar. Sie müssen ähnlich wie ein Betriebssystem für spätere Architekturen erworben worden sein. Lediglich für einige Systeme existieren frei verfügbare Nachprogrammierungen. Für die X86-Architektur beispielsweise liegt der Fall mit der Verfügbarkeit eines Open Source Systems und Grafikkarten-BIOS einfacher.

Der überwiegende Anteil von Emulatoren und Virtualisierern wurde oftmals aus ganz anderen als Langzeitarchivierungsgründen erstellt. Sie sind heutzutage Standardwerkzeuge in der Software-Entwicklung. Nichtsdestotrotz eignen sich

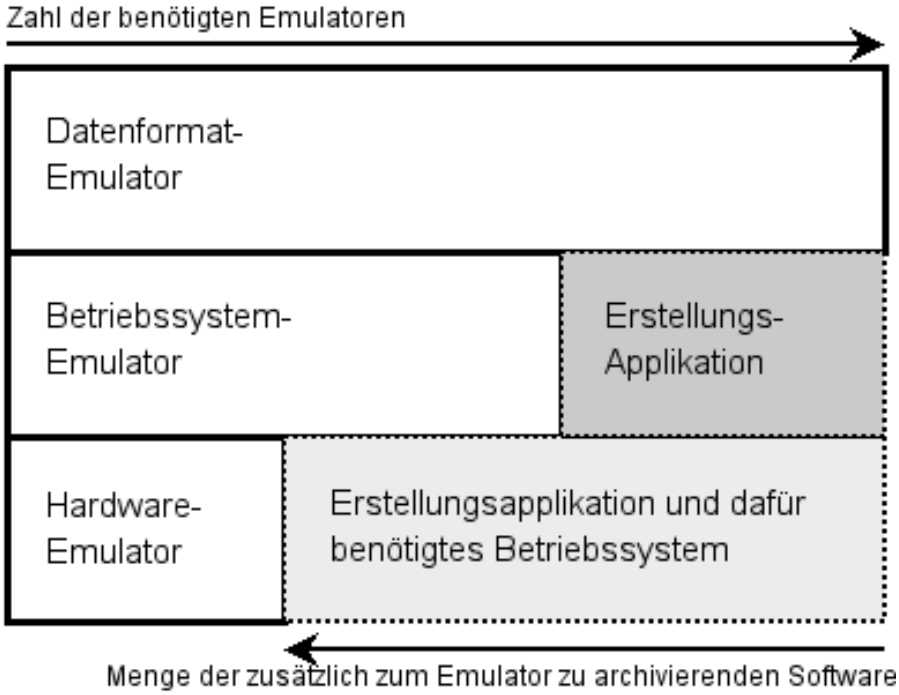

Abbildung 15.3.4.2: Je nach gewählter Ebene der Emulation für die spätere erfolgreiche Darstellung des interessierenden Primärobjekts notwendige Menge von Sekundärobjekten

viele dieser Werkzeuge für eine Teilmenge möglicher Langzeitarchivierungsaufgaben. Institutionen und private Nutzern reichen für zeitlich befristete Archivierung derzeitig verfügbare Programme oftmals aus.

Softwarearchiv als eine Erfolgsbedingung für Emulationsstrategien

Je nach gewählter Ebene der Emulation werden zusätzliche Softwarekomponenten benötigt. Bei der Emulation von Applikationen für den alternativen Zugriff auf ein bestimmtes Datenformat entfällt der Bedarf an weiterer Software. Optimalerweise laufen die emulierten Applikationen auf einer aktuellen Plattform und erlauben aus dieser heraus den direkten Zugriff auf die digitalen Objekte des entsprechenden Formates. 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 wohldokumentierten Bildformate.

Die Emulation eines Betriebssystems oder dessen Schnittstellen erlaubt theoretisch alle Applikationen für dieses Betriebssystem ablaufen zu lassen. Dann müssen neben dem Emulator für das entsprechende Betriebssystem auch sämtliche auszuführende Applikationen in einem Softwarearchiv aufbewahrt werden (Abbildung 15.3.4.2). Bei der Portierung des Betriebssystememulators muss darauf geachtet werden, dass sämtliche in einem Softwarearchiv eingestellten Applikationen weiterhin ablaufen können.

Die Nachbildung einer kompletten Hardware in Software verspricht die besten Ergebnisse und verfolgt den generellsten Ansatz. Nun benötigt man jedoch in jedem Fall mindestens eines oder je nach Bedarf mehrere Betriebssysteme, die sich als Grundlage der darauf aufesetzenden Applikationen ausführen lassen. Das bedeutet für ein Softwarearchiv, dass neben dem Emulator für eine Plattform auch die darauf ablauffähigen Betriebssysteme aufgehoben werden müssen. Das gilt ebenfalls für die darauf basierenden Applikationen, die zur Darstellung der verschiedenen Datenformate erforderlich sind. Zusätzlich zu den benötigten Applikationen sind oft eine Reihe von Hilfsprogrammen oder -komponenten erforderlich. Teilweise sind Dateien komprimiert oder in einem gepackten Archiv zusammengeführt und müssen vor dem Zugriff erst entpackt werden. Einige Dokumente erfordern zusätzliche Schriftarten zu ihrer Darstellung, wie einige digitale Videos nur mit einem bestimmten Codec abgespielt werden können. Erfolgt eine Portierung, also Migration des Hardwareemulators, muss anschließend überprüft werden, dass die gewünschten Betriebssysteme weiterhin ablauffähig bleiben. Da die meisten Applikationen lediglich die Schnittstellen des Betriebssystems nutzen (sollten), folgt ihre Funktionsfähigkeit direkt aus der der Betriebssysteme. Betriebssysteme benötigen zur Ansteuerung der Hardware, wie Netzwerk- und Soundkarte oder Grafikadapter passende Treiber, die mit den jeweiligen Komponenten korrespondieren. Deshalb kann das äußere Update des Emulators oder Virtualisierers, neben der Migration des Containerformats der virtuellen Festplatte, auch eine Migration des Treiber-Sets des darin installierten Betriebssystems nach sich ziehen.

Dieses Dilemma löst beispielsweise VMware derzeit noch durch die Pflege und Bereitstellung geeigneter Treiber für jedes offiziell unterstützte Betriebssystem. Noch ist die Liste der Treiber sehr lang und im Sinne der Archivierung fast vollständig. Ein weiteres Problem ist die Art der Prozessor-Nutzung. Viele virtuelle Maschinen reichen CPU-Befehle des Gastes direkt an die Host-CPU

weiter. Das setzt voraus, dass diese mit diesen Befehlen umgehen kann.

Geeigneter im Sinne einer wirklichen Langzeitarchivierung sind quelloffene Implementierungen, wie QEMU, Bochs oder das Java-basierte Dioscuri. 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, wie ISA-Netzwerk- und Soundkarten, dauerhaft virtualisiert wird und nicht einem Produktzyklus zum Opfer fällt. Für alte Betriebssysteme, wie DOS oder Windows 3.1 besteht andernfalls die Gefahr, dass sie nur noch beschränkt ausgeführt werden können.

Ein weiterer Punkt ist die unter Umständen notwendige Anpassung der Containerdateien, in denen die Gastsysteme auf der virtuellen Maschine 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.

Ein Softwarearchiv sollte daher so angelegt sein, dass auch künftige Änderungen der Emulationsstrategie, bedingt durch veränderte Szenarien des Zugriffs auf die dynamischen digitalen Objekte, realisiert werden können.

Softwarearchiv – Komponenten

Nach den eher theoretisch angelegten Vorbetrachtungen steht nun das Softwarearchiv als Hilfsmittel der Emulationsstrategie im Mittelpunkt. Der Erhalt möglichst originalgetreuer Ablaufumgebungen für die verschiedensten Typen digitaler Objekte erfordert nicht nur die Bereitstellung der Primärwerkzeuge. Neben den eigentlichen Emulatoren sind eine ganze Reihe weiterer Software-Komponenten erforderlich.

Die Hardwareemulation setzt auf der untersten Ebene des Schichtenmodells 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 zu sehen, müssen je nach Architektur die Ebenen zwischen der emulierten Hardware und dem Objekt selbst "überbrückt" werden. So kann ein Betrachter nicht auf einer nackten X86-Maschine ein PDF-Dokument öffnen. Er braucht hierfür mindestens ein Programm zur Betrach-

tung, welches seinerseits nicht direkt auf der Hardware ausgeführt wird und deren Schnittstellen direkt programmiert. Dieses Programm setzt seinerseits ein Betriebssystem als Intermediär voraus, welches sich um die Ansteuerung der Ein- und Ausgabeschnittstellen der Hardware kümmert.

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

Da sowohl die für den Anwender interessanten Daten und Anwendungsprogramme als auch die zur Ansicht notwendigen Hilfsprogramme, Emulatoren, Betriebssysteme digitale Objekte sind, sollen sie wie folgt unterschieden werden:

- Ein Zielobjekt oder Primärobjekt gehört zu den primär interessierenden Objekten eines Archivbenutzers, welches er in irgendeiner Form geeignet betrachten will.
- Für die Betrachtung wird Software (Erstellung einer Nutzungsumgebung für das Primärobjekt) benötigt. Die notwendigen Komponenten ergeben sich aus dem jeweiligen View-Path (siehe unten) und können je nach Wahl dessen variieren. Diese Software in den verschiedenen Ausformungen wird als Hilfs- oder Sekundärobjekt bezeichnet. Es ist für den Benutzer nicht primär von Interesse, wird aber gebraucht, um überhaupt mit dem Zielobjekt umgehen zu können.
- Je nach Art des Zielobjektes können eine ganze Reihe verschiedener Sekundärobjekte notwendig sein.

Das Softwarearchiv selbst kann wieder als Bestandteil eines größeren Archivs nach dem OAIS-Modell angesehen werden. Für die Aufbewahrung der Emulatoren, der Betriebssysteme, Applikationen und Hilfsprogramme gelten die identischen Regeln, wie für die eigentlichen digitalen Primärobjekte. Trotzdem kann es von Interesse sein, diese Daten in direkt zugreifbarer Weise oder auch in speziell aufbereiteter Form vorzuhalten.

Ein weiterer 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. Da es sich hierbei um Software handelt, ist diese auf eine bestimmte Nutzungsum-

gebung angewiesen. 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. Andererseits wird sich je nach Erstellungsdatum des Objektes 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 interessierenden 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 muss deshalb eine ganze Reihe verschiedener Softwarekomponenten umfassen, so sind:

- die Emulatoren zu speichern, so dass mit ihrer Hilfe die Wiederherstellung einer Rechner-Architektur für bestimmte Nutzungsumgebungen erfolgen kann.
- die Betriebssysteme abzulegen, die je nach Rechnerplattform einen Teil der Nutzungsumgebung ausmachen.
- Treiber der Betriebssysteme zusätzlich zu speichern, da sie den Betriebssystemen überhaupt erlauben mit einer bestimmten Hardware umzugehen.
- die Applikationen zu archivieren, mit denen die verschiedenen digitalen Objekte erstellt wurden. Diese Applikationen sind ebenfalls Bestandteil der Nutzungsumgebung des Objektes. Sie sind in vielen Fällen auf die vorgenannten Betriebssysteme angewiesen.
- die unter Umständen notwendigen Erweiterungen einer Applikationsumgebung, wie bestimmte Funktionsbibliotheken, Codecs oder Schriftartenpakete zur Darstellung.
- 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.
- je nach Primärobjekt oder gewünschter Nutzungsumgebung sind mehrere Varianten derselben Software zu archivieren, um beispielsweise die Lokalisierung in eine bestimmte Ein- und Ausgabesprache zu erreichen.

View-Paths und Referenzumgebungen

Digitale Objekte können nicht aus sich alleine heraus genutzt werden, sie

bedürfen eines geeigneten Kontextes, damit auf sie zugegriffen werden kann. Dieser Kontext, Nutzungsumgebung genannt, muss geeignete Hardware- und Softwarekomponenten so kombinieren, dass je nach Objekttyp die Erstellungsumgebung oder ein geeignetes Äquivalent abgebildet werden.

Die Wiederherstellung von Nutzungsumgebungen oder geeigneter Äquivalente läßt sich am besten durch sogenannte „View-Paths", Wege ausgehend vom Primärobjekt des Interesses zur Arbeitsumgebung des Betrachters oder Anwenders, veranschaulichen und formalisieren. Im Zuge des DIAS Projekts an der Königlichen Bibliothek der Niederlande[26] wurde dieses Konzept vorgestellt. Die Abbildung 15.3.4.3 zeigt einen typischen View-Path ausgehend vom Primärobjekt, über seine Erstellungsapplikation, das durch diese erforderte Betriebssystem und daraus resultierendem Hardwareemulator.

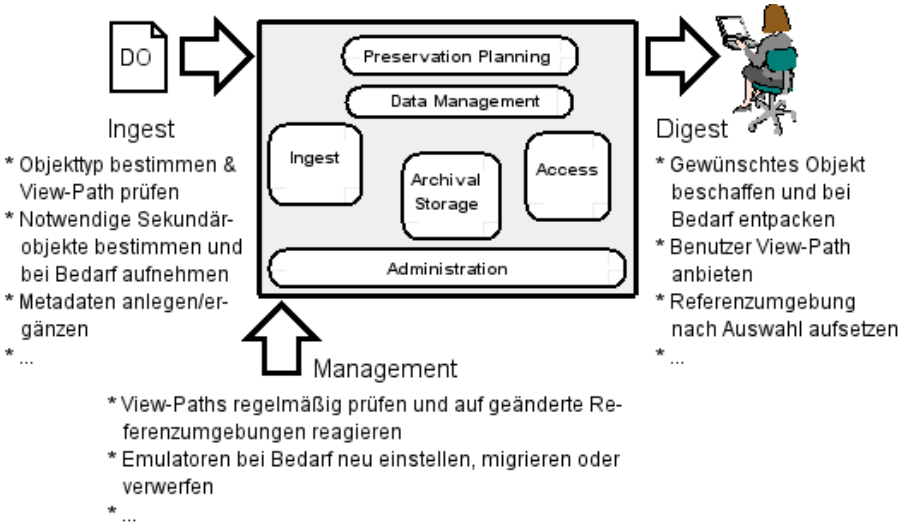

Abbildung 15.3.4.3: Verschiedene Aufgabenstellungen für das Management eines Langzeitarchivs

Für ein digitales Langzeitarchiv erwachsen daraus verschiedene Aufgaben, die von der Bestimmung des Objekttyps beim Einstellen in das Archiv, der Generierung und Ablage der benötigten Metadaten bis zur Überprüfung auf Vorhandensein oder der Herstellbarkeit einer Ablaufumgebung reichen. Abbildung 15.3.4.4 zeigt ein digitales Objekt, das für seine Betrachtung eine Reihe von

26 siehe dazu das Konzept des Preservation-Managers in http://www.kb.nl/hrd/dd/dd_onderzoek/preservation_subsystem-en.html

Sekundärobjekten braucht - angefangen von der Applikation, mit der es erstellt wurde, über das Betriebssystem, das diese ausführen kann. Letzteres wird sich je nach Alter nicht mehr auf moderner Hardware ausführen lassen und läuft deshalb in einem geeigneten Hardwareemulator ab.

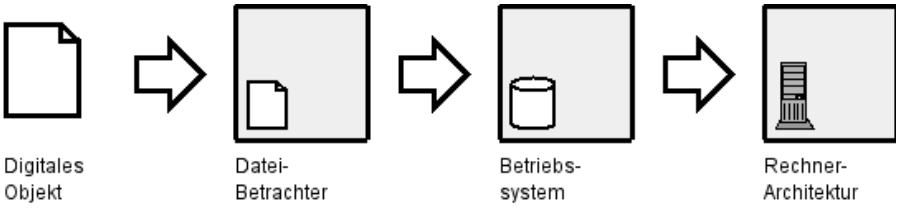

Abbildung 15.3.4.4: Ein typischer View-Path ausgehend von einem digitalen Primärobjekt

Für das OAIS-Management leiten sich daraus verschiedene Aufgaben ab. Eine Rolle spielt beispielsweise die Klassifikation der Objekte nach ihrem Objekttyp (neben ihren Metadaten), da dieser über die später notwendigen Schritte für den Zugriff mitbestimmt. Ebenso können je nach Benutzergruppe oder Einrichtung verschiedene Kriterien existieren, nach denen View-Paths bestimmt werden. Es lassen sich drei wesentliche Phasen voneinander unterscheiden:

- Notwendige Handlungen bei der Objektaufnahme ins Archiv (Ingest)
- Regelmäßige Schritte und Arbeitsabläufe während des Archivbetriebs
- Handlungsanweisungen für den Objektzugriff durch den beziehungsweise die Objektausgabe an den Endbenutzer (Digest)

Zur Bestimmung des Objekttyps existieren verschiedene Möglichkeiten; ein über das Internet nutzbarer Dienst mit einer breiten Unterstützung ist beispielsweise PRONOM. Die Auswahl über die in das Softwarearchiv einzustellenden und später zu pflegenden Emulatoren entscheidet wesentlich über die Nachnutzbarkeit der Primärobjekte des Langzeitarchivs. Deshalb ist bei der Einstellung eines digitalen Objekts in das Archiv zu bestimmen, ob für diesen Objekttyp ein View-Path bestimmt werden kann und auf welche später nachzubildende Rechner-Plattform dieser zeigt. Umgekehrt kann das beim Ingest bedeuten, dass ein Objekt zurückgewiesen werden kann oder es nur in Erwartung auf eine später verfügbare Lösung eingestellt wird.

Aus Sicht des Archivmanagements unterscheiden sich Emulatoren nicht wesentlich von den Primärobjekten. Sie werden je nach Archivstrategie und Referenzplattform im Laufe der Zeit obsolet, müssen geeignet migriert oder ihre Nutzungsumgebung erhalten werden. Zu einem gegebenen Zeitpunkt wird ein Primärobjekt von einem Archivbenutzer nachgefragt werden. Dieser erwartet

typischerweise eine geeignete Umgebung, in der er das Objekt betrachten oder benutzen 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 zu treffen, wie dieser geeignet an die notwendige Software und ihre Schnittstellen herangeführt werden kann, um mit ihr umgehen zu können. Ebenso kann es notwendig werden eine Reihe von Hilfsmitteln zu präsentieren.

Während der Ausgangspunkt des View-Paths durch das Primärobjekt festgelegt ist, wird sich, erzwungen 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.
- Es kann Primärobjekte geben, zu denen zu bestimmten Zeitpunkten keine View-Paths vorhanden sind.

Zur sinnvollen Bestimmung der Existenz von View-Paths sollten sie sich deshalb auf bestimmte Referenzumgebungen mit jeweils festgelegten Eigenschaften aus Hard- und Software beziehen. Einen View-Path kann man sich damit als Entscheidungsbaum vorstellen, an dessen Wurzel das Primärobjekt steht. Zur Veranschaulichung des Aufbaus der benötigten Ablaufumgebung läßt sich wiederum ein Schichtenmodell vorstellen.

Viele Primärobjekte lassen sich durch mehr als eine Applikation (Viewer) darstellen. Dabei können die Anzeigeergebnisse in Authentizität, Komplexität oder Qualität differieren. Damit ergibt sich eine Pfadverzweigung und auf der Schicht der Applikation eine Auswahl. Ähnliches trifft auf die Anforderung der Applikation nach einem Betriebssystem zu, so dass in dieser Schicht eine weitere Verzweigung auftreten kann. Die Rekursion setzt sich mit dem Betriebssystem und einer möglichen Auswahl an geeigneten Hardware-Emulatoren fort.

Die Modellierung des View-Path in Schichten erfolgt nicht starr: So reduziert sich beispielsweise bei einem digitalen Primärobjekt in Form eines Programms die Zahl der Schichten. Ähnliches gilt für einfache Plattformen, wie Home-Computer, wo keine Trennung zwischen Betriebssystem und Applikation vorliegt. Darüber hinaus können Schichten wiederum gestapelt sein, wenn es

erforderlich wird für einen bestimmten Emulator seinerseits eine geeignete Ablaufumgebung herzustellen (vgl. Abbildung 15.3.4.1).

Eine sinnvolle Erweiterung des etwas unbestimmten Ansatzes im starren DIAS-Modell (eine der ersten Realisierungen des View-Path-Konzepts) könnte in der Gewichtung der einzelnen View-Path-Optionen liegen, die 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 (anhand bestimmter Metriken) vorzunehmen, ob beispielsweise besonderer Wert auf die Authentizität der Darstellung oder eine besonders einfach 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-Path verbunden ist, um bei Bedarf eine zusätzliche ökonomische Bewertung zu erlauben.

Eine Schlussfolgerung könnten mehrdimensionale Metriken sein, die mit den Objektmetadaten gespeichert und durch das Archivmanagement regelmäßig aktualisiert werden. Da die Entscheidung über die Qualität einer Darstellung oder Ausführung eines Primärobjekts oft nur vom Anwender getroffen werden kann, sollte man Überlegungen anstellen, wie Benutzerrückmeldungen zur Erstellung der Metriken einfließen könnten.

Geeignete *Referenzumgebungen*, als Bezugspunkte für die Darstellung der Primärobjekte, versuchen in möglichst kompakter und gut bedienbarer Form ein ganzes Spektrum von Ablaufumgebungen zur Verfügung stellen zu können. Dabei sollte die Basisplattform möglichst der jeweils aktuellen Hardware mit jeweils üblichen Betriebssystemen entsprechen. Das verhindert einerseits das Entstehen eines Hardwaremuseums mit hohen Betriebskosten. Andererseits findet sich der Benutzer zumindest für das Basissystem in der gewohnten Umgebung wieder.

Jede historische Rechnerplattform, bis zu den heutigen, weist ihre eigenen Komplexitäten auf, die nicht von jedem durchschnittlichen Computeranwender sinnvoll bewältigt werden können. Zudem kann es nicht die Voraussetzung für den Zugriff auf ein bestimmtes Primärobjekt sein, dass die interessierte Person sich mit der Erstellungsumgebung und ihrer Einrichtung auskennt. Je nach Typus ihrer Benutzer werden deshalb Betreiber digitaler Langzeitarchive nach Lösungen suchen, die es erlauben in geeigneter Weise auf ihre jeweiligen Pri-

märobjekte zuzugreifen. In jedem Fall ist eine gewisse Abstraktionsschicht zu schaffen, über die ein Objektzugriff erfolgen kann. Hierfür sind verschiedene Varianten denkbar, die an unterschiedlichen Schnittstellen ansetzen:

- Referenzplattformen aus einer bestimmten Hardware und Software werden als Endpunkte von View-Paths benötigt. Sie können ortsnah zum Langzeitarchiv, beispielsweise parallel oder als Erweiterung der üblichen Recherchesysteme der jeweiligen Gedächtnisorganisation aufgestellt oder in diese integriert werden.
- Sie ließen sich in einem gewissen Umfang virtualisieren, um sie per Web-Browser oder anderen geeigneten und allgemein genutzten Internet-Technologien entfernt zur Verfügung zu stellen.
- Alternativen bestehen darin definierte virtuelle Maschinen, wie Java, zur Ausführung von Emulatoren und ihren enthaltenen Nutzungsumgebungen zu verwenden (Dioscuri). Hierdurch erreicht man einerseits eine breitere Auswahl von Referenzumgebungen bei einer stabileren Schnittstelle zur virtuellen Maschine. Andererseits verlagert man damit das Problem eine Schicht weiter nach unten und hängt nun von der Weiterentwicklung virtuellen Maschine ab.

Eine Referenzumgebung sollte in der Lage sein, neben der jeweiligen Nutzungsumgebung zusätzlich die notwendigen Hinweise zum Aufsetzen und zur Bedienung bereitzustellen, welche einen geeigneten Zugriff auf die Objektmetadaten beinhalten. Weitere Kriterien liegen in der Güte der Darstellung der Nutzungsumgebung. Wegen ihrer durch die eingesetzten Emulatoren und Viewer spezielleren Anforderungen ist es für die Betreiber von Langzeitarchiven vielfach sinnvoll eine Referenzplattform selbst zu definieren und bereitzustellen. Diese wird sich je nach Anwendung und Gedächtnisorganisation unterscheiden: Bibliotheken und Archive benötigen in erster Linie Viewer für migrierte statische Objekte und Emulatoren für die nicht-migrierten Archivinhalte. Soweit sinnvoll kann diese Aufgabe auf bereits vorhandenen Recherche- oder Anzeigesystemen der Institution untergebracht werden, um den Benutzern einen leichten Zugriff zu erlauben. Ein Datenaustausch mit der Außenwelt kann benötigt werden, wenn Datenarchäologie angeboten werden oder ein Ausdruck eines Dokuments erfolgen soll.

Technische Museen oder Ausstellungen leben eher von interaktiven Objekten. Die Referenzworkstation ist je nach zu zeigendem Exponat zu bestücken; ein Datenaustausch ist üblicherweise nicht vorgesehen. Für Firmen oder Institutionen kann bereits ein X86-Virtualisierer ausreichen, um der Zugreifbarkeit auf den Archivbestand Genüge zu tun. Die erwarteten Objekte sind eher statischer Natur und wurden typischerweise auf PC's verschiedener Generationen

erstellt. Generell muss es sich bei den eingesetzten Referenzworkstations nicht um die jeweils allerneueste Hardwaregeneration handeln. Stattdessen sollte die Technologie angestrebt werden, die einen optimalen Austausch erlaubt und den Anforderungen der jeweiligen Nutzer gerecht wird.

Es muss ein ausreichendes Bedienungswissen vorgehalten werden, welches bei speziellen Nutzergruppen wie digitalen Archivaren, auch für recht alte Nutzungsumgebungen erwartet werden kann. Auf diese Weise lassen sich zudem Versionen der Hardware und Betriebssysteme bei allfälligen Generationswechseln der Referenzworkstations überspringen.