



nestor

Funktionale Langzeitarchivierung  
digitaler Objekte  
Erfolgsbedingungen des Einsatzes  
von Emulationsstrategien

nestor edition

Dirk von Suchodoletz





# Funktionale Langzeitarchivierung digitaler Objekte –

## Erfolgsbedingungen des Einsatzes von Emulationsstrategien

Dissertation zur Erlangung des Doktorgrades der Fakultät der  
Angewandten Wissenschaften der Albert-Ludwigs-Universität  
Freiburg im Breisgau.

Eingereicht am 22. Februar 2008, Disputation am 21. Juli 2008.

Referent: Dirk von Suchodoletz

Lehrstuhl für Kommunikationssysteme  
Rechenzentrum & Institut für Informatik an der Technischen Fakultät  
Herrmann-Herder-Str. 10  
79104 FREIBURG i.Br.  
dsuchod@uni-freiburg.de

Erstgutachter: Prof. Dr. G. Schneider

Zweitgutachter: Prof. Dr. E. Mittler

Dekan: Prof. Dr. B. Nebel







# Funktionale Langzeitarchivierung digitaler Objekte

Erfolgsbedingungen des Einsatzes  
von Emulationsstrategien

Dirk von Suchodoletz

nestor edition

Herausgegeben 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>

Projektpartner:

Bayerische Staatsbibliothek, München

Bundesarchiv

Deutsche Nationalbibliothek (Projektleitung)

FernUniversität in Hagen

Humboldt-Universität zu Berlin - Computer- und Medienservice / Universitätsbibliothek

Institut für Museumsforschung, Berlin

Niedersächsische Staats- und Universitätsbibliothek, Göttingen

© 2008

**nestor** - Kompetenznetzwerk Langzeitarchivierung und Langzeitverfügbarkeit  
Digitaler Ressourcen für Deutschland

Der Inhalt dieser Veröffentlichung darf vervielfältigt und verbreitet werden, sofern der Name des Rechteinhabers "nestor - Kompetenznetzwerk Langzeitarchivierung" genannt wird. Eine kommerzielle Nutzung ist nur mit Zustimmung des Rechteinhabers zulässig.

URN: urn:nbn:de:0008-2008070219

[<http://nbn-resolving.de/urn/resolver.pl?urn=urn:nbn:de:0008-2008070219>]

Die Schriftenreihe „nestor edition“ präsentiert ausgewählte wissenschaftliche Arbeiten mit dem Schwerpunkt Langzeitarchivierung. Die Reihe wird in loser Folge von nestor – Kompetenznetzwerk Langzeitarchivierung herausgegeben. Damit entsteht ein Forum, in dem Beiträge zu verschiedenen Aspekten der digitalen Langzeitarchivierung einer breiten Öffentlichkeit zugänglich gemacht werden. Die Arbeiten werden von ausgewiesenen Experten aus den jeweiligen Fachgebieten für die „nestor edition“ gezielt ausgewählt, wenn sie einen besonderen Beitrag zu wichtigen Themenfeldern oder zu neuen wissenschaftlichen Forschungen auf dem Gebiet leisten.

Der vorliegende erste Band „Funktionale Langzeitarchivierung digitaler Objekte. Erfolgsbedingungen des Einsatzes von Emulationsstrategien“ von Dirk von Suchodoletz eröffnet die Reihe.

Bemerkungen zu dieser Publikation, aber auch Vorschläge für die Aufnahme weiterer Beiträge in der Edition gerne an: [info@langzeitarchivierung.de](mailto:info@langzeitarchivierung.de)

Für die Partner des Projekts nestor – Kompetenznetzwerk Langzeitarchivierung  
Reinhard Altenhöner und Natascha Schumann  
Deutsche Nationalbibliothek





# Inhaltsverzeichnis

|                                                                |           |
|----------------------------------------------------------------|-----------|
| Inhaltsverzeichnis                                             | I         |
| <b>I Überblick und Motivation der Arbeit</b>                   | <b>1</b>  |
| <b>1 Einleitung</b>                                            | <b>3</b>  |
| 1.1 Motivation . . . . .                                       | 4         |
| 1.2 Einordnung und Fokus der Fragestellung . . . . .           | 5         |
| 1.3 Kapitelaufteilung und Struktur der Arbeit . . . . .        | 9         |
| <b>2 Herausforderung Langzeitarchivierung</b>                  | <b>11</b> |
| 2.1 Information, Überlieferung und Archivierung . . . . .      | 11        |
| 2.1.1 Digitale Revolution . . . . .                            | 13        |
| 2.2 Computer als "Kultur"-Maschine . . . . .                   | 14        |
| 2.2.1 Computer in Wissenschaft und Forschung . . . . .         | 15        |
| 2.2.2 Kultur und Alltag . . . . .                              | 16        |
| 2.2.3 Wirtschaft und Verwaltung . . . . .                      | 17        |
| 2.3 Innere und äußere Organisation digitaler Objekte . . . . . | 18        |
| 2.3.1 Digitale Bitströme und Dateien . . . . .                 | 19        |
| 2.3.2 Dateiformate und Standards . . . . .                     | 21        |
| 2.3.3 Datenablage . . . . .                                    | 23        |
| 2.4 Haltbarkeit digitaler Objekte . . . . .                    | 24        |
| 2.4.1 Illusion einer ewigen digitalen Kopie . . . . .          | 24        |
| 2.4.2 Randbedingungen der Lesbarkeit . . . . .                 | 25        |
| 2.4.3 Kontexte . . . . .                                       | 26        |
| 2.4.4 Abreißen der kulturellen Überlieferung . . . . .         | 27        |
| 2.5 Begrenzter Ansatz - Hardwaremuseum . . . . .               | 28        |
| 2.6 Forschungsansätze, Aktivitäten und Projekte . . . . .      | 30        |
| 2.6.1 Nationale Akteure . . . . .                              | 30        |
| 2.6.2 Initiativen und Kooperationen in Europa . . . . .        | 32        |
| 2.6.3 Internationale Aktivitäten . . . . .                     | 35        |

|           |                                                                    |           |
|-----------|--------------------------------------------------------------------|-----------|
| <b>II</b> | <b>Emulation zur Langzeitarchivierung digitaler Objekte</b>        | <b>39</b> |
| <b>3</b>  | <b>Digitale Archive als neuartige Aufgabenstellung</b>             | <b>41</b> |
| 3.1       | Langzeitarchivierungsstrategien . . . . .                          | 42        |
| 3.1.1     | OAIS-Referenzmodell . . . . .                                      | 42        |
| 3.1.2     | Digitalisierung analoger Medien . . . . .                          | 45        |
| 3.1.3     | Standardisierung . . . . .                                         | 46        |
| 3.1.4     | Migration und Emulation . . . . .                                  | 46        |
| 3.2       | Statische und dynamische digitale Objekte . . . . .                | 47        |
| 3.2.1     | Klassische Daten . . . . .                                         | 47        |
| 3.2.2     | Dynamische Daten, interaktive Objekte . . . . .                    | 48        |
| 3.2.3     | Betriebssysteme und Komponenten . . . . .                          | 49        |
| 3.2.4     | Anwendungsprogramme . . . . .                                      | 49        |
| 3.2.5     | Computerspiele . . . . .                                           | 49        |
| 3.2.6     | Interaktive Medien und Unterhaltung . . . . .                      | 50        |
| 3.2.7     | Datenbanken . . . . .                                              | 51        |
| 3.3       | Metadaten zur Beschreibung des Archivinhaltes . . . . .            | 51        |
| 3.3.1     | Dublin Core Metadata Set . . . . .                                 | 53        |
| 3.3.2     | PREMIS Metadata Set . . . . .                                      | 57        |
| 3.3.3     | Formate für Metadaten . . . . .                                    | 57        |
| 3.4       | Migration . . . . .                                                | 58        |
| 3.4.1     | Begriffsbestimmung . . . . .                                       | 58        |
| 3.4.2     | Migrationsstrategien für Datenformate . . . . .                    | 60        |
| 3.4.3     | Regeneration und Migration von Datenträgern . . . . .              | 61        |
| 3.4.4     | Probleme der Datentransformation . . . . .                         | 62        |
| 3.5       | Emulation . . . . .                                                | 64        |
| 3.5.1     | Ansatzpunkte . . . . .                                             | 67        |
| 3.5.2     | Nachbildung von Applikationen oder ihrer Funktionen . . . . .      | 67        |
| 3.5.3     | Emulation des Betriebssystems oder dessen Schnittstellen . . . . . | 70        |
| 3.5.4     | Emulation der kompletten Hardwareplattform . . . . .               | 71        |
| 3.5.5     | Emulation im Einsatz . . . . .                                     | 72        |
| 3.6       | Emulation und Migration im Vergleich . . . . .                     | 72        |
| <b>4</b>  | <b>Emulatoren - Recherche und Auswahlkriterien</b>                 | <b>77</b> |
| 4.1       | Recherche verfügbarer Emulatoren . . . . .                         | 79        |
| 4.2       | Klassen von Rechnerplattformen . . . . .                           | 80        |
| 4.2.1     | Frühe Großrechner . . . . .                                        | 80        |
| 4.2.2     | Der sogenannte Industriestandard: X86 . . . . .                    | 80        |
| 4.2.3     | Weitere Plattformen . . . . .                                      | 81        |
| 4.2.4     | Home-Computer . . . . .                                            | 82        |
| 4.2.5     | Spielplattformen . . . . .                                         | 82        |
| 4.3       | Erste Iteration der Emulatorensuche . . . . .                      | 83        |
| 4.3.1     | X86-Computer . . . . .                                             | 84        |
| 4.3.2     | Die ersten Consumer-Geräte: Home-Computer . . . . .                | 86        |
| 4.3.3     | Konsolenspiele und Arcade-Emulatoren . . . . .                     | 87        |

|          |                                                                  |            |
|----------|------------------------------------------------------------------|------------|
| 4.3.4    | Emulatoren der diversen Apple Rechner . . . . .                  | 88         |
| 4.3.5    | Emulatorensuche zu sonstigen Architekturen . . . . .             | 89         |
| 4.4      | Metadaten und Auswahlkriterien . . . . .                         | 90         |
| 4.4.1    | Auswahlkriterien für die digitale Langzeitarchivierung . . . . . | 91         |
| 4.4.2    | Generelle Informationen . . . . .                                | 92         |
| 4.5      | Benutzergruppen und institutionelle Rahmenbedingungen . . . . .  | 93         |
| 4.5.1    | Archive . . . . .                                                | 93         |
| 4.5.2    | National-, Universitäts- und Allgemeinbibliotheken . . . . .     | 93         |
| 4.5.3    | Forschungseinrichtungen . . . . .                                | 94         |
| 4.5.4    | Computer-, Kunst- und technische Museen . . . . .                | 94         |
| 4.5.5    | Unternehmen und Organisationen . . . . .                         | 95         |
| 4.5.6    | Privatpersonen . . . . .                                         | 96         |
| 4.6      | Nichttechnische Auswahlkriterien . . . . .                       | 96         |
| 4.6.1    | Kommerzielle oder Open Source Software . . . . .                 | 96         |
| 4.6.2    | Weitere Kriterien . . . . .                                      | 98         |
| 4.6.3    | Ökonomische Erwägungen . . . . .                                 | 99         |
| 4.7      | Technische Auswahlkriterien . . . . .                            | 99         |
| 4.7.1    | Anforderungen an das Host-System . . . . .                       | 99         |
| 4.7.2    | Datenim- und -export, Interaktion mit dem Host-System . . . . .  | 100        |
| 4.7.3    | Anforderungen seitens der Nutzungsumgebung . . . . .             | 101        |
| 4.8      | Langzeitverfügbarkeit von Emulatoren . . . . .                   | 103        |
| 4.8.1    | Softwaremigration . . . . .                                      | 104        |
| 4.8.2    | Geschachtelte Emulation . . . . .                                | 105        |
| 4.8.3    | Universal Virtual Machine . . . . .                              | 106        |
| 4.8.4    | Modulare Emulationsansätze . . . . .                             | 107        |
| <b>5</b> | <b>Hardwareemulatoren für den Langzeitzugriff</b>                | <b>109</b> |
| 5.1      | Emulation von Home-Computern . . . . .                           | 110        |
| 5.1.1    | MESS . . . . .                                                   | 110        |
| 5.1.2    | ARAnyM . . . . .                                                 | 112        |
| 5.1.3    | Home-Computer-Emulatoren auf Java-Basis . . . . .                | 113        |
| 5.2      | Virtuelle X86-Maschinen . . . . .                                | 114        |
| 5.3      | VMware - Pionier der X86-Virtualisierung . . . . .               | 115        |
| 5.3.1    | Virtualisierer für den Desktop - Workstation . . . . .           | 115        |
| 5.3.2    | Workstation 3.2 . . . . .                                        | 119        |
| 5.3.3    | Workstation 4.0 . . . . .                                        | 120        |
| 5.3.4    | Workstation 4.5 . . . . .                                        | 121        |
| 5.3.5    | Workstation 5.0 . . . . .                                        | 121        |
| 5.3.6    | Workstation 5.5 . . . . .                                        | 122        |
| 5.3.7    | Workstation 6.0 . . . . .                                        | 122        |
| 5.3.8    | VMware Player . . . . .                                          | 123        |
| 5.3.9    | VMware Server . . . . .                                          | 123        |
| 5.3.10   | VMware Fusion . . . . .                                          | 123        |
| 5.4      | Weitere kommerzielle Virtualisierer . . . . .                    | 124        |
| 5.4.1    | Parallels Workstation und Desktop . . . . .                      | 124        |

|       |                                                       |     |
|-------|-------------------------------------------------------|-----|
| 5.4.2 | Virtual PC und Server für X86 . . . . .               | 125 |
| 5.5   | Open-Source-Virtualisierer . . . . .                  | 126 |
| 5.5.1 | VirtualBox . . . . .                                  | 127 |
| 5.5.2 | DOSEMU . . . . .                                      | 130 |
| 5.6   | X86-Hardwareemulatoren . . . . .                      | 131 |
| 5.6.1 | QEMU . . . . .                                        | 131 |
| 5.6.2 | Bochs . . . . .                                       | 135 |
| 5.6.3 | Java PC Emulator - Dioscuri . . . . .                 | 135 |
| 5.6.4 | University of Oxford Java PC Emulator . . . . .       | 137 |
| 5.6.5 | DOSBOX . . . . .                                      | 139 |
| 5.7   | Emulatoren und -Virtualisierer im Vergleich . . . . . | 140 |
| 5.7.1 | Praktische Implikationen . . . . .                    | 142 |

### **III View-Path, Referenzumgebung und Softwarearchiv 145**

|          |                                                  |            |
|----------|--------------------------------------------------|------------|
| <b>6</b> | <b>View-Paths und Referenzumgebungen</b>         | <b>147</b> |
| 6.1      | Anforderungen an die Objektaufnahme . . . . .    | 149        |
| 6.2      | Objektzugriff . . . . .                          | 151        |
| 6.2.1    | Zugriff auf nicht-archivierte Objekte . . . . .  | 153        |
| 6.3      | Der Weg zum Objekt . . . . .                     | 153        |
| 6.4      | Aggregation von View-Paths . . . . .             | 156        |
| 6.4.1    | Ökonomie und Aufwand . . . . .                   | 157        |
| 6.5      | Referenzumgebungen . . . . .                     | 159        |
| 6.5.1    | Lokale Workstations . . . . .                    | 160        |
| 6.5.2    | Virtuelle Maschinen . . . . .                    | 162        |
| 6.5.3    | Zentralisierte Systeme mit Fernzugriff . . . . . | 163        |
| 6.5.4    | Problemfelder und Herausforderungen . . . . .    | 164        |
| 6.6      | Vorbereitete Umgebungen . . . . .                | 166        |
| 6.6.1    | Kombinierte Primärobjekte . . . . .              | 167        |
| 6.6.2    | Thematische Zusammenfassung . . . . .            | 168        |
| 6.7      | View-Paths im Zeitablauf . . . . .               | 169        |
| 6.7.1    | Archivtypen . . . . .                            | 171        |
| 6.7.2    | Nutzergruppen . . . . .                          | 171        |
| 6.7.3    | Langfristige Perspektive . . . . .               | 173        |
| 6.8      | Wege des Datenaustauschs . . . . .               | 174        |
| 6.8.1    | Netzwerkprotokolle . . . . .                     | 175        |
| 6.8.2    | Disketten-Images . . . . .                       | 176        |
| 6.8.3    | Virtuelle optische Wechseldatenträger . . . . .  | 177        |
| 6.8.4    | Virtuelle Festplatten . . . . .                  | 178        |
| 6.8.5    | Spezielle Datentransportpfade . . . . .          | 179        |



|           |                                                                 |            |
|-----------|-----------------------------------------------------------------|------------|
| <b>7</b>  | <b>Softwarearchiv der Sekundärobjekte</b>                       | <b>181</b> |
| 7.1       | View-Paths und benötigte Software . . . . .                     | 182        |
| 7.2       | Bestandteile eines Emulationssoftwarearchivs . . . . .          | 184        |
| 7.2.1     | Art der Ablage . . . . .                                        | 185        |
| 7.2.2     | Softwarevarianten . . . . .                                     | 186        |
| 7.3       | Applikationen . . . . .                                         | 187        |
| 7.3.1     | Viewer und Editoren . . . . .                                   | 187        |
| 7.3.2     | Skriptsprachen und Runtime Environments . . . . .               | 189        |
| 7.3.3     | Spiele . . . . .                                                | 189        |
| 7.4       | Betriebssysteme . . . . .                                       | 190        |
| 7.5       | Hardware- und Softwaretreiber . . . . .                         | 190        |
| 7.5.1     | Benötigte Treiber . . . . .                                     | 191        |
| 7.5.2     | Hardwaretreiber kommerzieller Betriebssysteme . . . . .         | 193        |
| 7.5.3     | Quellen und Beschaffung . . . . .                               | 193        |
| 7.5.4     | Hardwaretreiber in Open-Source-Betriebssystemen . . . . .       | 195        |
| 7.6       | Emulatoren . . . . .                                            | 195        |
| 7.7       | Hilfsprogramme . . . . .                                        | 196        |
| 7.7.1     | Softwarebibliotheken für verschiedene Aufgaben . . . . .        | 197        |
| 7.7.2     | Host-Plattform . . . . .                                        | 197        |
| 7.7.3     | Nutzungsumgebung . . . . .                                      | 198        |
| 7.8       | Archivmanagement . . . . .                                      | 199        |
| 7.9       | Verteilung und Redundanz . . . . .                              | 200        |
| 7.10      | Archivierung von Datenträgern . . . . .                         | 202        |
| 7.10.1    | Rechtzeitige Datenträgermigration . . . . .                     | 203        |
| 7.10.2    | Dateien direkt auslesen . . . . .                               | 205        |
| 7.10.3    | Virtuelle Datenträger . . . . .                                 | 206        |
| 7.10.4    | Installationsdisketten . . . . .                                | 207        |
| 7.10.5    | Optische Datenträger . . . . .                                  | 208        |
| 7.10.6    | Virtuelle Festplatten . . . . .                                 | 209        |
| 7.10.7    | Problemstellungen . . . . .                                     | 210        |
| 7.11      | Zugriffssteuerung und Rechtemanagement . . . . .                | 211        |
| 7.11.1    | Schutzvorrichtungen auf Datenträgerebene . . . . .              | 212        |
| 7.11.2    | Schutzvorrichtung an Software und statischen Objekten . . . . . | 213        |
| 7.11.3    | Rechtemanagement im Softwarearchiv . . . . .                    | 213        |
| <b>8</b>  | <b>Fazit</b>                                                    | <b>215</b> |
| 8.1       | Grenzen und neue Herausforderungen . . . . .                    | 220        |
| <b>IV</b> | <b>Anhang</b>                                                   | <b>223</b> |
| <b>A</b>  | <b>Anhang</b>                                                   | <b>225</b> |
| A.1       | Emulator-Recherche: Erste Ergebnisse . . . . .                  | 225        |
| A.1.1     | Großrechner . . . . .                                           | 226        |
| A.1.2     | Home-Computer . . . . .                                         | 226        |

|          |                                                              |            |
|----------|--------------------------------------------------------------|------------|
| A.1.3    | Konsolen und Arcade Emulatoren . . . . .                     | 228        |
| A.1.4    | Emulatorensuche zu sonstigen Ansätzen . . . . .              | 229        |
| A.2      | Begleitende Experimente . . . . .                            | 230        |
| A.3      | Spezielle virtuelle Multi-Installationsdatenträger . . . . . | 231        |
| A.3.1    | Multi-Installations-ISO . . . . .                            | 231        |
| A.3.2    | Installation aus einer Netzwerkquelle . . . . .              | 232        |
| A.4      | Anwendung: Computerspiele-Museum . . . . .                   | 234        |
| A.4.1    | Experimente und Setups . . . . .                             | 235        |
| A.4.2    | Erkenntnisse . . . . .                                       | 236        |
| A.5      | Datenträger . . . . .                                        | 238        |
| A.5.1    | Computerdisketten . . . . .                                  | 239        |
| A.5.2    | Magnetfestplatten . . . . .                                  | 240        |
| A.5.3    | Optische Datenträger . . . . .                               | 241        |
| A.5.4    | Weitere Datenträger . . . . .                                | 244        |
| <b>B</b> | <b>Glossar</b>                                               | <b>245</b> |
|          | <b>Literaturverzeichnis</b>                                  | <b>263</b> |

## Zusammenfassung

Die Langzeitverfügbarkeit von Wissen und Informationen in digitaler Form stellt die Menschheit vor neuartige Herausforderungen. Digitale Objekte erweisen sich, anders als klassische Informationsträger, 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 oder ihre Nutzungsumgebung erhalten bleiben.

Eine bekannte Methode zur Langzeitarchivierung digitaler Objekte besteht in der Migration. Sie erlaubt es, in jeweils aktuellen Arbeitsumgebungen Objekte zu betrachten, die zu diesem Zeitpunkt auf schon längst nicht mehr verfügbaren Rechnerplattformen erstellt wurden. Diese Herangehensweise eignet sich jedoch nicht uneingeschränkt für alle Objekttypen und ist besonders auf dynamische digitale Objekte nicht anwendbar. Zudem treten auch bei klassischen statischen Objekten Probleme, wie die Frage nach der Authentizität auf. Letztere stellt jedoch ein zentrales Moment vertrauenswürdiger Langzeitarchive dar.

Deshalb ist es sinnvoll, Langzeitarchivierungsverfahren um die Komponente der Emulation von Ablaufumgebungen zu ergänzen. Die Verfolgung einer Emulationsstrategie bedarf einer ganzen Reihe von Überlegungen. So ist bereits beim Einstellen digitaler Archivalien in ein Langzeitarchiv festzustellen, welche Art von Ablaufumgebung und welche Zusatzinformationen für spätere Zugriffe benötigt werden. Diese Informationen sollten sich anhand der Objektmetadaten beispielsweise unter Zuhilfenahme von Format-Registries ermitteln lassen.

Ausgehend vom jeweiligen Objekttyp und dessen Anforderungen ergibt sich ein "View-Path", der sich aus Komponenten wie Erstellungssapplikation, benötigtes Betriebssystem, dafür geeigneter Emulator und geeignete Referenzplattform zusammensetzt. Während Objekt, Applikation oder Betriebssystem gegebenermaßen weitgehend festliegen, sind Emulatoren, welche die Schnittstelle zwischen dem gewünschten Objekt und dem jeweils aktuellen technologischen Stand bilden, anhand zu bestimmender Kriterien auszuwählen. Es existiert nicht die eine Emulationsstrategie, sondern es sind verschiedene Konzepte und Wege, allein aus Gründen der Zugriffssicherung und Risikoverteilung, anzustreben.

Für eine gesicherte Anwendung der Emulation in der Langzeitarchivierung gilt es deshalb, die notwendigen Erfolgsbedingungen zu ermitteln: Neben dem eigentlichen, dem primären Objekt und seinen typischen Metadaten kann es erforderlich sein, ein sekundäres Softwarearchiv zu betreiben. Dieses beinhaltet die vom View-Path her vorgegebene und für die jeweils aktuelle Referenzumgebung geeignete Emulatoren. Neben den für den Datentransport verwendeten Hilfsprogrammen speichert das Archiv alle für die geforderten View-Paths benötigte Software, wie Betriebssysteme und deren Treiber sowie Applikationen und Patches.

Daneben können im Archiv bereits vorbereitete Pakete, sogenannte Caches, für besonders häufig angeforderte Darstellungspfade vorgehalten werden. Diese Pakete eignen sich wegen ihrer Kompaktheit zudem zum einfachen Austausch über verschiedene Gedächtnisorganisationen hinweg. Zudem lässt sich ein solches Archiv um Komponenten der automatischen Datenverteilung oder ein Rechtemanagement und Zugriffskontrollsystem ergänzen. Ein an die Archivobjekte, sowohl primäre als auch sekundäre, gebundenes Rechtemanagement sollte auf jeden Fall vermieden werden.

In Abhängigkeit von der Archivbenutzergruppe, des Alters einer wiederherzustellenden Umgebung oder dem gewählten Emulator sind ausreichende Betriebs- und Bedienungsinformationen vorzuhalten und gegebenenfalls laufend anzupassen. So haben Institutionen, Endanwender zu Hause, Nationalarchive, Staats- und Universitätsbibliotheken oder technische Museen durchaus sehr unterschiedliches Betriebswissen und heterogene Anforderungen. Aus diesen Überlegungen ergeben sich sinnvolle Erweiterungen für das OAIS-Referenzmodell.

## Danksagung

Mein besonderer Dank gilt Herrn Prof. Dr. G. Schneider für die Aufnahme als Assistent an seinem Lehrstuhl und das dortige gute Arbeitsklima. Ich habe die weitgehenden Freiheiten bei der Wahl des Themas und in der Bearbeitungszeit sehr genossen. Im Laufe der Beschäftigung mit der Thematik bildete sich eine kleine Arbeitsgruppe zur funktionalen Langzeitarchivierung am Lehrstuhl und Rechenzentrum heraus, die sich am "PLANETS Project" beteiligte. Dadurch erhielt sie eine Förderung durch die Europäische Kommission im Rahmen des "Framework Programme 6". Diese Unterstützung ermöglichte den intensiven Austausch mit internationalen Partnern und gab wichtige Anregungen für die inhaltliche Weiterentwicklung. Gleichfalls nicht unerwähnt bleiben sollte die Media AG des Nestor Expertennetzwerks. Sie hat mir einige sehr wertvolle Kontakte und Anregungen verschafft, die zur Vervollständigung der Arbeit beigetragen haben.

Ebenso geht mein Dank an die Arbeitsgruppe am Lehrstuhl für den intensiven Gedankenaustausch und die Entlastungen zum Ende der Promotion. Gleichzeitig nicht versäumen möchte ich, mich bei Herrn Dr. D. Degenhardt für die Tipps für Verbesserungen und hilfreichen Anmerkungen in den verschiedenen Zwischenphasen zu bedanken. Darüber hinaus gilt ein sehr herzlicher Dank Anja und meinen Eltern für die Geduld während der Erstellung des Werks und das kritische Gegenlesen kurz vor seiner Vollendung.

Keinesfalls vergessen möchte ich Herrn Prof. Dr. E. Mittler, der dankenswerter Weise die Rolle des Zweitgutachters meiner Arbeit angenommen hat. Mit ihm konnte ich einen Fachmann aus dem Bibliothekswesen gewinnen, der sich als langjähriger Direktor der Staats- und Universitätsbibliothek in Göttingen mit der Thematik der digitalen Langzeitarchivierung bereits sehr früh beschäftigt hat.

## Teil I

# Überblick und Motivation der Arbeit



# 1 Einleitung

Fast kaum ein Vorgang der heutigen Informationsgewinnung, -speicherung und -übermittlung kommt ohne elektronische Hilfsmittel aus. Die Informationstechnik, die so den Wissensumsatz beschleunigt und ganz neue Formen von Arbeit und Kooperation erlaubt, ist auch nach 50 Jahren noch wenig auf Nachhaltigkeit ausgerichtet: Sie bringt durch ihre rasante Weiterentwicklung ihre eigenen Produkte in Gefahr.

Diese Gefahr wird seit der Jahrtausendwende zunehmend wahrgenommen. Deshalb gibt es inzwischen eine ganze Reihe von Initiativen und Bestrebungen gerade von *Gedächtnisorganisationen*,<sup>1</sup> die sich zum Ziel gesetzt haben, den permanenten Zugriff auf die digitale Wissensbasis und das elektronische kulturelle Erbe sicherzustellen. Die meisten der in diesem Zusammenhang gewählten Strategien, *digitale Objekte* mit der technischen Entwicklung mitzuführen und für den permanenten Zugriff bereitzuhalten, setzen auf *Migration*. Für Datenträger, üblicherweise das simple Umkopieren der Bitströme auf das jeweils aktuelle Speichermedium, stellt sie das Mittel der Wahl dar. Jedoch muss auch der Inhalt der Bitströme für die jeweils aktuelle Form des Zugriffs angepasst werden. Diese Art der Migration impliziert die (wiederholte) Veränderung des digitalen Objekts selbst.

Die in Betracht gezogenen Strategien der funktionalen Langzeitarchivierung sollten deshalb Komponentenbündel enthalten, die jederzeit einen Zugriff auf das Objekt in seinem Originalzustand erlauben. Das kann bedeuten, dass diese Ergänzungen überhaupt erst eine spätere Darstellung der Objekte gewährleisten. Emulation hat eine Wiederherstellung einer Entstehungsumgebung zum Ziel, in der ein Objekt ursprünglich erzeugt wurde oder primär ablief. *Emulation* verhindert auf diese Weise, dass die *Primärobjekte* eines *digitalen Langzeitarchivs* verändert werden müssen.

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 aufgehoben werden muss, benötigt man ein Archiv an Werkzeugen und geeignete *Referenzumgebungen* 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.

---

<sup>1</sup>Spezielle, im Zusammenhang dieser Arbeit relevante Fachbegriffe, die im Glossar am Ende des Anhangs näher erläutert werden, sind kursiv gesetzt.

Damit ein solches Archiv seine Aufgaben erfüllen kann, sind einige Randbedingungen einzuhalten. Mit der Einstellung digitaler Objekte in ein Langzeitarchiv ergeben sich Anforderungen an deren Wiedergabe, denen seitens des Managements des Softwarearchivs Rechnung getragen werden muss. Ein solches Archiv muss geeignet bestückt, gepflegt und regelmäßig aktualisiert werden. Ein Teil der Archivobjekte, die Emulatoren, sind ebenso von Veraltung betroffen, wie alle anderen Primärobjekte auch. Sie spielen jedoch eine kritische Mittlerrolle als Schnittstelle zum jeweils gültigen Stand der Technik: Digitale Objekte aus einem Langzeitarchiv sind nur dann sinnvoll für die Anwender ihrer jeweiligen Zeit nutzbar, wenn sie in geeigneter Weise in der zu der Zeit verfügbaren technischen Umgebung zugreifbar sind.

## 1.1 Motivation

Das Gedächtnis der menschlichen Spezies speist sich aus Überlieferung. Die heutige Kultur ist nicht ohne ihre Wurzeln vorstellbar. Die Ursprünge reichen bis weit vor unsere Zeitrechnung zurück. So besitzt beispielsweise die europäische Kultur starke antike und christlich-religiöse Fundamente. Das heutige Wissen speist sich aus dem fortgesetzten Umsatz von Information. Dabei kommt es insbesondere darauf an, dieses Wissen nicht nur im relativ kleinen persönlichen Kreis weiterzugeben, sondern dieses auch über Zeitspannen zu transportieren, die über das eigene Leben und das mehrerer Generationen weit hinausgehen. Eine Verbreitung von Information über größere Räume hinweg konstituiert ein wesentliches Element der Vergrößerung der Wissensbasis.

Für die Verbreitung von Wissen über Raum und Zeit werden Informationsträger benötigt. Die Entwicklung der menschlichen Kultur und ihrer Befähigung zum Informationsumsatz sind eng mit diesen Trägern verbunden. Bereits die Bewahrung früher und mittelalterlicher Schriften gelang in vielen Fällen nicht ohne Auffrischung. Man kann immer noch die in Stein gehauenen Hieroglyphen auf der Stele des Hammurabi und als Wandinschriften in diversen Grabkammern oder Tempeln entziffern. So lassen sich heute noch über 4000 Jahre alte Pyramiden mit ihren Inschriften bewundern, die von der Macht damaliger Herrscher in Ägypten künden. Die notwendige Bedingung, dass man etwas über die "Heldentaten" eines Königs, das Alltagsleben seiner Untertanen erfährt, liegt darin begründet, dass es zum großen Teil gelungen ist, die Bedeutung der Zeichen der damaligen Schriftsprache zu entschlüsseln.

Die ursprünglichen Papyri<sup>2</sup> oder Tierhäute, auf denen Texte vor mehreren tausend Jahren niedergeschrieben wurden, sind heute oft nicht mehr existent oder kaum mehr entzifferbar. In diesen Fällen musste schon seit früher Zeit durch wiederholtes Umkopieren für den Fortbestand der Information gesorgt werden. Die Aufgabe des Erhaltens übernahmen beispielsweise im Europa des Mittelalters die Klöster. In diesen waren viele Mönche mit nichts Anderem beschäftigt, als Bücher "abzuschreiben" oder zu übersetzen, um so das gesammelte Wissen zu bewahren und mit den Mitteln der jeweiligen Epoche zugänglich zu halten. Nicht zu unrecht bemaß sich der Reichtum der Klöster am Umfang und Inhalt ihrer Bibliotheken.

Mit der Zunahme des Wissens reicht dessen lediglich mündliche Überlieferung nicht mehr aus. Sind am Prozess der Weiterverbreitung und Übermittlung nur wenige Individuen beteiligt, steigt zudem das Risiko der Verfälschung oder des Totalverlustes.

---

<sup>2</sup>Papyrus als Vorläufer des Papiers im Altertum, siehe hierzu auch den Überblick "Datenspeicher: Das Gedächtnis der Kultur" in [Spilker 2007].



Nicht jede kulturelle Äußerung hat sich für zukünftige Generationen erhalten lassen. Das hängt durchaus von den Möglichkeiten des jeweils erreichten Stands der technischen Entwicklung ab. Jede Art der Darstellung, jede Form der Information verlangt nach durchaus sehr unterschiedlichen Strategien der Konservierung.

Im Grunde hat sich an der Jahrhunderte alten Fragestellung nicht viel geändert: Auch in unserer Epoche geht es um die geeignete Bewahrung von und die langfristige Zugreifbarkeit auf Information, selbst wenn sich mit den Möglichkeiten moderner Technik die Art der Lösung dieses Problems verändert hat.

## 1.2 Einordnung und Fokus der Fragestellung

Die Zahl der Akteure auf dem Gebiet der Langzeitarchivierung digitaler Objekte ist im Laufe der letzten zehn Jahre deutlich angestiegen. Inzwischen bestehen etliche Initiativen zum Erhalt des elektronischen kulturellen Erbes. Die zunehmende Spezialisierung hat eine breite Palette an Fragestellungen<sup>3</sup> identifiziert, die sich zur Einordnung dieser Arbeit wie folgt gliedern lassen:

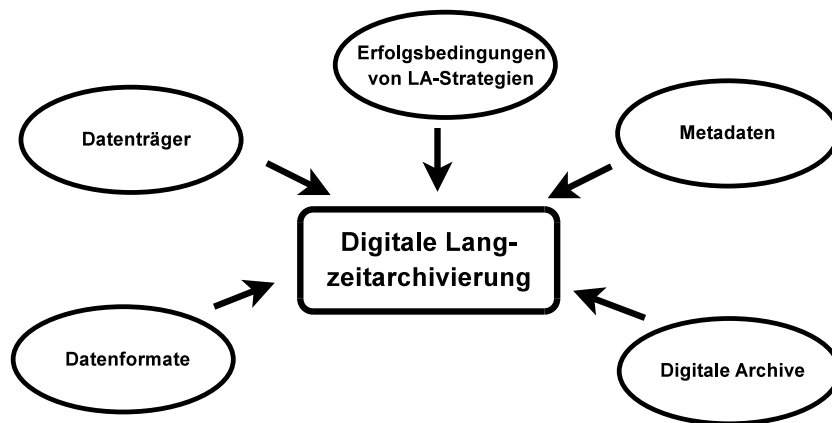
- **Datenträger** - Für die mittel- und langfristige Datenablage zu archivierender Objekte müssen geeignete Methoden gefunden werden, die über einen ausreichend langen Zeitraum Informationen stabil erhalten können. In diesem Bereich existieren erfolgreiche Ansätze wie DIAS oder Kopal.<sup>4</sup> Daher befasst sich diese Arbeit mit Datenträgern lediglich an Stellen, wo diese als Installationsmedien von *Software* eine Rolle spielen und geeignet archiviert oder virtualisiert werden sollten.
- **Datenformate** - Aufgrund ihrer unüberschaubaren Anzahl sind *Datenformate* zur längerfristigen Nutzbarkeit auf Standardisierung angewiesen. Denn mit der Festlegung im Sinne der Langzeitbewahrung geeigneter Datenformate für eine Vielzahl digitaler Objekte steigt die Chance, dass entsprechende Software zur Interpretation dieser Formate zu späteren Zeitpunkten noch erhältlich ist oder aufgrund des hohen Datenbestandes in den Archiven verfügbar gehalten wird. In diesem Bereich wurden Vorschläge wie das ODF<sup>5</sup> gemacht und als Standards aufgenommen. Solche Bemühungen wirken sich erst langsam auf zukünftige Archivbestände aus und sind daher für die an dieser Stelle vorgenommenen Betrachtungen weniger relevant.
- **Metadaten** - Digitale Objekte erklären sich vielfach nicht aus sich heraus. Oft benötigt man eine Reihe von Zusatzinformationen. Es existieren inzwischen einige gute Vorschläge für Metadatenschemata. Sie sind unter dem Aspekt der geeigneten Erweiterbarkeit für die Objekte des später erörterten Softwarearchivs zu erwähnen.

<sup>3</sup>Siehe beispielsweise die in [Vaughn u. a. 2000], S. 176 ff. in "Preservation Challenges" getroffene Unterteilung.

<sup>4</sup>DIAS-Projekt der Königlichen Bibliothek der Niederlande, siehe [van Diessen und Steenbakkers 2002], Kopal [kopal 2008]. Einen kompakten Überblick zu Forschungsansätzen und Initiativen bietet darüberhinaus Abschnitt 2.6 in Kapitel 2.

<sup>5</sup>Open Document Format – unbeschränkt offener Standard (ISO/IEC 26300:2006) für sogenannte Office-Dokumente, siehe [http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=office](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=office).

- Digitales Archiv - Es wird zu erörtern sein, wie ein Archiv digitaler Objekte aufgebaut sein sollte und welche zusätzlichen Inhalte, *Sekundärobjekte*, für zukünftige Zugriffe benötigt werden.
- Langzeitarchivierungsstrategien und ihre Erfolgsbedingungen - Hierfür ist zu klären, welche Maßnahmen und Arbeitsabläufe notwendig sind, um ein vertrauenswürdiges Archiv<sup>6</sup> zu betreiben und was bei der Objektaufnahme und späteren Rückgewinnung beachtet werden sollte.
- Standards - Sie helfen bei der Strukturierung der Problemstellung und dem Austausch zwischen Institutionen. Sie spiegeln sich in Referenzmodellen und Metadatenschemata wieder.



**Abbildung 1.1:** Das Problem der Langzeitarchivierung (LA) digitaler Objekte verschiedenen Typs lässt sich in eine Reihe von Teilfragestellungen ordnen.

Generell bedarf es einer Vervollständigung der bisherigen Ansätze, um es zu erlauben, digitale Objekte in ihrem Erstellungskontext zu behandeln. Dieser Kontext meint die Software- und Hardwarekonfiguration, in der das jeweilige Objekt erstellt oder zuletzt sinnvoll bearbeitet wurde. Diese Anforderungen erweitern das Archivkonzept und bringen eine Reihe neuer Aufgabenstellungen mit sich. Ausgehend von bereits bestehenden Überlegungen erfolgt im Rahmen dieser Arbeit eine Untersuchung und die Bereitstellung von Ablaufumgebungen. Für diese sind ebenso wie die ursprünglich zu erhaltenden Objekte Vorkehrungen zur langfristigen Fortschreibbarkeit zu treffen. Der Schwerpunkt dieser Arbeit liegt deshalb auf den Erfolgsbedingungen der Langzeitarchivierung einer Vielzahl digitaler Objekte und der Untersuchung, welche zusätzlichen Maßnahmen notwendig sind, um dieses zu erreichen. Hierzu werden Aufgaben und Anforderungen eines Archivs zur Emulation von *Rechnerplattformen* diskutiert.

Während die Anforderungen an Migrationsstrategien weitgehend bestimmt und die notwendigen Abläufe und Archivmanagementprozesse Untersuchungsgegenstand<sup>7</sup> vieler Projekte und Initiativen sind, vermisst man diese Breite zur Emulation. In der bisherigen Forschung

<sup>6</sup>Vgl. hierzu die Nestor-Schrift [nestor 2006].

<sup>7</sup>Siehe 2.6 zu bisherigen Arbeiten und Untersuchungen - einen allgemeinen Überblick bringt Kapitel 2. Migration und Emulation werden in weiteren Abschnitten 3.4 und 3.5 im dritten Kapitel gewürdigt.

finden die speziellen Bedingungen dieses Zweigs der Langzeitarchivierungsstrategien noch wenig Aufmerksamkeit. Da sich die Emulation durchaus wesentlich in ihrer Handhabung, ihren Arbeitsabläufen und Erfolgsbedingungen von Migrationsprozessen unterscheidet, unternimmt diese Arbeit eine breiter angelegte Untersuchung zu Einsatz und Randbedingungen.

Die Verfolgung der Emulationsstrategie erfordert demnach eine ganzen Reihe von ergänzenden Überlegungen. Bezogen auf den Archivbetrieb, beispielsweise nach dem *OAIS-Modell*,<sup>8</sup> unterscheiden sie sich in vielen Bereichen grundlegend vom Migrationsansatz:

- Emulation operiert nicht am Objekt selbst, sondern arbeitet an seiner Umgebung. Sie kann aus diesen Gründen eine sehr breite Palette digitaler Objekttypen abdecken. Dabei kann man nicht von der einen Strategie sprechen, sondern von verschiedenen Ansätzen mit derselben Zielrichtung.
- So ist bereits bei der Einstellung digitaler *Archivalien* in ein Langzeitarchiv festzustellen, welche Art von Ablaufumgebung benötigt wird. Diese Information sollte sich anhand der Objektmetadaten, beispielsweise unter Zurhilfenahme von Format-Registries ermitteln lassen. Dabei können verschiedene Wege zur Darstellung existieren, die allein aus Gründen der Zugriffssicherheit vorgehalten werden könnten.
- Ausgehend vom jeweiligen Objekt und dessen Anforderungen ergibt sich ein "View-Path", der sich aus Komponenten, wie Erstellungs-Applikation, benötigtes *Betriebssystem*, dafür geeigneter *Emulator* und geeignete Host-Plattform zusammensetzt. Wo Objekt, Applikation oder Betriebssystem gegebenermaßen weitgehend festliegen, sind Emulatoren anhand zu bestimmender Kriterien auszuwählen. Letztere bilden die Schnittstelle zum gewünschten Primärobjekt und dem jeweils aktuellen technologischen Stand, manifestiert durch die jeweilige Referenzumgebung.

Es reicht jedoch nicht, lediglich die ausgewählten Emulatoren zu archivieren. Diese müssen mindestens um Hilfsmittel für den Datentransport in und aus der Ablaufumgebung ergänzt werden. Neben dem eigentlichen, dem primären Objekt und seinen typischen Metadaten wird es deshalb erforderlich sein, ein sekundäres Softwarearchiv zu betreiben. Dieses beinhaltet die vom View-Path her vorgegebene und für die jeweils aktuelle *Nutzungsumgebung* geeignete Emulatoren. Zusätzlich zu den für den Datentransport verwendeten Hilfsprogrammen nimmt das Archiv alle für die geforderten View-Paths benötigte Software, Treiber und Patches auf. Dabei können im Archiv bereits vorbereitete Pakete für besonders häufig angeforderte View-Paths vorgehalten werden. Diese Pakete eignen sich wegen ihrer Kompaktheit zudem zum einfachen Austausch über verschiedene Gedächtnisorganisationen hinweg. Zudem lässt sich ein solches Archiv um Komponenten der automatischen Datenverteilung oder ein Rechtemanagement und Zugriffskontrollsystem ergänzen. Ein an die Archivobjekte selbst gebundenes Rechtemanagement ist kritisch zu hinterfragen.

In Abhängigkeit von der Archivbenutzergruppe, des Alters einer wiederherzustellenden Umgebung oder des gewählten Emulators sind ausreichende Betriebs- und Bedienungsinformationen vorzuhalten und gegebenenfalls laufend anzupassen. Jedoch haben Endanwender zu

---

<sup>8</sup>Open Archival Information System - Zentrales Referenzmodell für elektronische Langzeitarchive, siehe dazu [CCSDS 2002] und Abschnitt 3.1 im weiteren Teil der Arbeit.

Hause, Nationalarchive, Staats- und Universitätsbibliotheken oder technische Museen durchaus sehr unterschiedliches Betriebswissen und heterogene Anforderungen.

Hierzu greift die Arbeit auf bereits vorhandene Ansätze zurück, versucht diese zu strukturieren und weiterzuentwickeln. Das Ziel ist die Erstellung einer Handlungsanleitung im Sinne von Best-Practice-Beispielen. Zudem werden Elemente einer Softwarebibliothek für "digitale Archivare" diskutiert. Deshalb beschäftigt sich die Arbeit außerdem mit:

- Konzepten digitaler Daten - Zu verschiedenen digitalen Objekten sind unterschiedliche Umgebungsbedingungen zu schaffen, um diese wieder zugreifbar zu machen.
- Softwarearchiven - Es soll ein Archiv für Computersoftware der unterschiedlichsten Rechnerarchitekturen diskutiert werden. In diesem erfolgt die Bewahrung von Programmen mit denen digitale Objekte ursprünglich erstellt wurden oder weitgehend optimal wiedergegeben werden können. Dieses erstreckt sich nicht nur auf die Applikation direkt, sondern muss die gesamte Nutzungsumgebung erfassen.
- Benutzerschnittstellen - Da ein Softwarearchiv für Emulation Endbenutzern die Möglichkeit geben könnte, ihre Daten selbst anzusehen oder sogar zu transformieren und zu bearbeiten, werden geeignete, leicht bedienbare Schnittstellen benötigt.
- Räumlicher Distribution und Redundanz von Archivdaten - Das benötigte Softwarearchiv umfasst viele Komponenten, die in mehreren Gedächtnisorganisationen erforderlich sind.

Diese Arbeit entstand im Rahmen der Arbeitsgruppe Langzeitarchivierung am Lehrstuhl für Kommunikationssysteme des Instituts für Information und des Rechenzentrums der Universität Freiburg. Die Arbeitsgruppe ist eingebettet in das PLANETS Project<sup>9</sup> und kooperiert mit der Nestor Media AG.<sup>10</sup> Sie arbeitet in den Teilprojekten "Preservation Action" insbesondere im Bereich "Strategy Development" und "Tools for Environments" mit und macht Vorschläge zur Einbettung von Emulation in das "Interoperability Framework"<sup>11</sup> Sie sieht sich als Ergänzung und im gewissen Sinne als Impulsgeberin zur Erarbeitung von Teilstrategien und Erfolgsbedingungen einer Langzeitarchivierung in erster Linie dynamischer Objekte.

Hierzu beschäftigt sich die vorliegende Arbeit mit den Randbedingungen der erfolgreichen Anwendung der Emulationsstrategie und beleuchtet dazu aus den eben genannten Punkten die Auswahlkriterien von Emulatoren unter dem Fokus der Langzeitarchivierbarkeit, die Bewertung von *Emulatoren* und *Virtualisierern* anhand dieser Kriterien, die Entwicklung und Auswahl von View-Paths und den hierfür notwendigen Sekundärobjekten. Hierzu wird ein Beispielarchiv diskutiert, welches die notwendigen Elemente enthält, um für eine Auswahl populärer Systeme Ablaufumgebungen für Endbenutzer anzubieten.<sup>12</sup> Die Überlegungen in

<sup>9</sup>EU-gefördertes Projekt zur Langzeitarchivierung, [PLANETS 2008]. Es wird unter dem "Information Society Technologies (IST) Programme" des Framework 6 anteilig finanziert (Project IST-033789).

<sup>10</sup>BMBF-geförderte Initiative Nestor [nestor 2008]. Siehe zudem Abschnitt 2.6 zu weiteren Aktivitäten auf dem Gebiet.

<sup>11</sup>Vgl. hierzu Ausführungen zu PLANETS in Abschnitt 2.6.2.

<sup>12</sup>Aufgrund des schieren Umfangs vergangener und aktueller *Rechnerarchitekturen* und der darauf lauffähigen Software kann es diese Arbeit jedoch nicht leisten, ein vollständiges Archiv zu erstellen.

einer weiteren Arbeit [Welte 2008] setzen am View-Path an und diskutieren speziell die Optionen eines Remote-Zugriffs auf Emulationsdienste. Dieses bietet eine Reihe von Vorteilen gegenüber lokalen Referenzumgebungen.

Die beiden Untersuchungen haben zum Ziel, denkbare Ergänzungen und Erfolgsbedingungen des OAIS aufzuzeigen. Sie stellen dazu einige ergänzende Überlegungen für den Zugriff auf die primären Archivobjekte vor. Dabei sind die Bestandteile eines Softwarewerkzeugkastens nicht vom OAIS isoliert.

## 1.3 Kapitelaufteilung und Struktur der Arbeit

Die Arbeit selbst wendet sich primär nicht nur an das reine Fachpublikum aus der Informatik, sondern bezieht wegen der Art der Fragestellung Archivare und Bibliothekare ein. Deshalb fallen Erläuterungen und Überblicksdarstellungen ausführlicher aus, so dass auch fachfernere Personen die Darstellung der Problematik und ihrer Erkenntnisse nachvollziehen können.

Das anschließende Kapitel dient der Motivation und der Begründung der Fragestellung. Es gibt einen kurzen Überblick über den kulturellen Hintergrund, den Anspruch an die Archivierung von Informationen und die Weitergabe von Wissen. Dazu zählt die Darstellung der Rolle des Computers und die Beschäftigung der mit ihm erstellten Objekte. Digitale Objekte weisen besondere innere Konzepte und Strukturen auf, die ihre Anforderungen aber auch Problemstellungen für eine Langzeitarchivierung begründen. In diesem Zuge wird ein Überblick zu verschiedenen Datentypen und ihren Eigenschaften gegeben. Weitere Fragen beschäftigen sich mit der geordneten Ablage von elektronischen Daten in Dateien und Verzeichnissen in Dateisystemen. Datenträger als physische Ablagemedien spielen in weiteren Teilen dieser Arbeit eine Rolle als Peripherie von Rechnerplattformen und als virtuelle Medien für Emulatoren. Das Kapitelende verschafft eine Einordnung der Arbeit in aktuelle Diskussionen und Entwicklungen. Hierzu stellt es einen Überblick zu bisherigen Gedanken und Forschungsansätzen zusammen und gibt Auskunft welche nationalen, europäischen sowie internationalen Akteure das Feld bestimmen.

Das folgende Kapitel bildet den Auftakt des zweiten Teilabschnitts, der sich mit der Emulation als Langzeitarchivierungsstrategie digitaler Objekte befasst. Hierfür stehen zu Beginn Langzeitarchivierungsstrategien ganz allgemein im Fokus: Was sind die verschiedenen Anforderungen an eine Langzeitarchivierung digitaler Daten und welche Ziele und Zielkonflikte gibt es? Weshalb werden bisher in erster Linie statische Daten betrachtet und welche Versuche gibt es zur Einflussnahme auf zukünftige Datenformate? Welche Metadaten-Systeme zur Beschreibung von Objekten kommen zum Einsatz? An dieser Stelle spielt das OAIS-Referenzmodell als zentraler Bezugspunkt der überwiegenden Überlegungen und Forschungen eine Rolle. Nach der abstrakteren Darstellung der Herausforderungen im Kapitel 2 erfolgt nun eine eingehendere Untersuchung und Klassifizierung verschiedener Objekttypen und die Herausstellung der speziellen Fragestellungen dynamischer Objekte. Auf dieser Basis werden Migration und Emulation untersucht und Überlegungen dargelegt, welche Langzeitstrategien für Emulatoren existieren. Beide Strategien spielen für die Fragestellung dieser Arbeit eine wichtige Rolle, wobei motiviert durch das vorangegangene Kapitel die Emulation besonders herausgehoben wird. Hierbei stehen die Chancen der Emulation von Rechnerarchitekturen im Mittelpunkt.

Nach den theoretischen Vorbetrachtungen zur Emulation stellt das Kapitel 4 Auswahlkriterien für Emulatoren als Werkzeuge in der Langzeitarchivierung zusammen. An dieser Stelle spielen sowohl technische, als auch eine Reihe nichttechnischer Kriterien eine Rolle. Neben Metadaten zu einzelnen Programmen und Produkten sind insbesondere die Anwender in der Gestalt verschiedener Gedächtnisorganisationen, Privatpersonen oder kommerzieller Einheiten von Interesse. Der Suchraum der Werkzeuge wird einerseits durch mögliche Ansatzpunkte von Emulation, wie in Kapitel 3 eingeführt, und andererseits durch das Ergebnis von Recherchen nach bereits vorhandenen Lösungen, eingegrenzt.

Das fünfte Kapitel nimmt das Thema des vorangegangenen auf und stellt eine Liste von geeigneten Emulatoren für den Langzeitzugriff auf Primärobjekte zusammen. Dabei erfolgt eine Kategorisierung nach Art der Rechnerplattform und eine Unterscheidung nach Virtualisierern und Emulatoren für die derzeit am weitesten verbreitete X86-Architektur. Für diese existiert inzwischen eine größere Zahl von Produkten und Projekten. Es kann sowohl kommerzielle als auch Open Source Software für den Einsatz vorgesehen werden. Beide Ansätze verfügen über inhärente Vorteile, weisen jedoch auch einige Nachteile auf, die einer Archivierung über sehr lange Zeiträume im Wege stehen können.

Der dritte Teil widmet sich dem Kern der Arbeit. Dazu beschäftigt sich Kapitel 6 mit "View-Paths", den notwendigen Schritten ausgehend vom Primärobjekt, der Nutzung ausgewählter Sekundärobjekte bis zur Sichtbarmachung oder Ausführung durch den Benutzer. Die Benutzerinteraktion sollte sich an jeweils aktuellen Host-Systemen, der sogenannten Referenzumgebung abspielen. Für verschiedene Nutzungsumgebungen lassen sich Optimierungen vorstellen, die den Zugriff auf Objekte vereinfachen könnten. Im Zuge der besprochenen Arbeitsabläufe treten besondere Herausforderungen auf: Wie kann ein Primärobjekt geeignet bis in die jeweilige Nutzungsumgebung transportiert und wie können Ergebnisse dieser wieder entnommen werden?

Ausgehend von den geschaffenen Grundlagen stellt das letzte Kapitel ein mögliches Softwarearchiv aus Sekundärobjekten vor: Wie lässt sich das Konzept der Emulation von Rechnerarchitekturen sinnvoll in der Langzeitarchivierung digitaler Objekte verankern und was sind die dazu notwendigen Softwarekomponenten. Das Archiv muss neben den Emulatoren der unterschiedlichen Hardware ebenfalls Betriebssysteme und Anwendungsprogramme verfügbar machen. Ziel muss es sein, dass auch nach längeren Fristen Ablaufumgebungen jederzeit wiederhergestellt werden können. An dieser Stelle wird der Frage nachgegangen, welche Informationen und Daten für die Erstellung einer kompletten Umgebung zur Darstellung der verschiedenen digitalen Objekte gegeben sein müssen.

Den Abschluss bilden Fazit und Ausblick. Die große Menge, teilweise auch noch gar nicht in der Wahrnehmung der Archivare aufgetauchter Objekte, wird verhindern, dass es einen einzigen komplett umfassenden Archivansatz geben kann. Jedoch kann durch eine Verbreiterung des Archivansatzes, wie in dieser Arbeit gezeigt, das Gesamtergebnis und die Nachhaltigkeit digitaler Objekte verbessert werden. Jedoch wird es trotz anspruchsvoller Nachbildung von Ablaufumgebungen nicht immer möglich sein, jeden Aspekt eines digitalen Objekts korrekt nachzubilden. So bleiben offene Fragen, die sich beispielsweise aus den aktuellen Entwicklungen einer weltweit vernetzten Gesellschaft ergeben. Ein Anhang und ein Glossar mit den Erläuterungen der wichtigsten und am häufigsten gebrauchten Begriffe schließen die Arbeit ab.

## 2 Herausforderung Langzeitarchivierung

Dieses Kapitel dient der Begründung der Fragestellung. Hierzu zählt der kulturelle Hintergrund, dessen zentrale Elemente auf der Weitergabe und Vermehrung des Wissens fußen. Hieraus leitet sich der Anspruch der Bewahrung von Informationen und Erkenntnissen ab. Zur Motivation werden anhand klassischer Fragestellungen der Langzeiterhaltung die Probleme digitaler elektronischer Archive erarbeitet.

Informationen für die Nachwelt werden unbewusst oder bewusst schon seit langem aufgehoben und weitergegeben. So sollte die Pyramide nicht nur eine komfortable Grabstätte sein, die ihrem Inhaber ein luxuriöses Leben nach dem irdischen Dasein erlaubt. Ebenso erzählt das Bauwerk eine deutliche Geschichte an nachfolgende Generationen und kündigt von der Bedeutung (und dem Größenwahn) seines Auftraggebers. Das trifft auf eine Reihe anderer Kunstwerke und Bauten ebenfalls zu.

Viele Spuren menschlichen Handelns haben sich so viele Jahrtausende erhalten und damit ihre Schöpfer weit überlebt. Dabei war in vielen Fällen eine derart lange Überlieferung gar nicht die ursprüngliche Intention. Nicht alle Artefakte vergangener Epochen sind derart perfekt erhalten, wie manche Höhlenzeichnungen unserer Vorfahren vor über 40.000 Jahren oder Alltagsgegenstände aus den griechischen Stadtstaaten. Es bedarf einer eigenen Wissenschaft, um die gefundenen Objekte einzuordnen und zu kategorisieren. Nicht in jedem Fall gelingt es der Archäologie alle Artefakte auf Anhieb zu entschlüsseln. Sie ist dabei auf vielfältige Kontextinformationen angewiesen. Trotzdem kann es vorkommen, dass Objekte, wie Tontafeln, Stelen oder Gebäudeinschriften zwar physisch fast perfekt erhalten sind, sich Inhalt und Bedeutung der enthaltenen Inschriften bisher nicht haben klären lassen. Ohne Anweisungen zur Interpretation sind selbst perfekt erhaltene Zeichenfolgen nahezu wertlos.

Viele Erfahrungen und Erkenntnisse aus dem Bereich der klassischen Archive lassen sich durchaus abstrakt auf Archive neuen Typs übertragen. Sie geben Anhaltspunkte, mit welchen Aufgaben- und Problemstellungen zu rechnen ist.

### 2.1 Information, Überlieferung und Archivierung

Bis zur Erfindung der Schriftsprache geschah Überlieferung von Geschichten und Wissen ausschließlich in mündlicher Form. Mit der wirtschaftlichen Entwicklung und gesellschaftlichen Blüte im Zweistromland erhöhten sich Wissensproduktion und -umsatz. Schriftsprachen

entwickelten sich, so dass die Überlieferung nicht mehr nur auf das gesprochene Wort angewiesen war. Jedoch war der Aufwand der Verschriftlichung hoch: Tontafeln oder Granitstelen<sup>1</sup> konnten nur verhältnismäßig wenige Zeichen aufnehmen und ließen sich nicht oder nur recht mühsam transportieren. Tierhäute waren sehr aufwändig und teuer in der Herstellung und Verarbeitung zum Datenträger.

Das wirtschaftliche und religiöse Leben der Antike stellte neue Anforderungen, Schulen und Orte des geistigen Austauschs entstanden. Die antiken Philosophen warfen wesentliche Fragen auf, an denen sich die heutige Geisteswissenschaft immer noch abarbeitet. Auch 2500 Jahre nach ihrem Tod weiss man von den Gedanken eines Sokrates, Platon oder Aristoteles. Zumindest Teile ihrer Werke sind durch schriftliche Überlieferung und Übersetzung bis in die heutige Zeit transferiert worden. Viele Stücke des Theaters und der Literatur lassen sich nur verstehen, wenn man die großen menschlichen Erzählungen, wie die der griechischen Antike oder der Bibel kennt. Das neu erworbene und strukturierte Wissen ließ sich nicht mehr nur durch mündliche Weitergabe transferieren. Mit der Erfindung des Papyrus<sup>2</sup> als neues Medium vereinfachten sich Transport und Lagerung erheblich. Leider wurde trotzdem von den Texten der griechischen Komödien- und Tragödienautoren nur ein Bruchteil überliefert. Große Bestände antiker Literatur sind für immer verloren.

Die Entdeckung der Welt aus europäischer Sicht durch Seefahrt und Überseehandel führte zur Verbindung vorher getrennter Kulturkreise, was zur immensen Verbreiterung der menschlichen Wissensbasis beitrug. Der im späten Mittelalter aufkommende Gutenbergsche Buchdruck erlaubte eine einfache Vervielfältigung. Der Wissensumsatz steigerte sich im 19. und 20. Jahrhundert, befördert durch die industrielle Revolution. Generell sanken die Kosten von Kommunikation und Ideenaustausch drastisch.

Am Beispiel dieser Arbeit lässt sich gut verdeutlichen, welche Beschleunigung der Informationsumschlag erfahren hat. Während man noch vor 15 Jahren auf eine beschränkte Anzahl von Bibliotheken und wissenschaftlichen Zeitschriften angewiesen war, hat sich die Recherche nach aktuellen Artikeln und Aufsätzen zum Thema deutlich vereinfacht. Erkenntnisse, die gerade erst gewonnen und veröffentlicht wurden, können berücksichtigt werden. Gleichfalls sinkt die Wahrscheinlichkeit redundanter Doppelforschung.<sup>3</sup>

Das Wort Archiv bezeichnet ursprünglich spezielle Gebäude später in erweitertem Wortsinne Institutionen oder Organisationseinheiten, die sich mit der Aufbewahrung, Erhaltung, Auswertung und Erschließung von Archivgut, den Archivalien, befassen. Je nach Aufgabenstellungen handelt es sich bei den Archivalien um eine Vielzahl verschiedener Objekte, wie Akten, Karten, Pläne, Siegel, Schriftstücke, Bild-, Audio- und Videomaterialien, die als archivwürdig befunden wurden.

Für die Erhaltung der klassischen Objekte von bleibendem historischen, künstlerischen, gesellschaftlichen oder wissenschaftlichen Wert bestehen etablierte Institutionen. Bibliotheken, Museen oder ähnlich gelagerte Einrichtungen, wie Denkmalämter, sind mit dieser Aufgabe betraut. Objekte des alltäglichen Lebens, Arbeitsgeräte und Materialien werden seit dem 18.

<sup>1</sup>Auf einer solchen waren die Gesetze des Hammurabi, dem 6. König der ersten Dynastie von Babylon, eingemeißelt.

<sup>2</sup>Ein kompakter Überblick zur Geschichte der Speichermedien und ihre wichtigen Vertreter der jeweiligen Zeit liefert [Spilker 2007].

<sup>3</sup>Damit ist nicht die sinnvolle Überprüfung der Erkenntnisse anderer Forschungsgruppen gemeint, sondern die exakt gleiche Bearbeitung einer Fragestellung, die keine neuen Erkenntnisse generiert.



Jahrhundert in Museen archiviert. Sie prägen das öffentliche Gedächtnis und sind ein Teil der Kulturgeschichte der Menschheit.

Vielfach wird geschätzt, dass sich der Umfang des menschlichen Wissens alle 10 bis 15 Jahre verdoppelt. Das klassische Medium Papier ist für den Informations- und Wissensaustausch, vermittelt über gedruckte Publikationen, oft zu teuer und zu unflexibel. Durch neue Entwicklungen werden Informationen immer besser such- und auffindbar: So treibt der Internet-Suchmaschinen-Konzern Google mit seiner Digitalisierungsinitiative<sup>4</sup> diesen Prozess voran. Ganze Bibliotheksbestände können nun mit Volltextsuche recherchiert werden, wo früher Karteikarten mit Verschlagwortung und Kurzzusammenfassungen von Aufsätzen, Artikeln und Büchern nur einen groben Überblick liefern konnten. Auf der anderen Seite startete Mitte des Jahres 2007 eine Initiative der UNESCO und der US-amerikanischen Library of Congress für eine frei zugängliche World Digital Library.<sup>5</sup> Diese digitale Bibliothek soll kostenlos über das Internet zugänglich sein. Die Inhalte - besondere Materialien aus Bibliotheken und anderen Gedächtnisorganisationen, wie seltene Bücher, Karten, Handschriften, Partituren, Filme, Fotografien, architektonische Pläne - sollen online zur Verfügung stehen. Sie werden dazu in den sechs offiziellen UNO-Sprachen Englisch, Französisch, Russisch, Spanisch, Arabisch, Chinesisch und darüber hinaus in Portugiesisch angeboten.

Fanden viele elektronische Verarbeitungsschritte bis Anfang der 1990er Jahre des letzten Jahrhunderts noch isoliert voneinander statt, so hat sich mit der Durchsetzung des Internets als weitgehend<sup>6</sup> öffentliches Datennetz dieser Zustand komplett gewandelt.

Wo früher noch große Teile der erzeugten und verarbeiteten Daten auf externe Datenträger, wie Papier ausgedruckt wurden, werden diese jetzt in überwiegendem Maße nur noch elektronisch ausgetauscht. Hier bieten Computer und ihre Vernetzung völlig neue Möglichkeiten.<sup>7</sup>

### 2.1.1 Digitale Revolution

Gerade im Bereich der Informationsverarbeitung, -speicherung und -übermittlung sind Computer nicht mehr wegzudenken. In diesem Zusammenhang nimmt der Anteil digitaler Informationen am kulturellen und wissenschaftlichen Erbe immer größeren Raum ein. Das bedeutet, dass sich historische Ereignisse, Erkenntnisse in Forschung und Wissenschaft sowie alle möglichen Arten kultureller Leistungen nur noch in elektronischer Form niederschlagen. Die großen Vorteile liegen im sekundenschnellen Suchen und Finden dieser Informationen, der schnellen Verbreitung und Übertragung und der Abrufbarkeit an geografisch sehr unterschiedlichen Orten. Jedoch nimmt man damit einen gravierenden Nachteil in Kauf: Die Erstellung und der

---

<sup>4</sup>Einen Eindruck bietet <http://www.google.de/books?hl=de>. Die Aktion ist nicht unumstritten und hat zu größeren Diskussionen unter Autoren, Verlagen und Bibliotheken geführt. Eine Alternative bietet die Open Content Alliance (OCA), vgl. <http://www.heise.de/newsticker/meldung/97729>.

<sup>5</sup>Siehe hierzu <http://www.heise.de/newsticker/meldung/97681>.

<sup>6</sup>Die Zahl der mit dem Internet verbundenen Rechner ist weit größer als die Zahl der Maschinen, auf die direkt zugegriffen werden kann.

<sup>7</sup>Siehe [Lehmann 1996], S. 210. Weitere Anmerkungen zu Information und Wissen liefert Kapitel 3 in [Endres und Fellner 2000]. Darüber hinaus bietet [Spilker 2007] einen guten Überblick zur Geschichte der Speichermedien. Diese wird, soweit für die im weiteren Verlauf betrachteten virtuellen Datenträger interessant, ebenfalls kurz im Anhang Abschnitt A.5 dargestellt.

Zugang zu digitaler Information ist nur mithilfe eines nicht unerheblichen technischen Apparates möglich und gerade dieser Apparat unterliegt einer rasanten Fortentwicklung. Was einerseits erlaubt, immer mehr Information immer umfassender zu speichern und zu verarbeiten, führt auf der anderen Seite zur schnellen Veralterung der eingesetzten Technik inklusive ihrer Langzeitspeichermedien.

Die digitale Revolution besitzt, wie jede technologische Veränderung vor ihr auch, mehrere Aspekte. Zum einen hat sie die geschilderten Auswirkungen auf die Erzeugung und den Austausch von Wissen. Man möchte verhindern, dass die durch ihre Mitwirkung erstellten Produkte im Laufe der Zeit nicht von Verlust bedroht werden. Zum anderen bedeutet die digitale Revolution eine Veränderung der Technik, vergleichbar mit der Einführung der Druckerpresse oder des Telefons. Wie jede Technologie zuvor hat auch die digitale Revolution unvorhergesehene Fragen aufgeworfen, mit der sich nachfolgende Generationen auseinandersetzen müssen.

## 2.2 Computer als "Kultur"-Maschine

Computer sind industriegeschichtlich gesehen ein junges Phänomen. Seit der Mitte des letzten Jahrhunderts wurden aus reinen Rechenmaschinen, die auf eine bestimmte Anzahl von Operationen festgelegt waren, Geräte, die immer weitgehender frei programmiert werden konnten. Diese Programme werden als Software bezeichnet und bilden eine zunehmend komfortablere Schnittstelle zwischen Anwender und Hardware, der physischen Maschine. Dadurch wurde die Bandbreite der Aufgaben immer vielfältiger, die durch diese Maschinen übernommen wurden. Die weitgehende Durchsetzung des Computers als Multifunktionswerkzeug hat mittlerweile dafür gesorgt, dass in sehr vielen Bereichen des wirtschaftlichen, kulturellen, wissenschaftlichen und Freizeitlebens Daten in vorwiegend elektronischer Form erzeugt werden.<sup>8</sup>

Seit den 1970er Jahren dringt die Informationstechnologie mehr und mehr sichtbar, getrieben durch die Fortschritte in Informatik und Mikroelektronik, in weite Bereiche der Volkswirtschaft und des öffentlichen Lebens ein. Mit Beginn der 1980er Jahre "demokratisiert" sich die Technologie: Sie beschränkt sich nicht mehr auf eine Anwendung in großen Unternehmen und Rechenzentren. Die Mainframes und teuren Workstations der Pionierzeit werden ergänzt durch "persönliche" Computer: Sogenannte Home-Computer werden zu Kaufhausmassenware und erlangen eine gewisse Verbreitung in den Haushalten der Industrieländer.

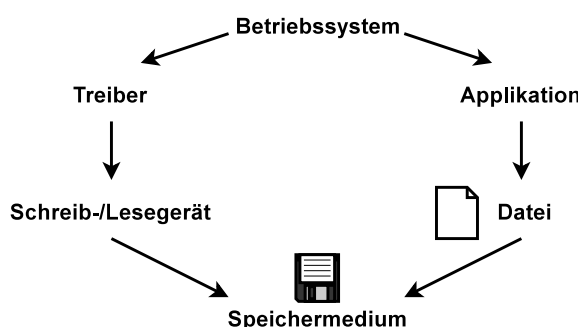
Mit dem Eindringen der Computertechnik in fast alle Bereiche des täglichen Lebens ändern sich auch die Techniken zur Speicherung, Verbreitung und Vervielfältigung von Informationen. Das Hauptbuch des Kaufmanns hat ausgedient und der Computer ist an seine Stelle getreten, Musik und Film ist nicht mehr an analoge Medien gebunden, sondern kann effizient digital gespeichert, kopiert und verbreitet werden. Wissenschaftler erheben ihre Daten immer seltener ohne elektronische Maschinen, erfasste Daten nehmen ohne Umweg über Papier und Bleistift den Weg zur Weiterverarbeitung, Auswertung und Archivierung. Rechner sind wie ihre Pendants der industriellen Revolution Arbeitsgeräte. Zweifellos hat der Computer die Datenverarbeitung revolutioniert und deren Effizienz erheblich gesteigert.

Durch jede Generation von neuen und immer schnelleren Computern verschwindet Software, die auf diesen Rechnern lief. Damit wird der Zugriff auf Gigabytes von Daten mit

---

<sup>8</sup>Einen Problemaufriss bieten beispielsweise [Waugh u. a. 2000] oder [Levy 1998].

eventuell historischer Tragweite unmöglich. Mit den Programmen haben jeweils einige Jahre lang Millionen von Menschen gearbeitet. Wer irgendwann die Briefwechsel von Personen der Gegenwart herausbringen möchte, könnte weitestgehend ohne Material dastehen, weil die Email-Korrespondenz zusammen mit dem alten Computer verschrottet wurde. Ebenso wird



**Abbildung 2.1:** Entstehungsumgebung digitaler Objekte nach [Rohde-Enslin 2004] mit den Zusammenhängen verschiedener Software- und Hardwarekomponenten.

es schwer, seinen Kindern zu vermitteln, wie die Anfänge der zunehmenden Begeisterung für Computerspiele auf den Home-Computern ausgesehen haben ([Baumgärtel 2002]). Die letzten verkauften Geräte sind noch nicht einmal 15 Jahre alt, aber kaum noch irgendwo zu sehen. Wenn man Glück hat, findet sich ein altes Gerät vereinzelt und in oft unbekanntem Zustand auf dem Flohmarkt. Darüber hinaus könnte dann das Wissen verlorengegangen sein, wie man Programme auf dieser Maschine ausführt, selbst wenn man die alten Datenträger irgendwo gefunden hat. Vor diesem Problem steht beispielsweise das Computerspiele-Museum in Berlin,<sup>9</sup> das über einen Bestand von über 17.000 Spieletiteln verfügt, die über knapp 30 Jahre Computer- und Spielekonsolengeschichte reichen.

Es besteht ein enger Zusammenhang im Erstellungskontext von Daten, der Entstehungsumgebung (Abb. 2.1), welche in den Kapiteln 6 und 7 von Bedeutung sein wird. Eine Darstellung der Problematik aus Sicht der Gedächtnisorganisationen liefert [Rohde-Enslin 2004].

### 2.2.1 Computer in Wissenschaft und Forschung

Die Entwicklung elektronischer Rechner beförderte einerseits Wissenschaft und Forschung. Immer komplexere Theorien, neue Forschungsergebnisse in Physik, Astronomie oder bessere Klimamodelle erforderten immer aufwändigere Berechnungen, die andererseits die Weiterentwicklung der Computer vorantrieben.

Die Einsatzbereiche von Computern in Forschung und Wissenschaft sind durchaus weit gestreut:

- Großforschungseinrichtungen - Diese, durchaus auch mit militärischem Hintergrund, zählen zu den ersten Institutionen, die sich neue revolutionäre Technologie leisten können.

<sup>9</sup>Vgl. [CSM 2008] - Dieses Museum wird durch den Förderverein für Jugend und Sozialarbeit e. V. getragen. Es betreibt keine dauerhafte Ausstellung sondern bestreitet regelmäßige Wanderausstellungen.

- Automatisierte Datenerfassung und Ablage - Bei sehr vielen Experimenten in der Physik oder Beobachtungen in der Astronomie fallen an vielen Stellen große Mengen von Daten gleichzeitig an. Diese können nicht mehr manuell abgelesen und abgelegt werden. Computer sorgen so dafür, dass große Datenmengen generiert und aus diesen neue Erkenntnisse gewonnen werden können.
- Komplexe Anlagen - Teleskope oder Teilchenbeschleuniger lassen sich nicht mehr durch manuelle Steuerungen betreiben. Sie erfordern leistungsfähige Computer, die den komplexen Aufgaben und deren zeitlichen Abfolgen gewachsen sind.
- Simulationen - Immer genauere Voraussagen zur Entwicklung des Klimas der Erde oder einzelner Regionen sowie Modellrechnungen in der Ökonomie benötigen gewaltige Rechenleistungen, die oft durch das Zusammenschalten sehr vieler Compute Nodes gewonnen wird.

Automatische Mess- und Beobachtungsdaten werden in den meisten Fällen nur noch in kompakter Form maschinenlesbar gespeichert, da sie computerbasiert weiterverarbeitet werden. Sie sind außerhalb ihres Kontexts nicht mehr sinnvoll interpretierbar.

In der Wissenschaft interessiert noch besonders ein weiterer Aspekt von Computern: Mit ihnen sind jahrzehntelang wissenschaftliche Daten erhoben und verarbeitet worden. Zum jeweiligen Zeitpunkt der Erstellung spielten Überlegungen zur Langzeitaufbewahrung dieser Daten keine oder eine nur sehr nachgeordnete Rolle: Mit der Publikation in einer Zeitschrift, mithin dem jahrhundertealten, klassischen Medium der Langzeitarchivierung waren viele Projekte abgeschlossen. Die Arbeitswerkzeuge, wie Rechner, Betriebssystem und eventuell selbstentwickelte Spezialsoftware wurden beiseite gelegt und vergessen. Publierte Ergebnisse, Berechnungen, Statistiken und Datensammlungen sind damit für die Nachwelt oftmals nicht mehr nachvollziehbar oder für neuere Forschungen verwendbar. Daten, deren Erhebung durchaus mehrere Millionen Euro verschlungen haben mag, sind nicht mehr für die aktuelle Forschung zugänglich, wichtige und wissenschaftlich interessante Vergleiche mit aktuellen Erhebungen somit unmöglich.

### 2.2.2 Kultur und Alltag

Anfang der 1980er Jahre des letzten Jahrhunderts setzte im Bereich der Informationstechnik eine Miniaturisierung und radikale Verbilligung elektronischer Geräte ein. Damit wurden Computer für den Heimanwender erschwinglich und eine erste Welle von Maschinen tauchte im privaten Bereich auf. Dadurch befördert nahm die Auseinandersetzung mit der Technik in breiten Bevölkerungsschichten zu. Dieses trug sicherlich mit dazu bei, die Verbreitung der Technik in den anderen Bereichen zu beschleunigen. Vielfach wurde die erste Generation der Home-Computer als reiner Zeitvertreib wahrgenommen. Es entstand eine komplett neue Unterhaltungsindustrie, die Spiele für den Heimgebrauch programmierte und vertrieb.

Eine neue Alltagskultur<sup>10</sup> entwickelte sich um Arcade-Spiele und Games auf den verschiedensten Minicomputern, neue Zeitschriften entstanden. Nach inzwischen 20 Jahren liegt diese

---

<sup>10</sup>Das bereits zuvor genannte Museum [CSM 2008] verfügt über eine der umfassendsten Sammlungen aus 30 Jahren Geschichte der Computerspiele.

Epoche schon wieder weit zurück, eine neue Ära mit neuen Technologien hat die Führung übernommen. Die Kurzlebigkeit des IT-Marktes führt dazu, dass oft die zur Darstellung von digitalen Informationen erforderlichen Programme auf aktueller Hardware nicht mehr ablauf-fähig sind, selbst wenn die Informationen vom Datenträger erfolgreich ausgelesen werden können.

Gesellschaftliche Entwicklungen mit einer bestimmten Relevanz schlagen sich auch kultu-rell nieder. Sie beflügelten zum einen die Fantasie von Buchautoren und Filmemachern. Zum anderen beförderten sie die Auseinandersetzung von Künstlern mit den neuen Möglichkeiten. Computer wurden technisches Hilfsmittel von Installationen und Performances und Gegen-stand künstlerischen Ausdrucks. Computerbasierte Kunstwerke lassen sich nur schwer oder nicht von ihrer Technik lösen.

Die Entwicklung immer leistungsfähigerer Computer zu immer geringeren Preisen revolu-tionierte ebenfalls die Produktionsverfahren in der Filmindustrie. Wo vorher virtuelle Welten mühsam in Modellen erschaffen oder sehr aufwändig in teuren Kulissen nachgebildet werden mussten, übernimmt der Computer die Virtualisierung. Landschaften,<sup>11</sup> Szenen und ganze Zeichentrickfilme werden im Computer nachgestellt und gerendert.<sup>12</sup> Hier werden inzwischen unvorstellbare Datenmengen generiert und zu bewegten Bildern animiert.

Seit einigen Jahren bilden sich neue Formen der Online-Interaktion heraus: Hierzu zäh-len Computerspiele, wie "World of Warcraft"<sup>13</sup> der Klasse der "Massively Multiplayer Online Games" oder Plattformen, wie "Second Life".<sup>14</sup>

### 2.2.3 Wirtschaft und Verwaltung

Große Rechenanlagen setzten sich schnell in Konzernen und Verwaltungen durch. Besonders leicht "computerisierbare" Probleme, wie Buchhaltungen wurden auf elektronische Datenver-arbeitung umgestellt. Dadurch konnten viele repetitive Standardtätigkeiten automatisiert und teilweise drastische Kostenreduktionen erreicht werden. Generell hat die informationstechni-sche Revolution zu einer Umgestaltung des Arbeitslebens geführt. Fast kein Arbeitsplatz in klassischen Verwaltungsbereichen kommt mehr ohne Computer aus.

Wirtschaftsunternehmen sind gesetzlich dazu verpflichtet, Unterlagen der verschiedenen Geschäftsprozesse für eine bestimmte Zeit aufzubewahren. Wenn ein weitgehender Teil inter-ner und externer Kommunikation, Prozessplanung und Entwicklung elektronisch stattfindet, ergeben sich für die Einhaltung der gesetzlichen Auflagen ganz neue Herausforderungen.

Die Aufbewahrungspflichten öffentlicher Körperschaften gehen oft noch über die Ver-pflichtungen der Wirtschaftsunternehmen hinaus. Hierbei geht es oft um die Nachprüfbarkeit von Bescheiden und Entscheidungen. Die Authentizität von Vorgängen muss auch nach vielen Jahren gerichtsfest belegbar sein.

Welche Rolle Daten für die Nachvollziehbarkeit von politischen Entscheidungsprozessen

<sup>11</sup>Atlantischer Ozean oder die Jungfernfahrt in "Titanic" - <http://www.vfxhq.com/1997/titanic.html>.

<sup>12</sup>Siehe hierzu die Darstellungen in <http://medien.informatik.uni-ulm.de/lehre/courses/ss02/Modelling-AndRendering/15-film+tv.pdf> und <http://medien.informatik.uni-ulm.de/lehre/courses/ss02/ModellingAndRendering/16-zeichentrick.pdf>.

<sup>13</sup>Produkt-Homepage - <http://www.worldofwarcraft.com>.

<sup>14</sup>Homepage des Projekts und der dahinterstehenden Firma - <http://secondlife.com>.

besitzen können, war vielleicht zu Teilen anhand der "Bundeslöschtage" zu den Leuna-Akten<sup>15</sup> beim Übergang von der Regierung Kohl auf die Regierung Schröder zu ermesen.

## 2.3 Innere und äußere Organisation digitaler Objekte

Digitale Objekte sind der Hauptgegenstand der Betrachtungen dieser Arbeit und bedürfen deshalb einer ausführlichen Erörterung. Deshalb erfolgt zunächst eine abstrakte Beschreibung, bevor es um ihre konkreten Ausformungen gehen wird.

Kenneth Thibodeau gibt in seinem Aufsatz<sup>16</sup> die folgende umfassende Definition: *"What are digital objects? We could try to answer this question by examining the types of digital objects that have been and are being created. Many types of digital information can and do exist in other forms. In fact, many types of digital information are rather straightforward transcriptions of traditional documents, such as books, reports, correspondence, and lists. Other types of digital information are variations of traditional forms. But many forms of digital information cannot be expressed in traditional hard-copy or analog media; for example, interactive Web pages, geographic information systems, and virtual reality models. One benefit of an extensive review of the variety of types of digital information is that it forces one to come to grips with this variety, which is growing both in terms of the number of types of digital objects and in terms of their complexity."*

Darüber hinaus liefert der Autor eine kompaktere Definition eines digitalen Objekts: *"... intensive definition of digital objects: a digital object is an information object, of any type of information or any format, that is expressed in digital form ... All digital objects are entities with multiple inheritance; that is, the properties of any digital object are inherited from three classes. Every digital object is a physical object, a logical object, and a conceptual object, and its properties at each of those levels can be significantly different. A physical object is simply an inscription of signs on some physical medium. A logical object is an object that is recognized and processed by software. The conceptual object is the object as it is recognized and understood by a person, or in some cases recognized and processed by a computer application capable of executing business transactions."*

Das bedeutet, dass für die Langzeitarchivierung digitaler Objekte eine Reihe von Teilspekten eine Rolle spielen, die gesondert betrachtet werden sollten. Hierzu setzt man sich mit verschiedenen Abstraktionsebenen der Objekte, ihres Aufbaus und ihrer Ablage auseinander (Abb. 2.2, S. 20).

**Physikalische Objekte** Physikalische Objekte sind: *"Signs inscribed on a medium"*.<sup>17</sup> Auf der physikalischen Ebene werden Bitströme aufbewahrt und kopiert unabhängig von der Bedeutung der dargestellten Bits. Diese Aufgabe übernimmt ein geeignetes Aufbewahrungssystem. Die Erhaltung dieses Bitstroms ist eine notwendige Bedingung zur Erhaltung digitaler Objekte und bildet damit die Grundlage aller weiteren Strategien.

<sup>15</sup>Siehe hierzu - <http://www.zeit.de/2003/26/Datenschwund>.

<sup>16</sup>Vgl. [Thibodeau 2002], S. 4, 5.

<sup>17</sup>Vgl. [Thibodeau 2002], S. 6. In [Rothenberg 1999], S. 2 ff. widmet sich der Autor Bitströmen wegen ihrer Bedeutung für die spätere Informationswiederbeschaffung einen eigenen Abschnitt *"Old bitstreams never die – they just become unreadable"*. Vgl. zudem die Ausführungen in [Rothenberg 2002], S. 7 ff.

Datenträger als "Behälter" der Bitströme spielen im Rahmen dieser Arbeit insofern eine Rolle, als dass sie einerseits das Ablagemedium für die archivierten Objekte stellen, andererseits die diversen digitalen Objekte von Archivinteresse<sup>18</sup> auf bestimmten Typen von Datenträgern vorliegen. Sie müssen üblicherweise von ihren ursprünglichen Trägern gelöst und geeignet in ein Softwarearchiv eingestellt werden.

**Logische Objekte** [Thibodeau 2002] definiert logische Objekte als "prozessierbare, das heißt verarbeitbare, Einheiten". Diese Einheiten werden von einer bestimmten Applikationssoftware erkannt. Die Erkennbarkeit durch die Applikation hängt dabei normalerweise vom *Datentyp* ab. Auf diese wird im Abschnitt 2.3.2 näher eingegangen.

**Konzeptionelle Objekte** Konzeptionelle Objekte<sup>19</sup> sind die eigentlichen, für den Benutzer bedeutungsvollen Einheiten: In der analogen Welt wären das Alltagsgegenstände, Karten, Bücher, Zeitschriften oder Musikstücke und Filme. In der digitalen Welt sind konzeptionelle Objekte *Dokumente*, aber auch Computerprogramme oder -spiele. Diese sind für die Nachwelt von eigentlichem Interesse.

Der Umgang mit konzeptionellen Objekten kann bedeuten, dass es verschiedene Repräsentationen desselben Objekts gibt. So könnte beispielsweise diese Arbeit als  $\text{\LaTeX}$ -Dokument abgelegt oder in PostScript oder PDF übersetzt worden sein. Ebenfalls wäre es möglich, jede Repräsentation in einem Hexadezimal-Editor<sup>20</sup> zu betrachten und damit sich den Bitstrom anzuschauen.

### 2.3.1 Digitale Bitströme und Dateien

Wenn ein Digitalarchäologe einer fernen Zukunft einen Datenträger findet und dessen physikalische Struktur und Aufbau verstanden hat, ist er in der Lage, einen Bitstrom zu extrahieren. Ohne Zusatzinformationen wird er jedoch wenig damit anfangen können.<sup>21</sup> Nach den Überlegungen zur inneren Struktur digitaler Objekte geht es deshalb um die Ablage der verschiedenen Objekte in einzelne Container. Dadurch wird eine äußere Struktur als Anordnungsprinzip definiert, die dem Benutzer in Form von Dateien und Verzeichnissen entgegentritt. Dateien und Verzeichnisse bilden damit eine Art äußere Struktur, die wiederum in einem Dateisystem auf einem Datenträger (Abb. 2.2) gespeichert ist.

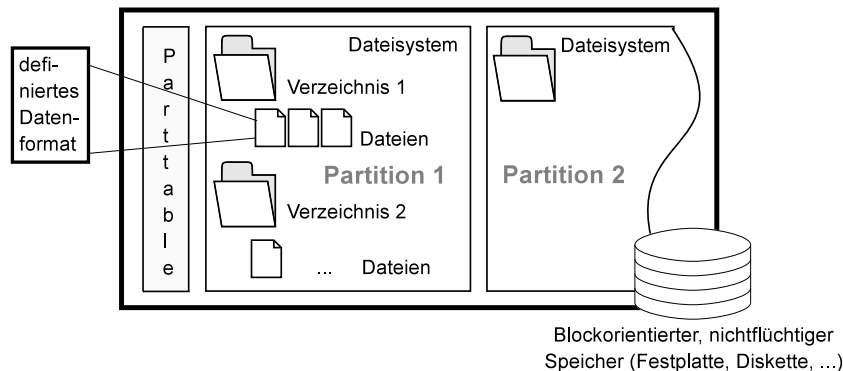
Auf Datenträgern kommt die physikalische Repräsentation der verschiedenen digitalen Objekte zu liegen. Sie erscheinen in der Grundform dem Betriebssystem gegenüber als "Flächen" oder "lineare Ströme" einer gewissen Kapazität. Im Fokus einer funktionalen Langzeitarchivierung steht der dauerhafte Zugriff auf digitale Objekte. Sie treten dem Archivbenutzer üblicherweise in Form von Dateien oder Bitströmen entgegen. Diese können vielfältige Formate besitzen. Für den Transport über Netzwerkverbindungen werden Dateien in der Quelle in Bitströme umgewandelt, serialisiert verschickt und am Ziel üblicherweise wieder in Dateien zurückgewandelt.

<sup>18</sup>Beispielsweise Installationsdatenträger, siehe auch Anhang A.5 zu verschiedenen Datenträgertypen.

<sup>19</sup>Vgl. gleichfalls [Thibodeau 2002].

<sup>20</sup>Spezieller Zeicheneditor, der den Rohinhalt von Dateien anzeigen kann.

<sup>21</sup>Vgl. hierzu [Thibodeau 2002], S. 12: "*So digital preservation is not a simple ... to reproduce the objects*".



**Abbildung 2.2:** Digitale Objekte besitzen eine innere Struktur, werden typischerweise als Dateien in Verzeichnissen eines Dateisystems auf einem Datenträger abgelegt, der wiederum über logische Untereinheiten verfügen kann.

[Lorie 2001] setzt digitale Objekte ins Verhältnis zur Form ihres Auftretens: *"A digital object generally corresponds to what we designate today as a file. Objects can vary greatly in complexity, forming a wide spectrum."*

**Verschiedene Arten** Es gibt eine ganze Reihe von Konzepten, Dateien in Hinblick auf ihre Langzeitarchivierbarkeit zu unterscheiden. Eine typische Einteilung erfolgt anhand des Datentyps: *"This recognition is typically based on data type. A set of rules for digitally representing information defines a data type. A data type can be primitive, such as ASCII or integer numbers, or it can be composite - that is, a data type composed of other data types that themselves might be composite."*<sup>22</sup> So unterscheidet der Autor einfache und zusammengesetzte Datentypen. Dabei ist man nicht auf eine vorgegebene Auswahl von Datentypen festgelegt – jede Applikation kann ihr eigenes proprietäres Format definieren.

Die Art der Zusammensetzung von Dateien entscheidet wesentlich über ihre spätere Interpretierbarkeit. Hierzu grenzt [Lorie 2001] folgende Fälle voneinander ab:

- *Case 0: It is the simplest case. The data object is readily understandable by a human reader. The only practical example is a piece of text encoded in a well known alphabet such as ASCII. This is clearly easy to archive - we only need to know the alphabet.*
- *Case 1: The data has to be decoded in some way by the reader or by a machine (assuming one knows the decoding rules). In the latter case, a program must be written in 2100 to decode the data, based on the stored description; the program must also be able to convey to the (human) reader information on what the data (or individual portions of the data) mean (the metadata). An image, a digital video clip, a table with ASCII fields, are all examples of such data objects. Essentially, the algorithm is simple and amenable to a description in plain English. Writing the program in 2100 should not be very difficult, whatever the programming language is at that time.*
- *Case 2: When the algorithm complexity reaches a certain level, it may become impractical to explain it in natural language. The alternative is to save a program that decodes*

<sup>22</sup>Vgl. [Thibodeau 2002], S. 6.



*the data; this may be the only way to be sure that the decoding is specified completely. The program is written in some language L; M 2100 must be able to understand that language and interpret it.*

- *Case 3: At the extreme end of the spectrum, we may be interested in archiving a computer program or system for its own sake. In this case, it is the ability to run that program that must be preserved by the archiving mechanism. Not only the bit stream that constitutes the program must be archived, but we must also make sure that the code can be executed at restore time. If you want to preserve the look and feel of Window 95 or MAC, or the user interface of a Computer Aided Design system, the only solution is to archive the whole body of code used during the execution, and enough information on how to run the code at restore time. Programs with complex interactions between the program and a user fall in this category.*

**Verzeichnisse** Verzeichnisse sind Konstrukte zur Strukturierung der Datenablage in Dateien. Wie beim physischen Vorbild<sup>23</sup> handelt es sich bei einem Verzeichnis um eine nach bestimmten Strukturen gegliederte Sammlung von Informationen. Als Verzeichnisstruktur bezeichnet man im allgemeinen Sprachgebrauch die logische Gestalt eines Dateisystems. Sicherlich am verbreitetsten und den meisten Computerbenutzern am geläufigsten ist eine baumartige Struktur. Die Struktur beginnt aus einer gemeinsamen Wurzel und verzweigt sich dann im Rahmen der vorgegebenen Eigenschaften nahezu beliebig. In grafischen Darstellungen wird die Wurzel typischerweise oben eingezeichnet und die Verzweigungen erfolgen nach unten hin.

Auf jeder Ebene eines modernen Dateisystems dürfen sowohl Dateien als auch Verzeichnisse liegen. Letztere bilden rekursiv die nächste Ebene des Baumes, für die erneut das eben Gesagte zutrifft. Typischerweise finden sich zwei Arten der Adressierung von Dateien: absolut und relativ. Name und Position jeder Datei kann durch eine entsprechende Pfadangabe spezifiziert werden, die an der Wurzel beginnt und alle zu durchlaufenden Verzeichnisse auflistet.

### 2.3.2 Dateiformate und Standards

Digitale Objekte haben als logische oder konzeptionelle Objekte mit Bitströmen oder Dateien zu tun, die in irgendeiner Form durch Software erzeugt wurden. Softwareprodukte sind eng mit dem Marktgeschehen verknüpft. Dieses hat einige Auswirkungen auf die Menge und Gestalt ihrer benutzten Formate:

- Softwareproduzenten haben ein Interesse daran, Kunden langfristig an ihre Produkte zu binden. Da dazu Qualität des Produktes und Funktionsumfang häufig als nicht ausreichend eingeschätzt werden, wird versucht, dieses über das Datenformat der erzeugten Objekte zu realisieren. Die Datenformate werden typischerweise nicht oder unvollständig veröffentlicht und bewusst so gestaltet, dass sie nicht leicht von Dritten nachempfunden werden können. Teilweise probieren Hersteller, sie durch Patente<sup>24</sup>

<sup>23</sup>Ein oft angeführtes Beispiel für Verzeichnisse sind das Branchenbuch oder interne Telefonbücher von Firmen oder Organisationen.

<sup>24</sup>Das kann dazu führen, dass ein Import solcher Objekte verhindert wird.

rechtlich zu schützen. Diese nicht allgemein veröffentlichten Datenformate werden oft als "proprietär", dem Erzeuger gehörend, bezeichnet.

- Führende Softwareprodukte definieren quasi durch ihre schiere Marktpresenz einen Quasistandard für Datenformate bestimmter Anwendungsbereiche, so beispielsweise Microsoft Excel für Tabellenkalkulationen.
- Open-Source-Programme erzeugen üblicherweise Dateien in offenen, jedem zugänglichen Formaten. Sie sind vielfach in der Lage, proprietäre Dateien von jeweils ähnlich gelagerten Programmen zu öffnen.<sup>25</sup>
- Anbieter von Konkurrenzprodukten in dominierten Märkten versuchen ihre Software oft mit guten Im- und Exportfiltern auszustatten, um eine Migration von anderen Anbietern zu eigenen Produkten zu erlauben. Dieses wird von Anwendern oft nur dann akzeptiert, wenn mit dem Alternativprodukt ähnliche Ergebnisse wie in den Quasistandardformaten erzielt werden können.
- Gibt ein Anbieter am Markt auf, verschwinden mit ihm dessen "proprietäre" Formate. Die ökonomische Konstellation beim Marktausscheiden sorgt in den seltensten Fällen dafür, dass ein Exportfilter zur Migration der alten Daten in das Format der überlegenen Produkte zur Verfügung steht.
- Je mehr Programme es gibt, die in unterschiedlichsten Formaten Daten ablegen können, desto größer wird die Matrix der Konversionswünsche.

Eine gute Illustration der Problematik bieten sogenannte Office-Produkte: Anwendungen wie Textverarbeitung, Präsentationsprogramm, Tabellenkalkulation, Termin- und Emailverwaltung.

Zur permanenten Verbreiterung des Angebotes zueinander inkompatibler Formate für ähnliche Problemstellungen lassen sich Gegenbewegungen beobachten:

- Mit der Durchsetzung von Computernetzen entsteht ein zunehmendes Bedürfnis nach einfachem Datenaustausch.
- Gerade kommerzielle Anwender fordern einheitliche Standards für die Datenablage, um zusätzliche Kosten zu vermeiden.
- Anwender sind immer weniger bereit, ihr Schicksal oder das Schicksal ihrer Daten mit dem des Softwareanbieters zu verknüpfen. Sie verlangen nach Standards für bestimmte Datenformate. Diese sollen zum einen auch nach größeren Zeitabständen noch interpretierbar sein und zum anderen einen leichten Austausch mit anderen Softwareprodukten erlauben.
- Die verwendeten Standards sollen möglichst offen und frei von Rechten sein, um unkalkulierbaren Risiken bei ihrer Benutzung aus dem Weg zu gehen.

Einen Überblick zu Standards und Datenformaten in Digitalen Bibliotheken findet sich in [Endres und Fellner 2000].

<sup>25</sup>So eignen sich beispielsweise die Importfilter von OpenOffice für ältere Microsoft Word Formate teilweise besser, als die der neueren Produkte des eigentlichen Herstellers.

### 2.3.3 Datenablage

Dateisysteme strukturieren Daten auf einem Datenträger in Dateien. Das Dateisystem selbst ist ein Ordnungs- und Zugriffssystem für diese Daten. Hierzu enthält jedes Betriebssystem<sup>26</sup> Zugriffsroutinen für mindestens ein Dateisystem.

Dateien besitzen in einem Dateisystem fast immer mindestens einen Dateinamen sowie Attribute, die nähere Auskünfte zur vorliegenden Datei geben. Die Dateinamen sind je nach logischem Aufbau des Dateisystems wiederum in speziellen Dateien, den Verzeichnissen, abgelegt. Durch diese Verzeichnisse kann ein Dateiname und damit eine Datei vom Betriebssystem gefunden werden. Ein Dateisystem bildet einen Namensraum. Alle Dateien oder dateiähnlichen Objekte sind auf diese Weise durch eine eindeutige Adresse, nämlich den Dateinamen inklusive Pfad oder Unified Resource Indicator (URI) innerhalb des Dateisystems aufrufbar. Die Zahl der Zeichen im Datei- oder Verzeichnisnamen entscheidet über Wiederfindbarkeit von Dateien und bestimmt über die erreichbare deskriptive Qualität. Der Name einer Datei und weitere Informationen, die den gespeicherten Daten zugeordnet sind, werden als Metadaten<sup>27</sup> bezeichnet.

Dateisysteme sind keine zwingende Voraussetzung zur Anordnung von Dateien und Verzeichnissen. So werden die Daten auf Magnetbändern üblicherweise gepackt als Bitstrom hintereinandergeschrieben. Für unterschiedliche Datenträger, wie Magnetband, Festplatte, optische Datenträger oder Flash-Speicher wurden spezielle Dateisysteme entwickelt, die deren Besonderheiten berücksichtigen. Sie definieren die für Verwaltung, Speicherung und Verarbeitung notwendigen Richtlinien zur Organisationsform der Datenträger, wie Größe der Cluster und Partitionen, Sicherheitseinstellungen, Zugriffsrechte, Attribute und die Organisation von Zugriffen.

Die hierarchische Ablage von Daten in Dateien und Verzeichnissen ist ein seit langer Zeit gültiges Paradigma, welches jedoch einige Probleme, wie die Ablage und Verknüpfung von Metainformationen zu einzelnen Dateien, nicht befriedigend löst. Deshalb tauchen regelmäßig Vorschläge auf, die versuchen, Prinzipien relationaler Datenbanken für Dateisysteme verfügbar zu machen. Das könnte bedeuten, dass konventionelle Verzeichnisstrukturen mit hierarchischer Dateiablage nur eine unter mehreren Varianten sein wird, ein "View" zu generieren. Das würde das gewohnte Paradigma, Datenbestände abzulegen, wiederzufinden, zu lesen oder zu modifizieren, verändern.

Dateisysteme sind in der Langzeitarchivierung an einigen Punkten von Interesse: Installationsdatenträger und Dateisysteme in den *Containerdateien* verschiedener Emulatoren oder Virtualisierer.<sup>28</sup>

Datenträger<sup>29</sup> kann man damit als Träger von Information, Speicher oder Speichermedium bezeichnen. Sie stellen physische Trägersubstanzen bereit, auf denen Daten abgelegt werden, was üblicherweise in den zuletzt beschriebenen Dateisystemen, Verzeichnissen und Dateien

<sup>26</sup>Ausgenommen sind sehr alte Betriebssysteme von Maschinen, die noch nicht mit blockorientierten Festspeichermedien bestückt waren. Hierzu zählen frühe Großrechner und einige Home-Computer.

<sup>27</sup>Sinnvollerweise verlässt man sich nicht auf die Eigenschaften eines Dateisystems zur Beschreibung der Objekte, sondern speichert in geeigneter Form Metadaten zu diesem Objekt separat ab und sorgt für eine geeignete Verknüpfung.

<sup>28</sup>Siehe hierzu die Abschnitte 6.8 und 7.10 in späteren Kapiteln.

<sup>29</sup>Enthalten die eingangs definierten physikalischen Objekte. Einen Überblick zu gängigen Datenträgern und ihrer Geschichte findet man im Anhang ab Abschnitt A.5.

erfolgt.

Verschiedene Datenträger mit unterschiedlichen physikalischen Eigenschaften sind integrale Teile des Hardwarebestandes. Sie sind ohne ihren Kontext aus Schreib-/Lesegeräten und Software, die für ihren Zugriff benötigt wird, nicht sinnvoll verwendbar. Einen guten Überblick liefern die Tabelle in [Rothenberg 1999], S. 3 und die Anmerkungen von [Thibodeau 2002], S. 7 zu physikalischen Archivierungsstrategien.

## 2.4 Haltbarkeit digitaler Objekte

Die informationstechnische Revolution hat ein Paradoxon beschert: Die Speicherung von immer mehr Daten, die neben den klassischen Dokumenten und Bildern nun auch Audio- und Videodatenströme, wissenschaftliche Rohdaten, interaktive Programme erlaubt, ist zu sowohl fallenden Kosten als auch auf immer geringerem Raum möglich. Wo jedoch in der Vor-Computer-Ära aufgrund der geringen Packungsdichte der Informationen eine Betrachtung auch ohne aufwändige Hilfsmittel, somit oft zu recht geringen Kosten, möglich war, steigt dieser Aufwand mit jedem neuen digitalen Objekt und jeder neuen physikalischen Speichertechnik. Wo vor einigen hundert Jahren die Grundzüge des menschlichen Wissens noch in eine mehrbändige Enzyklopädie passten und dort nachgeschlagen werden konnten, benötigt man nun gewaltige Speichersysteme. Diese erfordern einen hohen technologischen Aufwand zu ihrem Betrieb, der sich nicht einfach in einem bestimmten Zustand einfrieren und zu einem beliebig späteren Zeitpunkt wieder aktivieren lässt.

Nachdem die Gutenbergsche Druckerpresse im 15. Jahrhundert die Vervielfältigung von Texten und bestimmte Formen der bildlichen Darstellung revolutioniert hat, schuf erst die elektronische Datenverarbeitung die Erreichbarkeit der "unendlichen" digitalen Kopie. Bis dahin stellten Kopien, die möglichst originalgetreue Abbildung der Vorlage, ein nicht unerhebliches Problem dar.

Die Probleme analoger Kopien sind hinlänglich bekannt, weshalb nur zwei Beispiele zur Illustration genannt seien. Nicht nur Archivare kennen die Probleme der Übertragung von einer VHS-Video-Kassette auf eine andere oder wiederholte Kopien von Kopien eines Buches im Xerox-Verfahren. Im ersten Beispiel besteht neben der Qualitätsreduktion durch den Kopiervorgang ein weiteres Problem: Das Original nutzt sich während des Kopiervorganges, im Beispiel bleibend durch das wiederholte Abspielen im Videorecorder, selbst ab. Allein dadurch leidet die Vorlage unumkehrbar.

### 2.4.1 Illusion einer ewigen digitalen Kopie

Die Qualität der Kopie nimmt je nach angewandtem Verfahren und betriebenem Aufwand in unterschiedlichen Graden ab. Jedoch gelingt es nie eine exakte Kopie des Originals zu erzeugen. Hinzu kommt ein weiteres Problem: Die Charakteristik des Datenträgers erlaubt eine unterschiedlich lange Aufbewahrung des Originals. Bereits hier kommt es im Laufe der Zeit zu Qualitätseinbußen. Ein Archivar steht vor dem Problem, zu welchem Zeitpunkt er eine Kopie anfertigt: Zu einem frühen Zeitpunkt ist der Verlust bezogen auf den Erstellungszeitpunkt des Originals noch gering, jedoch beginnt auch die Kopie im Augenblick ihrer Erstellung zu altern. Je früher man den Auffrischungsvorgang einplant, desto häufiger ist er – bezogen auf

eine bestimmte Zeitspanne – durchzuführen. Umgekehrt enthielte die Kopie zu einem sehr späten Zeitpunkt des Lebenszyklus eines Objekts vielleicht nicht mehr alle Informationen.

Es gelingt fast immer, von einem Buch, welches kurz vor dem Zerfall aufgrund von Säurezersetzung oder Tierfrass steht, noch eine Kopie anzufertigen, die zumindest den Inhalt weitgehend rettet, wenn vielleicht auch die Form der Kopie nicht perfekt mit dem Original übereinstimmt. Das Ergebnis des beschriebenen Rettungsvorganges ist weiterhin nutzbar, wenn vielleicht auch zusätzliche Informationen, wie die handwerkliche Komponente der Bindung, verloren gegangen sind.

Auf diese Schwierigkeiten schien die digitale Kopie eine perfekte Antwort gefunden zu haben, wenn man den Aspekt der Digitalisierung ursprünglich analoger Objekte einmal außen vor lässt. Bei der Digitalisierung handelt man sich zwangsläufig Reduktionen ein, da beispielsweise die Auflösung einer Fotografie, die nur durch Molekülgröße und Abstand begrenzt ist, nicht auf die digitalen Kopien übertragen werden kann. Zum einen ist die Packungsdichte der fotoaktiven elektronischen Bauteile in der Kamera- oder Scanner-Optik noch deutlich niedriger. Zum anderen wären die Datenmengen so hoch, dass sie sich selbst mit den derzeit verfügbaren Computern und Speichersystemen nicht sinnvoll in großer Anzahl archivieren ließen.

Die Probleme der digitalen Kopien wurden in der ersten Euphorie der neuen Möglichkeiten vielfach nicht wahrgenommen. Während alle analogen Kopiertechniken anerkanntermaßen mit Verlusten und Verfälschungen kämpfen mussten, ist es hingegen sehr leicht, von einem digitalen Objekt eine Kopie herzustellen, die sich in keiner Weise vom Original unterscheidet.

Digitale Kopien haben darüber hinaus eine Reihe sehr attraktiver Eigenschaften. Die Kosten ihrer Erstellung sind oft sehr gering. Hinzu kommt ein völlig neuer Aspekt: Sie lassen sich bequem über Computernetze austauschen, womit räumliche Abstände der Betrachtungsorte keine besondere Rolle spielen. Das erlaubt es, ein Objekt viel leichter geografisch entfernten Betrachtern zur Verfügung zu stellen. Gleichzeitig wird eine deutlich höhere Sicherheit in der Archivierung erreicht, da mit der räumlichen Verteilung und einer größeren Anzahl von Kopien die Wahrscheinlichkeit eines Totalverlustes sinkt. In Abhängigkeit vom Typ des Objekts kann der Betrachter oft noch zusätzlich entscheiden, welche Art der Ansicht er wählt. So kann ein PDF-Dokument bequem am Bildschirm betrachtet oder alternativ auf Papier ausgedruckt werden. Eine gute Einführung in die Problematik findet sich in [Lehmann 1996], S. 3 ff.

## 2.4.2 Randbedingungen der Lesbarkeit

Analoge Objekte wie Bücher, Fotografien, Skulpturen oder Malerei erschließen sich dem Betrachter sofort, ohne dass er dafür spezielle technische Hilfsmittel gebrauchen muss. Für den Betrachter ist es komplett unerheblich, dass das Buch das Ergebnis eines sehr aufwändigen Maschineneinsatzes ist. Ein belichtetes und entwickeltes Papierfoto benötigt keine komplexe Technik oder aufwändige chemische Verfahren zur Betrachtung. Schwieriger wird es bereits bei "jüngeren" Techniken, wie analogen Audio- und Videoaufnahmen. Hier reicht es zur Darstellung seitens des Anwenders nicht mehr aus, lediglich den Datenträger zu betrachten. Für eine VHS-Video-Kassette<sup>30</sup> benötigt man meistens eine Kombination aus Abspiel- und Be-

<sup>30</sup>Der VHS-Standard für Videokassetten (Video Home System) wurde in den 1970er Jahren von JVC definiert und ist mediengeschichtlich damit noch nicht besonders alt.

trachtungsgeräten: In diesem Fall einen Videorecorder der VHS-Norm und ein Fernseh- oder anderes geeignetes Gerät.<sup>31</sup>

Wegen vieler Probleme der Langzeitqualität der Datenträger und der technologischen Fortentwicklung im Bereich Videoaufzeichnung und Wiedergabe wird die Verfügbarkeit der notwendigen VHS-Abspielgeräte in Zukunft abnehmen. Die Problematik ist an sich nicht neu: Archive standen schon häufiger vor dem Problem, dass die technologische Weiterentwicklung alte Systeme obsolet werden und damit aus dem Gebrauch verschwinden ließ.

An dieser Stelle bleibt häufig nichts anderes übrig, als unter Inkaufnahme gradueller Qualitätsdegradation eine Kopie in ein aktuelles Audio- oder Videosystem vorzunehmen. Das bedeutet immer, dass jeweils ein Abspielgerät der alten als auch ein Aufnahmegerät der neuen Generation verfügbar sein muss und deren Aus- und Eingänge in geeigneter Form verknüpfbar sein müssen. Diese Art des Umgangs mit abgelösten, ersetzten Techniken suggeriert eine ähnliche Lösung für digitale Archive.

Digitale Objekte stehen vor einem deutlich komplexeren Problem. Anders als bei vielen analogen Objekten, die komplett ohne ihren Erstellungskontext betrachtet werden können, ist dieses bei digitalen Objekten nicht mehr gegeben. Computer basieren auf dem Zusammenwirken von Software und Hardware. Die Maschinen selbst können als statische Objekte quasi beliebig lange aufbewahrt werden. Der reine physische Erhalt garantiert jedoch nicht die Funktionsfähigkeit. Ebenso wie die Abspielgeräte aus der analogen Welt teilen sie mit diesen die Probleme mechanischer oder elektrischer Ausfälle. Darüber hinaus nimmt die Komplexität der Gesamtsysteme permanent zu, so dass die Hoffnung auf relativ einfache Reparaturen oder Ersatzteilbeschaffung abnehmen muss.

Ist das Abspielsystem "Computer" nicht mehr einsetzbar, ergeben sich daraus fatale Folgen für dynamische digitale Objekte, wie Betriebssysteme und Anwendungsprogramme. Ein simpler Programmausdruck auf Papier oder Anzeige auf einem Bildschirm ist nicht ausreichend, weil es keine Interaktivität ermöglicht und damit der Dynamik des Mediums nicht gerecht wird. Mit der Gefährdung der Software, mit denen digitale Objekte erstellt wurden, ist auch das Objekt selbst gefährdet.

Anders als bei überlieferten Theaterstücken aus der Epoche des griechischen Altertums oder Noten aus der Bach-Zeit ist ein Informationsgewinn aus der Interpretation statisch abgelegter Programme sehr schwer zu realisieren. Er fällt damit eher spärlich aus und ist von geringem Wert. Die fehlende Ausführung verhindert die Möglichkeit eines Nachempfindens, womit Sinn und Funktion eines Programmes im Dunklen bleiben.

### 2.4.3 Kontexte

Viele Daten sind nur im gemeinsamen Kontext mit ihrem erzeugenden Programm heraus sinnvoll interpretierbar, was man schon beim fehlschlagenden Versuch feststellt, ein Word-Dokument sinnvoll in einem klassischen Texteditor darzustellen. Dies stellt ein zentrales Problem dar, welches häufig unterschätzt wird. Im Gegensatz dazu lassen sich statische HTML-Dokumente, die im klassischen Text-ASCII-Format vorliegen, selbst ohne Browser nach vielen

<sup>31</sup>Es muss sichergestellt sein, dass das Abspielgerät ein Ausgangssignal produziert, welches dann vom System zur Betrachtung wiederum geeignet interpretiert werden kann. Probleme können bereits die unterschiedlichen Fernsehnormen PAL und NTSC bereiten.

Jahren mithilfe der Dokumentation der Beschreibungssprache rekonstruieren. Ebenso zeigt sich die Langlebigkeit des T<sub>E</sub>X-Formates, eines Textsatzsystems, welches Mitte der 1980er Jahre entstanden ist und auch heute noch die Generierung von vor 15 Jahren geschriebenen Dokumenten für die Druckausgabe erlaubt. Dieses schlägt jedoch bei unvollständig oder nicht dokumentierten Binärformaten fehl. Enthaltene Informationen gehen zum großen Teil verloren oder lassen sich nur mit hohem Aufwand in ihrem Gesamtsinn rekonstruieren.

Konvertierungen von Betriebssystemen und Programmen auf aktuelle Hardwarearchitekturen erfolgt nur dann, wenn es sich ökonomisch lohnt oder besondere Interessen dahinter stehen. Häufig ist der Aufwand einer Aktualisierung oder Konvertierung so hoch, dass er nicht geleistet wird. Vielfach stellt sich der Wert bestimmter Daten erst zu einem viel späteren nicht vorhersehbaren Zeitpunkt heraus.

#### 2.4.4 Abreißen der kulturellen Überlieferung

Eine pointierte Zusammenfassung der Probleme digitaler Archive, die auch schon an anderer Stelle, wie [Borghoff u. a. 2003] S. 3 zitiert wurde, liefert Rothenberg: "Digital documents last forever - or five years, whichever comes first". Ein etwas konstruiertes, aber gut nachvollziehbares Szenario beschreibt derselbe Autor in seiner Einleitung zu [Rothenberg 1999].

Sehr lange Zeit war der bestimmende Datenträger auch zur langfristigen Aufbewahrung von Information das beschriebene oder bedruckte Papier. Auch wenn die seit zehn Jahren gemachten Versprechungen vom papierlosen Büro noch nicht erfüllt wurden, so ist die Tendenz klar. Immer mehr Daten werden ausschließlich in elektronischer Form vorgehalten. In diesem Zuge sind jedoch zwei wesentliche Vorteile des klassischen Mediums Papier verloren gegangen – der Zugriff auf die Inhalte ist im Regelfall ohne Hilfsmittel möglich und so vom technischen Wandel unbeeinflusst. Zusätzlich ist das Medium Papier verglichen mit den im Computerbereich üblichen Speichermedien extrem langlebig.

Eine breite Palette an sogenannten Multimedia-Daten erweitert die Möglichkeiten der Speicherung von Bild und Ton oder löst analoge Verfahren ab, die zum Teil mehr als 200 Jahre alt sind. Gerade durch bessere Kompressionsverfahren können teilweise deutlich mehr Daten als auf analogen Medien abgelegt werden. Anders als bei analogen Medien erleiden Kopien von digitalen Objekten keinen Qualitätsverlust.<sup>32</sup>

Hierbei steht man jedoch vor einem Dilemma: Es existieren zwar langjährige Erfahrungen mit traditionellen Datenträgern, wie dem Papier. Jedoch sind die Erkenntnisse zu aktuellen Datenträgern systembedingt beschränkt. Physische "Inschriften" auf Datenträgern verlieren im Laufe der Zeit durch externe Einflüsse und medienbedingt an Unterscheidbarkeit.

- Magnetbänder gibt es schon seit den 1950er Jahren des letzten Jahrhunderts. Allgemein ging man von einer längeren Haltbarkeit aus. Vielfach sorgte jedoch ungünstige Materialwahl oder Materialermüdung dafür, dass Bänder vorzeitig unbrauchbar wurden. So beeinflussen sich beispielsweise die unterschiedlich polarisierten Magnetisierungen gegenseitig und heben sich im Laufe der Zeit langsam auf. Dieses ist besonders für Bänder, bei denen der Datenträger in engen Schichten aufgewickelt ist, problematisch.

---

<sup>32</sup>Er ist dann jedoch total, wenn die technische Lesbarkeit des Objekts nicht mehr gegeben ist.

- Unlesbarkeit von Disketten und CDs treten auf: Verformungen optischer Datenträger können dafür sorgen, dass sich die Spuren auf dem Medium leicht verschieben und damit bei sehr hoher Dichte nicht mehr getroffen werden. Ein Ausfall der Abspielgeräte und die Veralterung ihrer Schnittstellen treten mit zunehmendem Alter in immer höherer Wahrscheinlichkeit auf.

Der Verfall der Datenträger ist nur eines der möglichen Probleme. Damit sie wieder eingelesen werden können, bedarf es der passenden Geräte mit passenden Schnittstellen zu heutigen Systemen. Ein recht prominentes Beispiel der jüngeren Zeit sind die Stasi-Magnetbänder, auf denen Aktenreferenzen gespeichert waren, die erst eine sinnvolle Zuordnung von Akten überwachter Personen und beteiligter Mitarbeiter erlaubten.<sup>33</sup> Erst nachdem die Abspielgeräte und das notwendige Wissen zur Bedienung wieder auftauchten, konnten die gut erhaltenen Datenträger eingelesen werden.

Selbst wenn Daten auf aktuelle Medien kopiert werden, ist damit nicht die Garantie gegeben, dass die Daten sinnvoll interpretiert werden können. Das Wissen um das Datenformat und Software zur Interpretation sind weitere Erfolgsmomente. Bisher gibt es noch wenig dokumentierte Beispiele des unwiederbringlichen Verlusts digitaler Daten. Daraus folgt im Umkehrschluss jedoch nicht, dass dieses Problem nicht auftritt.

## 2.5 Begrenzter Ansatz - Hardwaremuseum

Die überwiegenden Ideen zur Langzeitarchivierung digitaler Objekte orientieren sich an bekannten Objekten statischer Natur. Bücher, Zeitschriften, Malerei sind Objekte, welche nicht erst durch die Interaktion mit dem Betrachter ihre Funktion entfalten. Der Fokus auf statische Objekte ist durch den Hintergrund der in der Langzeitarchivierung bestimmenden Organisationen, wie Bibliotheken und Museen bedingt. Diese Vorstellungen werden erst langsam durch die Aufnahme neuartiger Objekttypen, wie Multimedia-Enzyklopädien und E-Learning-Materialien erweitert. Diese Objekte stellen nicht so sehr an die Langzeitspeicherung, sondern an die Bewahrung des Langzeitzugriffs neue Anforderungen.

Analog muss geklärt werden, was von der Computergeschichte aufgehoben werden sollte. Hierbei sind eine eher *wissenschaftszentrierte* und eine eher *museale oder alltagskulturelle* Betrachtungsweise zu berücksichtigen. Die Wissenschaft muss Wert darauf legen, dass komplette Funktionsmodelle erhalten werden, die notwendige Schnittstellen zum Einspeisen und Bearbeiten alter Datenbestände enthält. Der Augenmerk liegt auf Architekturen, die vielleicht einen geringen Ausstellungswert aber einen hohen funktionalen Wert besitzen.

Thibodeau gibt in seinem Aufsatz<sup>34</sup> den Hinweis: *"In addition to identifying and retrieving the digital components, it is necessary to process them correctly. To access any digital document, stored bit sequences must be interpreted as logical objects and presented as conceptual objects. So digital preservation is not a simple process of preserving physical objects but one of preserving the ability to reproduce the objects."* und beschreibt das Ziel einer Langzeitarchivierung:<sup>35</sup>

<sup>33</sup>Vgl. hierzu die Anmerkungen zur sogenannten "Rosenholz-Datei" in [DBT 2007].

<sup>34</sup>Vgl. [Thibodeau 2002], S. 12.

<sup>35</sup>Ebd. S. 13.



*"Thus, it is impossible to preserve a digital document as a physical object. One can only preserve the ability to reproduce the document. Whatever exists in digital storage is not in the form that makes sense to a person or to a business application. The preservation of an information object in digital form is complete only when the object is successfully output.*

*The real object is not so much retrieved as it is reproduced by processing the physical and logical components using software that recognizes and properly handles the files and data types. So, the black box for digital preservation is not just a storage container: it includes a process for ingesting objects into storage and a process for retrieving them from storage and delivering them to customers. ...*

*For archives, libraries, data centers, or any other organizations that need to preserve information objects over time, the ultimate outcome of the preservation process should be authentic preserved objects; that is, the outputs of a preservation process ought to be identical, in all essential respects, to what went into that process. The emphasis has to be on the identity, but the qualifier of »all essential respects« is important. ... One can only preserve the ability to reproduce the document. Whatever exists in digital storage is not in the form that makes sense to a person or to a business application. The preservation of an information object in digital form is complete only when the object is successfully output."*

Zur Archivierung der Alltagskultur lassen sich zuallererst die Geräte selbst so wie eine Dampfmaschine in ein Museum stellen. Wenn sie dort jedoch nur statische Ausstellungshüllen sind, bleibt auch hier ihr Eindruck und Nutzen unvollständig. Ein wesentlicher Bestandteil des Gerätes ist die Software, die auf der jeweiligen Maschine läuft und hinter die das Gerät oft selbst zurücktritt. Bei den Mainframes ihrer Zeit stand die Nutzung der Software im Vordergrund, da die wenigsten Anwender vor der Maschine selbst gesessen haben. Vom bekannten Home-Computer Commodore 64 ist der visuelle Eindruck eines laufenden Spiels mit Soundkulisse genauso wichtig, wie das Design des Gerätes selbst.

Die klassische museale Archivierung elektronischer Geräte<sup>36</sup> findet bereits statt, hat jedoch ihre Grenzen. Die Zahl der zu archivierenden Geräte auf gegebenem Raum ist eingeschränkt. Es müssen Ersatzteile und Wissen ständig zum Betrieb vorgehalten werden. Vielfach können deshalb Geräte nur an ausgesuchten Orten vorgehalten werden, die physische Präsenz ist manchmal nur in einem einzigen Museum möglich, statt an vielen Stellen. Hinzu kommen Defekte, mechanische Abnutzung, hohe Wartungs- und Betriebskosten. Diese ergeben sich aus Art der Maschine – Mainframes konnten sich damals nur sehr große Unternehmen leisten, für die sich die hohen Betriebskosten aus Wartung, Personal, Energiezu- und Wärmeabfuhr sowie Raumbedarf rentierten. Solche Kalkulationen können für ein Museum auf längere Sicht kaum aufgehen.

Der Artikel [Dooijes 2000] zeigt in einem Erfahrungsbericht die Probleme und Schwierigkeiten auf, die bei der Erhaltung einer recht großen Computersammlung<sup>37</sup> auftreten. Ein weiteres nicht zu unterschätzendes Problem liegt in der Beschränkung der physikalischen Haltbarkeit elektronischer Komponenten. Diese stellt im klassischen Betrieb normalerweise keine Schwierigkeiten dar, da in den meisten Fällen nach wenigen Jahren ein Aus-

<sup>36</sup>Beispiele sind das Computer History Museum in Mountain View Kalifornien [CHM 2008], das Computermuseum der Universität Amsterdam [CMUvA 2008] oder die Bemühungen des Computer History Simulation Project [CHSP 2007].

<sup>37</sup>Maschinen und einzelne Hardwarekomponenten, siehe auch [CMUvA 2008] - Computermuseum der Universität Amsterdam, oder die Anmerkungen in [Granger 2000] "Reliance on computer museums".

tauch erfolgt. Ein solcher ist jedoch für eine langfristige Archivierung ein gravierendes Problem. Die Herausforderungen und Gründe zur Bewahrung älterer Plattformen findet man in [Burnet und Supnik 1996] beschrieben. In diesem Aufsatz wird gleichzeitig eine Brücke zur Emulation als Ergänzungsstrategie zur Bewahrung alter Hardware geschlagen. Der Aufsatz [Rothenberg 2002], S. 12 setzt sich kritisch mit dem Konzept einer Computerhardwaresammlung auseinander und leitet daraus im weiteren Verlauf die Notwendigkeit einer Emulationsstrategie ab.

Das in Abschnitt 2.2 bereits genannte Computerspiele-Museum verfügt über einen großen Bestand historischer Geräte. Jedoch ist fragwürdig, wie lange einzelne Komponenten noch funktionsfähig bleiben. Zudem wird es kaum gelingen, alte Hardware außerhalb von Museen in großer Vielfalt aufzuheben. Es ist jedoch realistisch, zumindest die Software für zukünftige Generationen aufzubewahren. Der Weg hierzu führt über die Emulation alter Hardwarearchitekturen.

## 2.6 Forschungsansätze, Aktivitäten und Projekte

Die Langzeitarchivierung digitaler Objekte ist noch ein sehr junges Problemfeld, gemessen an den sonstigen Aufgaben der Hauptakteure in diesem Feld. Derzeit prägen Gedächtnisorganisationen, wie Bibliotheken, Archive und Museen die Diskussion. Obwohl die Themenstellung sehr stark mit der Informatik verknüpft ist, spielt sie bisher eher als Auftragnehmerin im Bereich der Forschung eine Rolle. Die Informatik arbeitet zwangsläufig an der Spitze der Weiterentwicklung und verliert dabei durchaus ihre älteren Erkenntnisse aus dem Blick. Bisher findet kaum eine Technikfolgenabschätzung im Sinne einer langfristigen Nutzbarkeit von Ergebnissen statt.

Archivare und Bibliothekare haben vielfach die Problematik früher als die Informatik selbst erkannt, da sie fachbedingt in ihrer Arbeit permanent eine Brücke zwischen durchaus fernerer Vergangenheit und dem Heute schlagen müssen. Dabei sollten sie gleichzeitig einen Blick in die Zukunft richten, da ihre Tätigkeit ebenso folgenden Generationen dient.

Seit der Jahrtausendwende hat die Zahl der Projekte und Initiativen zur Erhaltung des digitalen Erbes, des Langzeitzugriffs auf digitale Objekte sehr verschiedenen Typs und ihre Beschreibung deutlich zugenommen. So gibt es inzwischen eine Vielzahl deutscher und europäischer, beziehungsweise internationaler Initiativen und Aktivitäten. Einen kompakten Überblick bis zu Beginn des neuen Jahrtausends liefert [Heinze 2005] Abschnitt 3 "Nationale und internationale Initiativen im Kontext der digitalen Langzeitarchivierung" oder deutlich ausführlicher online auf [nestor 2008].

### 2.6.1 Nationale Akteure

Das Problem der digitalen Langzeitarchivierung wird in Deutschland derzeit von einer Reihe von Institutionen angegangen. Zu den Akteuren zählt dabei das Bundesministerium für Bildung und Forschung (BMBF)<sup>38</sup> als Förderer, die Deutsche Nationalbibliothek (DNB) und eine Reihe verschiedener Archive und Landesbibliotheken.

<sup>38</sup>So wurde beispielsweise im Jahr 2002 das Positionspapier "Informationen vernetzen - Wissen aktivieren" herausgegeben [BMBF 2002].

Das Ziel des Projekts "nestor" (Network of Expertise in longterm STOrage and availability of digital Resources)<sup>39</sup> besteht im Aufbau eines Kompetenznetzwerks zur Langzeitarchivierung und Langzeitverfügbarkeit digitaler Quellen für Deutschland als dauerhafte Organisationsform. Zudem soll die geschaffene Plattform einer Abstimmung über die Übernahme von Daueraufgaben im Aufgabenfeld Langzeitarchivierung dienen. Das Projekt startete 2003 und wurde 2006 um weitere drei Jahre verlängert.

- Nestor soll verteilte Kompetenzen sichtbar machen und bündeln, Informationen sammeln und Materialien zu technischen, organisatorischen und rechtlichen Aspekten der digitalen Langzeitarchivierung erarbeiten und bereitstellen.
- Es soll Kooperation ermöglichen und dabei Bibliotheken, Archive, Museen und Rechenzentren bei der Lösung des Problems der Langzeitarchivierung einbeziehen und unterstützen: Zur Langzeitarchivierung digitaler Ressourcen sollen und müssen angesichts der komplexen Fragestellungen kooperative Lösungen gefunden werden.
- Nestor soll Best-Practice-Modelle präsentieren und in diesem Zuge organisatorische und technische Verfahrensmodelle zur Langzeitarchivierung digitaler Ressourcen zur Diskussion stellen und eine breite Nachnutzung geeigneter Modelle anregen.
- Standards sollen auf dem speziellen Gebiet der Langzeitarchivierung befördert und Deutschland in internationalen Standardisierungsforen und -gremien vertreten werden.
- Nestor soll das Bewusstsein um Problematik und Dringlichkeit des Themas in der Fach- und der allgemeinen Öffentlichkeit verstärken und als Thema in die Diskussion um ein "Information Life Cycle Management" einbringen.
- Eine dauerhafte Organisationsform soll vorbereitet werden, die nach Ende des Projekts (August 2009) die Belange der Langzeitarchivierung digitaler Ressourcen in Deutschland wirksam koordinieren und vertreten soll.

Nestor betreibt eine Expertendatenbank und bearbeitet das Thema Langzeitarchivierung in einer Reihe von Arbeitsgruppen.

Kopal<sup>40</sup> ist etwas anders gelagert als Nestor: Ziel der Kooperation der Deutschen Nationalbibliothek, der SUB Göttingen, der Gesellschaft für wissenschaftliche Datenverarbeitung Göttingen (GwdG) mit IBM besteht im Aufbau und Betrieb einer technischen und organisatorischen Lösung, um die Langzeitverfügbarkeit elektronischer Publikationen zu sichern. Diese basiert auf DIAS,<sup>41</sup> einer Kooperation von IBM mit der Königlichen Bibliothek der Nieder-

<sup>39</sup>Siehe nähere Ausführungen in [Mittler 2005], S. 90 f. Eine Selbstdarstellung bietet zudem die Projekt-Homepage [nestor 2008]. Nestor gibt eine eigene Publikationsreihe die "nestor-materialen" heraus, wie beispielsweise [Borghoff u. a. 2005], [Coy 2006] oder [nestor 2006].

<sup>40</sup>Abkürzung für das Ziel des Projektes: Kooperativer Aufbau eines Langzeitarchivs digitaler Informationen; vgl. hierzu [kopal 2008] oder die Überblicksdarstellung in [Mittler 2005], S. 92 ff.

<sup>41</sup>DIAS ist eines der ersten großen technischen Projekte: Es lief bis 2002, führte verschiedene Studien zur Langzeitarchivierung durch und beschäftigte sich u.a. mit dem Problem der Authentizität digitaler Umgebungen [van Diessen und van der Werf-Davelaar 2002], Konzepten eines UVC [van Diessen und van der Werf-Davelaar 2002] und der Migration von digitalen Medienarchiven [van Diessen und van Rijnsoever 2002]. Siehe auch [KB 2007] zum Einsatz von DIAS im e-Depot.

lande. Zudem spielt die transparente Integration in vorhandene Bibliothekssysteme und die Nachnutzbarkeit durch Gedächtnisorganisationen eine wesentliche Rolle.

Bei der Implementierung des Systems werden internationale Standards in den Bereichen Langzeitarchivierung und Metadaten verwendet. Auf diese Weise werden sowohl die Zukunftsfähigkeit als auch die Möglichkeit zu einer Weiterentwicklung des Systems gewährleistet.

Innerhalb des Projekts werden digitale Materialien aller Art der Partner Deutsche Nationalbibliothek und Niedersächsische Staats- und Universitätsbibliothek Göttingen im Massenverfahren in das Langzeitarchiv eingestellt. Sie sollen von digitalen Dokumenten in Form von PDF, TIFF oder  $\LaTeX$  bis hin zu komplexen Objekten wie digitalen Videos reichen.

Zudem gab es ein von der Deutschen Forschungsgemeinschaft (DFG) gefördertes Projekt "Langzeitarchivierung digitaler Publikationen"<sup>42</sup> an der Bayerischen Staatsbibliothek in Kooperation mit der Universität der Bundeswehr München von 1999 bis 2001. Eine gute Zusammenfassung der Aktivitäten von Nestor, Kopal und des DFN liefert [Vögler 2007] in "Die Deutsche Forschungsgemeinschaft und deutsche Initiativen zur Langzeitarchivierung für die Wissenschaft".

Als Informationsportal zur Thematik Digitale Bibliothek betreibt, im Auftrag des BMBF mit Förderung seitens der DFG, das Deutsche Zentrum für Luft- und Raumfahrt (DLR) das "digital library forum".<sup>43</sup> Dieses Forum dient dem Informationsaustausch zwischen Bibliotheken, Medienexperten und politischen Entscheidern und informiert über aktuelle Förderprogramme.

Im Zuge eines von der DFG von 2005 bis 2007 geförderten Projekts zum "Aufbau einer organisatorischen und technischen Infrastruktur für die Langzeitarchivierung von Netzpublikationen" führt die Bayerische Staatsbibliothek gemeinsam mit dem Leibniz-Rechenzentrum Forschungen zur Ermittlung sinnvoller Metadatenschemata<sup>44</sup> durch.

## 2.6.2 Initiativen und Kooperationen in Europa

Im europäischen Kontext bestehen inzwischen eine ganze Reihe verschiedener Projekte und Initiativen. Bereits im Jahr 1994 wurde die European Commission on Preservation and Access (ECPA) ins Leben gerufen. Dieses Gremium aus Vertretern europäischer Bibliotheken und Archive, wie des schwedischen Nationalarchivs, der portugiesischen Nationalbibliothek oder der Niedersächsischen Staats- und Universitätsbibliothek hat die Beschäftigung und Vernetzung im Bereich des Bibliothekswesens zur Aufgabe. So soll beispielsweise der Erhalt und Zugang des kulturellen Erbes Europas gefördert werden.<sup>45</sup>

Digital Culture (DigiCULT - Technology Challenges for a Digital Culture) wurde im Rahmen der EU/IST (Information Society Technologies) als Projekt zur regelmäßigen Einschätzung aktueller Technologie für die Erhaltung des kulturellen und wissenschaftlichen Erbes über zweieinhalb Jahre von 2002 bis 2004 gefördert. Ziel dieses Projekts war es, durch Beoach-

<sup>42</sup>Vgl. [http://www.dl-forum.de/deutsch/projekte/projekte\\_316\\_DEU\\_HTML.htm](http://www.dl-forum.de/deutsch/projekte/projekte_316_DEU_HTML.htm) [DLR 2004].

<sup>43</sup>Siehe [DLR 2004]. Die Seiten werden seit Mitte 2007 aufgrund des Abschlusses dieses Projektes nicht mehr aktualisiert.

<sup>44</sup>Vgl. <http://www.lrz-muenchen.de/projekte/langzeitarchivierung> bzw. <http://www.babs-muenchen.de>: Aufbau von digitalen Entitäten durch Kombination verschiedener Schemata, wie MARC, Dublin Core und Formatverifikation durch PREMIS oder JHOVE.

<sup>45</sup>Vgl. hierzu die Homepage - <http://www.knaw.nl/ecpa>.

tung und Bewertung bestehender und sich neu entwickelnder Technologien die Bemühung der europäischen Gedächtnisorganisationen um Langzeitzugriff und -archivierung durch Berichte, Workshops und Veranstaltungen zu unterstützen.<sup>46</sup>

Etwas anders angelegt wurde das ERPANET (Electronic Research Preservation and Access) Projekt,<sup>47</sup> das von 2001 bis 2004 lief. Von der Europäischen Kommission im Rahmen des "Framework Programme 5" finanziert, verfolgt es einen deutlich breiter angelegten Ansatz, was sich sowohl in den Zielen als auch den angesprochenen und beteiligten Organisationen ausdrückt. So sollen praxistaugliche Informationen gesammelt und sichtbar gemacht sowie Best-Practice und Fähigkeiten auf dem Gebiet der digitalen Erhaltung des kulturellen Erbes und wissenschaftlicher Forschung gefördert werden.

Im Rahmen dieser Initiative sollten Gedächtnisorganisationen, wie Archive, Museen und Bibliotheken auf der einen mit Forschungseinrichtungen, der Software- und Unterhaltungsindustrie sowie Stellen der öffentlichen Verwaltung auf der anderen Seite zusammengebracht werden. Dazu wurde der Aufbau eines Kompetenznetzwerks mittels einer Kooperationsplattform gestartet, welche als Wissensbasis für aktuelle Entwicklungen und Forschungen dienen soll.

DELOS<sup>48</sup> setzt die Reihe der "Networks of Excellence" für Digitale Bibliotheken fort. Es erhielt hierzu eine Teilfinanzierung der EU-Kommission. Die Hauptaufgaben des Netzwerks liegen im Bereich Forschung, deren Ergebnisse allgemein zugänglich gemacht werden sollen und dem Technologietransfer durch Kooperationsvereinbarungen mit interessierten Partnern. Das Projekt war auf eine Laufzeit von 2004 bis 2007 angelegt.

PLANETS (Preservation and Long-term Access through Networked Services) baut auf den eingangs genannten Initiativen auf und vereinigt deshalb eine Reihe bereits bekannter Akteure aus Großbritannien, den Niederlanden und Österreich. Es umfasst Mitglieder der Europäische National Bibliotheken und Archive, Forschungsinstitutionen und Technologiefirmen und adressiert die Herausforderungen des dauerhaften Zugangs und der Erhaltung des digitalen kulturellen und wissenschaftlichen Wissens.

Das PLANETS-Projekt<sup>49</sup> wurde ursprünglich von der Königlichen Bibliothek der Niederlande, dem Nationalarchiv und der British Library initiiert. Der Projektstart erfolgte dann zum 1. Juni 2006. Angelegt ist PLANETS auf vier Jahre, der finanzielle Umfang, gefördert im Rahmen des "Framework Programme 6" beträgt ca. 14 Mio. Euro. Insgesamt sind mehr als 100 Wissenschaftler und Angehörige der Partnerorganisationen am Vorhaben beteiligt. Das Projekt repräsentiert drei große Gruppen aus insgesamt fünf Nationen:

- Staats- und Nationalbibliotheken sowie -archive: Königliche Bibliothek der Niederlande und das Niederländische Nationalarchiv, Österreichische Nationalbibliothek, British Library und The National Archive GB, die beiden Nationalbibliotheken Dänemarks und das Schweizerische Bundesarchiv.
- Universitäten und Forschungseinrichtungen bzw. Auftragsforschung: University of Glas-

<sup>46</sup>Homepage des Projekts <http://www.digicult.info>, siehe auch "Core Technologies for the Cultural and Scientific Heritage Sector", DigiCULT Technology Watch Report 3, January 2005.

<sup>47</sup>Vgl. Projekt-Homepage - <http://www.erpanet.org>.

<sup>48</sup>DELOS Network of Excellence, siehe <http://www.delos.info>.

<sup>49</sup>Siehe Beschreibung der Zielsetzungen auf der Homepage [PLANETS 2008] unter <http://www.planets-project.eu/about>.

gow, Technische Universität Wien, Austrian Research Center (vergleichbar mit Fraunhofer Instituten), Universität zu Köln, Universität Freiburg und die niederländisch/britische Tessella (Technologie-Support).

- Große Unternehmen der IT-Branche: IBM mit Forschungen zu UVM (Universal Virtual Machine) und Microsoft für die Unterstützung bei den älteren Dokumentenformaten.

Die Ziele von PLANETS liegen wie schon in vielen anderen der vorgestellten Initiativen im Zusammenbringen von Anwendern und Forschern, der Durchdringung des Problemfeldes und der Zusammenstellung und Verbreitung des bestehenden Wissens. Dabei erhoffen sich die beteiligten Bibliotheken und Nationalarchive Antworten auf Fragen zur Langzeitarchivierung und Handlungsrichtlinien für ihre digitalen Bibliotheken. Insgesamt werden sechs Teilprojekte bearbeitet:

- Preservation Planning - Erstellung von Tools zur Überprüfung von Entscheidungen und Voraussagen, Entwicklung von Werkzeugen und Umsetzung der proaktiven LA-Planungen, Erarbeitung von Vorschlägen zur Erweiterung des bestehenden OAIS Modells (Open Archival Information System) siehe hierzu [Strodl u. a. 2007].
- Preservation Action - Generelle Beschreibung, Entwicklung, Bewertung von Archivierungsstrategien, Beschreibung, Entwicklung, Bewertung von Werkzeugen, am Objekt ansetzen oder die Ablaufumgebungen für unveränderte digitale Objekte bereitstellen.
- Preservation Characterization - Entwicklung von Strategien und Methoden zur (abstrakten) Objektbeschreibung, Objektbeschreibungsdatenbank als Kern zur Einbindung ins Interoperability Framework.
- Interoperability Framework - Auswahl möglicher Architekturen (Technologien, Software, Programmiersprachen), Service Interfaces zur Verbindung der verschiedenen Komponenten, Schemata für und Einbindung der verschiedenen Datenbanken der anderen Subprojekte.
- Testbed - Aufsetzen und Betreiben einer Umgebung für Testaktivitäten und Drittanbieter-Zertifizierungen.
- Dissemination and Takeup - Kontakt zu Technologiepartnern, Softwareherstellern, Entwickler-Communities, Training, Weiterbildung, Workshops, Information.

CASPAR, "Cultural, Artistic and Scientific Knowledge for Preservation, Access and Retrieval",<sup>50</sup> wird ebenfalls von der EU im Rahmen des IST kofinanziert und startete ebenfalls im Jahr 2006 als Komplementärprojekt parallel zu PLANETS. Viele Fragestellungen und einige Antworten überschneiden sich durchaus. Der Hauptfokus liegt auf dem Ziel verschiedene Partner aus sehr unterschiedlichen Bereichen des kulturellen Lebens, der Wissenschaft und dem künstlerischen Bereich zusammenzubringen. Das Projekt definiert hierfür folgende Ziele:

- Aufbau eines Langzeitarchivs basierend auf dem OAIS-Modell.

<sup>50</sup>Siehe <http://www.casparpreserves.eu>. Eine Erörterung des Forschungsziels mit Bezug zum OAIS bietet zudem [Factor u. a. 2007].

- Demonstration, dass ein solches Langzeitarchiv die Bedürfnisse der unterschiedlichen Nutzergruppen erfüllen kann.
- Anpassung und Integration aktueller Technologien für die Nutzung zur digitalen Langzeitarchivierung.
- Entwicklung von technischen Lösungen für nachhaltige Erweiterungen.

DPE, Digital Preservation Europe,<sup>51</sup> ist eine Fortsetzung des ERPANET Projekts im IST-Programm der EU. Diese Initiative kümmert sich um Koordination, Integration von Forschungsinitiativen, Aus- und Weiterbildung sowie den Aufbau von Arbeitsgruppen zum Thema.

Die drei letztgenannten EU-weiten Aktivitäten haben nach den breiter angelegten Überblicksprojekten zur Schaffung des Problembewusstseins das Ziel, neben weiterer Forschung und Information prototypische Software für die Nutzung in der Langzeitarchivierung zu schaffen.

In Großbritannien spielt das Digital Curation Centre<sup>52</sup> eine wichtige Rolle in der aktuellen Diskussion zur Gestaltung der digitalen Langzeitarchivierung. Es behandelt Fragestellungen zur langfristigen Sicherstellung von Zitaten, die auf Datenbanken und Online-Ressourcen verweisen. Weitere Inhalte sind Datenintegration und geeignete Publikation, automatische Extraktion von semantischen Metadaten zur besseren Beschreibung, beispielsweise von PDF-Dokumenten. Zudem geht es um die Verifikation von OAIS-basierten Archivierungsansätzen.

Besonders hervorgerufen für die Emulation als Langzeitarchivierungsstrategie hatte sich das britisch-amerikanische CAMILEON<sup>53</sup> Projekt, welches gemeinsam von den Universitäten Michigan und Leeds mit Förderung der JISC und NSF<sup>54</sup> von 1999 bis 2003 durchgeführt wurde. Der Fokus der Arbeiten lag im Bereich der Ermittlung von Möglichkeiten zur Langzeiterhaltung der ursprünglichen Funktionalität und Aussehen digitaler Objekte, der Untersuchung von Emulation als Langzeitarchivierungsstrategie und der Einordnung von Emulation in die digitale Langzeitarchivierung. In diesem Projekt wurde einerseits diskutiert, digitale Objekte in eine bestimmte Langzeiterhaltungsform zu bringen und dann die für diese Ebene notwendige Nutzungsumgebung zu erhalten. Auf der anderen Seite wurden Varianten der Migration diskutiert, welche weniger stark die Authentizität der Objekte gefährden.<sup>55</sup>

### 2.6.3 Internationale Aktivitäten

Neben den deutschen und europäischen Initiativen und Projekten existieren eine Reihe internationaler Entsprechungen: So wurde 2001 in Großbritannien Digital Preservation Coalition

<sup>51</sup>Diese Initiative, gestartet 2006, vereinigt Universitäten, Nationalbibliotheken, Ministerien und Archive, siehe <http://www.digitalpreservationeurope.eu>.

<sup>52</sup>Projektbeginn 2004; siehe <http://www.dcc.ac.uk>.

<sup>53</sup>„Creative Archiving at Michigan & Leeds: Emulating the Old on the New“, vgl. hierzu <http://www.si.umich.edu/CAMILEON/about/aboutcam.html>. Es ergänzte das CEDARS-Projekt - <http://www.leeds.ac.uk/cedars>, das von 1998 bis 2002 lief und sich mit der Erstellung, Langzeitbewahrung, Zugriff und ausreichenden Beschreibung (Metadatenschema) von digitalen Objekten befasste. Einen guten Überblick zu den Ideen und Aktivitäten gibt zudem [Holdsworth und Wheatley 2004].

<sup>54</sup>Bezeichnet das Britisches Joint Information Systems Committee und die National Science Foundation der Vereinigten Staaten.

<sup>55</sup>Vgl. hierzu die Aufsätze [Holdsworth und Wheatley 2001] und [Wheatley 2001].

(DPC)<sup>56</sup> gegründet. Sie hat das Ziel, die Herausforderungen der Langzeitarchivierung und -verfügbarkeit digitaler Quellen aufzugreifen und die Langzeitverfügbarkeit des digitalen Erbes in nationaler und internationaler Zusammenarbeit zu sichern. Die DPC versteht sich als ein Forum, welches Informationen über den gegenwärtigen Forschungsstand sowie Ansätze aus der Praxis digitaler Langzeitarchivierung dokumentiert und weiterverbreitet. Hierzu zählt die Verbreitung von Information zu aktueller Forschung und Praxis und die Weiterbildung der Mitglieder in Bezug auf Langzeitarchivierung. Zudem sollen Ziele und Fragen des Erhalts digitaler Objekte in das Bewusstsein von Entscheidungsträgern aus Wirtschaft und Politik gebracht werden. Darüber hinaus werden Technologien und Standards für die Langzeitarchivierung gefördert und definiert.

In den Vereinigten Staaten ist die Library of Congress (LoC) führend. Sie übernimmt die typischen Aufgaben einer Nationalbibliothek und verabschiedete im Jahr 2000 ein Programm zum Aufbau einer nationalen digitalen Informationsinfrastruktur samt Programm zur Langzeitverfügbarkeit digitaler Ressourcen.<sup>57</sup> Die Aufgabe dieser Aktivitäten liegt in der Kooperation mit anderen Bibliotheken, Forschungseinrichtungen und kommerziellen Institutionen.

Bei International Research on Permanent Authentic Records in Electronic Systems (InterPARES)<sup>58</sup> handelt es sich um ein internationales Archivprojekt, das vom kanadischen Research Council (SSHRC) und der US-amerikanischen NSF in zwei Phasen gefördert wurde. Zielsetzung war die Entwicklung von theoretischem und methodologischem Wissen für die Langzeitverfügbarkeit und Authentizität digitaler Objekte. Auf dieser Basis sollten Verfahren und Standards entwickelt werden, die die langfristige Erhaltung, den Zugriff und die Authentizität des Archivmaterials zum Ziel haben. In der ersten Projektphase lag der Fokus auf Daten in Datenbanken und Content Management Systemen sowie digitalem Sound. Die zweite Phase beschäftigte sich, neben der Authentizität der Daten, mit der Zuverlässigkeit und Fehlerfreiheit von der Erzeugung digitaler Objekte über den gesamten Lebenszyklus hinweg. In diesem Zuge wurden Authentizitätsanforderungen, Ersteller- und Archivrichtlinien festgelegt, Managementmodelle und Terminologiedatenbanken erstellt. Zudem wurde MADRAS, das Metadata and Archival Description Registry and Analysis System,<sup>59</sup> aufgebaut, welches ein umfangreiches Metadaten-set für die Langzeitarchivierung definiert. Die aktuelle Phase baut auf den beiden vorherigen auf und soll die Anwendung in der Praxis an kleinen und mittleren Archiven überprüfen.

Die Initiative PADI – National Library of Australia's Preserving Access to Digital Information [PADI 2008] – zielt in Australien darauf ab, Mechanismen zu entwickeln, die dafür sorgen, dass digitale Daten sinnvoll im Sinne eines Langzeitzugriffs abgelegt und gehandhabt werden. Diese Initiative gehört zu den ältesten überhaupt und beschäftigt sich seit fast fünfzehn Jahren mit den Fragestellungen des digitalen Erbes. Sie bietet einen nach The-

<sup>56</sup>Siehe [DPC 2001], <http://www.dpconline.org/graphics/about/history.html>.

<sup>57</sup>"Preserving Our Digital Heritage: Plan for the National Digital Information Infrastructure and Preservation Program", Washington: Library of Congress, 2002. [http://www.digitalpreservation.gov/library/pdf/ndiipp\\_plan.pdf](http://www.digitalpreservation.gov/library/pdf/ndiipp_plan.pdf), siehe zu Aktivitäten und Workshops der LoC [Kwon u. a. 2006] zum Aufbau organisationsübergreifender Communities. Zudem werden verschiedene Modelle der Kooperation diskutiert und bewertet.

<sup>58</sup>Siehe Homepage - <http://www.interpares.org>. Die erste Projektphase lief von 1999 bis 2001, die zweite von 2002 bis 2006. Die Phase startete 2007 und ist bis 2012 projektiert.

<sup>59</sup>Projekt-Homepage - <http://www.gseis.ucla.edu/us-interpares/madras>.



men gruppierten Überblick zu aktuellen und abgeschlossenen Aktivitäten sowie Ergebnissen von Forschungsprojekten in ähnlicher Form wie Nestor an. Beide Initiativen kooperieren seit geraumer Zeit miteinander.



## Teil II

# Emulation zur Langzeitarchivierung digitaler Objekte



### 3 Digitale Archive als neuartige Aufgabenstellung

Langzeitarchivierung digitaler Objekte hängt von einer Vielzahl Faktoren und Anforderungen ab. Diese stellen einerseits die verschiedenen Akteure und Nutzer in diesem Gebiet und werden andererseits durch die sehr unterschiedlichen Objekttypen und die damit verbundenen Ansprüche bestimmt. Deshalb liefert der Einstieg in dieses Kapitel einen Überblick zu Langzeitarchivierungsstrategien. Hierzu zählen das OAIS-Referenzmodell für digitale Archive, Standardisierung von Formaten und Prozessen, sowie Strategien, die entweder am Objekt selbst oder seiner Umgebung ansetzen, um auf diese Weise auf den technologischen Wandel zu reagieren.

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 stellt OAIS lediglich grobe Vorgabe zum Aufbau und zur Organisation dar. Deshalb sind die aktuellen Forschungsprojekte und Initiativen mit seiner Verfeinerung und Ergänzung befasst, wie im vorausgegangenen Abschnitt 2.6 dargestellt.

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

Statische und dynamische digitale Objekte sind der Gegenstand von OAIS-basierten Langzeitarchiven. Während für statische Objekte eine Menge von Erfahrungen vorliegen, Strategien schon erprobt und erforscht sind, generieren dynamische Objekte eine Zahl neuer Herausforderungen. Dynamische Objekte prägen zunehmend die digitale Welt: Spiele, große Datenbanksysteme oder ganze Betriebssysteme sind mit klassischen Migrationsansätzen nicht archivierbar.

Unabhängig vom konkreten Objekttyp erfolgt bei der Einstellung in ein OAIS-konformes Archiv eine Anreicherung mit zusätzlichen Informationen, den Metadaten. Inzwischen wurden eine ganze Reihe von Metadaten-Schemata definiert, von denen Dublin Core als flexibles und erweiterbares Modell herausragt und sich im Bereich der digitalen Langzeitarchivierung als

Standard herauskristallisiert.

Das Ziel der Bemühungen um dynamische Objekte liegt in der Wiederherstellung oder Erhaltung alter Ablaufumgebungen, ohne dafür ein Hardwarearchiv dauerhaft aufrecht erhalten zu müssen. Emulation – sie kann an verschiedenen Stellen des Hardware-Software-Stacks ansetzen – ist eine zunehmend stärker wahrgenommene Strategie für den Umgang mit dynamischen Objekten. Von Emulation abzugrenzen sind Virtualisierung und Simulation. Nach der Darstellung wesentlicher Eigenschaften und Vorteile der Emulation erfolgt eine Einordnung ihrer Ansatzpunkte. Der Vergleich der beiden zentralen Strategien Migration und Emulation schließt das Kapitel und damit den allgemeinen Überblick zu Langzeitarchivierungsstrategien ab.

### 3.1 Langzeitarchivierungsstrategien

Nach dem Aufriss der Problemstellung sollen die folgenden Abschnitte die bisher bereits genutzten, sich in Anwendung befindenden und in die Diskussion aufgenommenen Langzeitarchivierungsstrategien behandeln. Dabei geht es einerseits um die Standardisierung digitaler Langzeitarchive generell und andererseits um konkrete Maßnahmen zur Sicherung des langfristigen Zugriffs auf digitale Objekte. Einen kompakten Überblick zur Thematik in Deutschland liefert [Dobratz und Tappenbeck 2002]. Die Autoren von [Holdsworth und Wheatley 2004] berichten über Erfahrungen verschiedener Langzeitarchivierungsprojekte britisch-amerikanischer Akteure. In [Borghoff u. a. 2005] findet sich ein Produktvergleich bestehender Systeme, wie DIAS, DSpace, DigiTool oder MyCoRe.

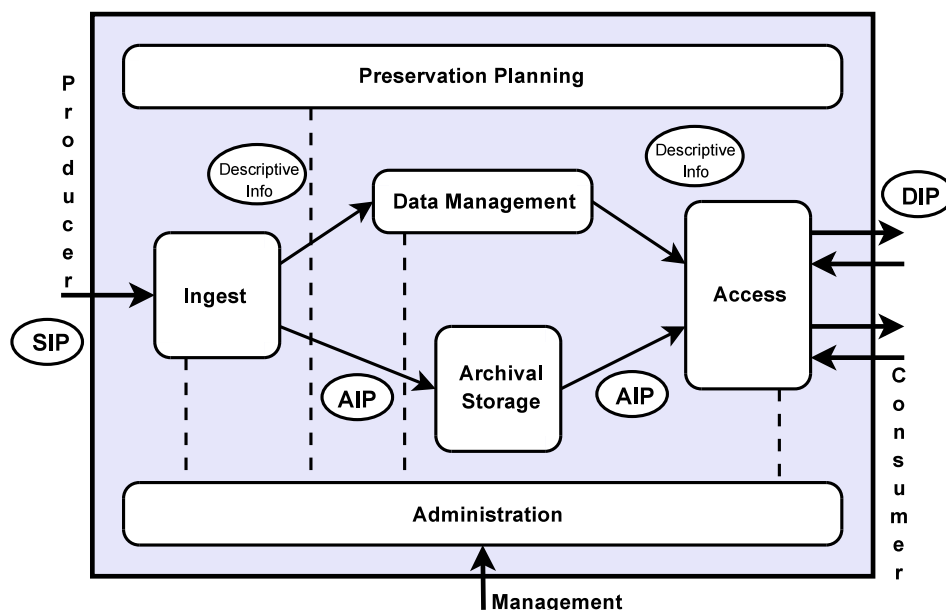
Das im folgenden vorgestellte OAIS-Referenzmodell eines Langzeitarchivs spezifiziert bisher eher notwendige Abläufe und Strukturen denn konkrete Implementierungen. OAIS hat sich als Standard in der Diskussion durchgesetzt, weshalb sich die meisten Projekte zur Langzeitarchivierung auf dieses Modell beziehen und Vorschläge zu seiner Erweiterung<sup>1</sup> erarbeiten. Bisher wird beispielsweise im Modell für das DIP nur gesagt, dass es ein digitales Objekt irgendeiner Art produziert. An dieser Stelle sollten weitere Arbeitsschritte ansetzen, um dem Archivbenutzer die Betrachtung des gewünschten Primärobjekts zu erlauben.

#### 3.1.1 OAIS-Referenzmodell

Das Open Archival Information System (OAIS) spezifiziert einen offenen Standard für die Langzeitarchivierung digitaler Objekte. Dieser Standard ist als Referenzmodell beschrieben, in welchem ein Archiv als Organisation betrachtet wird. Innerhalb dieser Organisation interagieren Menschen und Systeme miteinander, um digitale Archivinhalte einer definierten Benutzerschaft zur Verfügung zu stellen. Primär entworfen wurde OAIS in einer Kooperation der internationalen Organisationen der Luft- und Raumfahrt<sup>2</sup> als Organisationsmodell für digitale Daten. Die ursprüngliche Definition des Standards erfolgte seitens der NASA und anderer Raumfahrtorganisationen. 1999 wurde der Vorschlag zu einem "Referenzmodell für ein

<sup>1</sup>Vgl. hierzu Abschnitt 2.6 des letzten Kapitels, speziell die Agenden der diversen Projekte und Initiativen.

<sup>2</sup>Council of the Consultative Committee for Space Data Systems (CCSDS), siehe Homepage - <http://public.ccsds.org>.



**Abbildung 3.1:** Das OAIS Referenzmodell als organisatorische Struktur mit funktionalen Einheiten.

Offenes Archivierendes Informationssystem (OAIS)<sup>3</sup> der ISO zur Standardisierung vorgelegt. Inzwischen wurde der Vorschlag akzeptiert und unter ISO 14721 beschrieben. OAIS bildet damit eine Richtlinie für weitere Standards. Der Standard spricht von sechs Aufgabenbereichen (Abb. 3.1):

1. Übernahme (Ingest) der digitalen Archivalien durch geeignete Vorbereitung der digitalen Objekte, Bereitstellung der notwendigen technischen Kapazitäten, Bestimmung der Metadaten und Qualitätssicherung. Siehe hierzu Abschnitt 6.1.
2. Datenaufbewahrung (Archival Storage) meint die digitale Speicherung und ihre Organisation. Hier liegen die AIPs, die regelmäßig gewartet und auf Wiederauffindbarkeit der archivischen Informationspakete überprüft werden.
3. Datenmanagement befasst sich mit der Zugänglichkeit der Verzeichnungsinformationen, ihrer Ergänzung und Aufbereitung, sowie der Verwaltung der verschiedenen Archivdatenbanken.
4. Systemverwaltung beziehungsweise Management meint die Beziehungen zwischen Nutzern und Archivaren einerseits sowie dem Archivsystem aus Soft- und Hardwarekomponenten andererseits.
5. Planung der Langzeitarchivierung (Preservation Planning) hat zur Aufgabe, den sicheren Informationszugang zu gewährleisten und diesen für zukünftige Perioden vorzubereiten.

<sup>3</sup>International Standards Organization.

6. Zugriff (Access) ist der Teil des Archivs, über den Nutzer Zugriff auf die sie interessierenden Objekte erhalten. Entsprechend liefert diese Einheit die notwendige Unterstützung bei der Bereitstellung der jeweils geforderten Ablaufumgebungen. Diese Aspekte werden in Kapitel 6 ausführlicher für die Anwendung von Emulationsstrategien erläutert.

OAIS legt damit eine Reihe von Verantwortlichkeiten fest, die sich von denen klassischer Archive unterscheiden. Im Modell werden die Zusammenhänge zwischen den Erzeugern von Archivpaketen, der Verwaltung des Systems und den Archivbenutzer dargestellt (Abb. 3.1). Die Erzeuger, hierbei kann es sich um Personen oder spezielle Software-Clients handeln, liefern die Information, die in das Archiv aufgenommen werden sollen. Die Informationen sind in einem sogenannten Submission Information Package (SIP) zusammengefasst und werden in das Archiv eingespeist. Die späteren Archivinhalte werden in sogenannten Archival Information Packages (AIP) verwaltet. Ein Archivbenutzer erhält die gewünschten Informationen in Form der sogenannten Dissemination Information Packages (DIP) als Objektausgabe. Die Anwender können wiederum Personen oder Client-Systeme sein, die OAIS-Dienste in Anspruch nehmen.

Das Archivmanagement hat die Aufgabe, die OAIS-Richtlinien zu definieren, die für Verwaltungsfunktionen und die Aktualisierung des Archivs gelten. Hierzu zählen Lifecycle-Management der primären Objekte und Preservation Planning für den langfristigen Zugriff. Die Anforderungen, die seitens der Emulationsstrategie auf das OAIS-Management zukommen, sind Thema der Abschnitte 6.7 und 7.2.

Referenzmodell bedeutet jedoch auch, dass die konkrete Implementierung eines solchen Archivs nicht in Form von Spezifikationen festgelegt ist. Es soll dem Verständnis der Problemstellungen der Langzeitarchivierung und -verfügbarkeit dienen und Kriterienkataloge für konkrete Implementierungen liefern. Ebenso liegt damit ein Rahmenwerk für die Beschreibung und den Vergleich verschiedener Systemarchitekturen und Archivierungsverfahren vor. Zudem definiert OAIS eine eigene Terminologie und Konzepte.

Ein zentraler Punkt bei den Diskussionen zur Standardisierung war die "Offenheit" des Ansatzes. Die Entwicklung des OAIS spielt sich in offenen Foren ab. So existiert beispielsweise eine gemeinsame Initiative der europäischen Nationalbibliotheken, die auf Basis des OAIS das DSEP-Prozessmodell (Deposit System for Electronic Publications)<sup>4</sup> entwickelt haben. Diese Offenheit bedeutet jedoch nicht, dass auf OAIS-konforme Archive uneingeschränkte Zugriffe von jedermann erfolgen können. Einen guten Überblick zum Referenzmodell kann man in [Borghoff u. a. 2005], S. 17 f., [Borghoff u. a. 2003] oder [CCSDS 2002] finden.

Das Referenzmodell blieb nicht nur theoretisch, sondern wurde beispielsweise im CEDAR Project [Holdsworth und Sergeant 2000] eingesetzt, welches eine praktische Ausgestaltung des OAIS vorschlug. Weitere Anwender fanden sich mit dem DIAS Project.<sup>5</sup> Darüberhinaus hat das nationale Datenarchiv Großbritanniens (NDAD) seine Abläufe auf das OAIS angepasst. Das vom MIT initiierte DSpace [Tansley u. a. 2003] implementiert ein OAIS-basiertes digitales Archiv nach dem Open-Source-Modell. Ebenso orientiert sich das australische Nationalarchiv im Rahmen des PANDORA-Projektes am Referenzmodell. Das amerikanische Nationalarchiv (NARA) nutzt die OAIS-Modellierung als Basis für eine große Ausschreibung

<sup>4</sup>Vgl. [Steenbakkers 2000] und [van der Werf 2000].

<sup>5</sup>DIAS/e-Depot ist als OAIS-konformes elektronisches Archiv konzipiert [Rothenberg 2000], [KB 2007], [van Diessen und Steenbakkers 2002] bzw. [Oltmans u. a. 2004].



zur Entwicklung des Electronic Records Archivs. Darüberhinaus werden verschiedene Vorschläge zur Konkretisierung und für Erweiterungen gemacht.<sup>6</sup>

### 3.1.2 Digitalisierung analoger Medien

Digitalisierung ist eine Langzeitarchivierungsstrategie für analoge Medien: So lassen sich Objekte, wie Bücher, die auf säurehaltigem Papier gedruckt wurden, Schriften, die aufgrund ihrer Fragilität nicht mehr direkt zugreifbar sind, digital archivieren und dem Anwender präsentieren. Digitalisierung ist eine der Quellen<sup>7</sup> aus denen derzeit im Umfeld der Gedächtnisorganisationen digitale Objekte entstehen. Letztere werden als Digitalisate bezeichnet. Einen guten Überblick zur Fragestellung hat beispielsweise [nestor 2008] zusammengestellt.

Digitalisierung als Langzeitarchivierungsstrategie spielt im Zuge dieser Arbeit keine Rolle, da bei der Betrachtung des OAIS bereits von nur-digitalen Objekten ausgegangen wird. Der Erfolg dieser Strategie ist jedoch eng mit dem langfristigen Zugang zum elektronischen Archiv verknüpft.

Neben den greifbaren Vorteilen der Vervielfältigung, der Suchbarkeit und des leichteren Zugriffs, sollten Nachteile der schwindenden Authentizität oder des reduzierten Gesamteindrucks<sup>8</sup> nicht verschwiegen werden. Der Vorgang der Digitalisierung ist eigentlich immer mit Informationsverlust verknüpft, da beispielsweise aus praktischen und ökonomischen Erwägungen die Scanner-Optik nicht eine beliebig feine Auflösung eines Objekts abfilmen kann und bei der Digitalisierung von Audio- und Videomaterial systembedingt Reduktionen vorgenommen werden. Hinzu kommen zusätzliche Eigenschaften einiger resultierender Datenformate, die bei der Digitalisierung in die Überlegungen einbezogen werden sollten:

**Verlustfreie Kompression** Zur Beherrschung wachsender Datenmengen werden Kompressionsverfahren eingesetzt. Hierbei ist zwischen verlustfreien Verfahren, wie GZIP oder BZIP2 zu unterscheiden, die Daten von einer logischen Struktur in eine andere umkehrbar abbilden. Damit die Umkehrung der Kompression auch nach größeren Zeitabständen erfolgen kann, muss Sorge getragen werden, dass die Werkzeuge verfügbar bleiben.

**Verlustbehaftete Kompression** Viele Datenformate, besonders die diversen Multimedia-Daten wie Video und Audio, liegen oft schon in komprimierter Form vor oder lassen sich mit den eben genannten Verfahren nicht sinnvoll verkleinern. Für Bilder, Video- und Audiodatenströme gibt es spezielle, verlustbehaftete Kompressionsverfahren. Diese erreichen oft eine wesentliche Reduktion der Ausgangsdatenmenge. Vielfach ist das Endergebnis jedoch eng mit einer bestimmten Applikation verknüpft und liegt in einem proprietären Datenformat vor. Dann muss dafür Sorge getragen werden, dass die spezielle Applikation weiterhin verfügbar bleibt. Ein nicht zu unterschätzender Faktor ist die Qualitätsreduktion der Ergebnisdaten,

<sup>6</sup>Vgl. hierzu die Aktivitäten der verschiedenen Initiativen in Abschnitt 2.6 oder auch den Erweiterungs- und Implementierungsvorschlag in [Rödig u. a. 2003].

<sup>7</sup>Vgl. hierzu [Endres und Fellner 2000].

<sup>8</sup>Haptik von Bucheinschlägen, Objektgrößen, sichtbares Alter und Abnutzung, Anordnung von Teilobjekten zueinander. Digitalisierung wird teilweise mit der "Rettung" des Ausgangsobjekts gleichgesetzt, was von vielen jedoch nicht uneingeschränkt so gesehen wird. Siehe auch [Bárány 2006].

welche sich in niedrigeren Auflösungen, geringerer Farbtiefe oder höheren Rauschwerten niederschlagen kann.

### 3.1.3 Standardisierung

In Bibliotheken und Archiven ist der überwiegende Bestand noch nicht-digital. Er wird jedoch durch neue Einlieferungen und Digitalisierung zunehmend einen elektronischen Charakter erhalten. Hier gilt es, im Interesse aller Beteiligten einen Überblick zu bewahren. Eine eher auf die Zukunft ausgerichtete Langzeitarchivierungsstrategie ist daher die Entwicklung von Dokumenten- und Objektstandards. Wird bereits bei der Erstellung von digitalen Objekten auf ihre langfristige Zugreif- und Verwertbarkeit fokussiert, lassen sich einige der zu Beginn dieser Arbeit beschriebenen Probleme vermeiden oder zumindestens abschwächen. Dabei kommt es darauf an, Standards in internationalem Rahmen zu entwickeln.

Wenn bestimmte Standards beschlossen und verabschiedet werden, wirken sich diese typischerweise nur auf Daten aus, die ab diesem Zeitpunkt geschrieben werden. Unabhängig davon, ob es sich "nur" um statischen Daten handelt, eine Festlegung auf Langzeitdatenformate arbeitet nicht automatisch rückwärts auf bereits erstellte und abgelegte Objekte. In [Rothenberg 2002] werden Standards jedoch sehr kritisch beleuchtet und ihre Grenzen aufgezeigt.

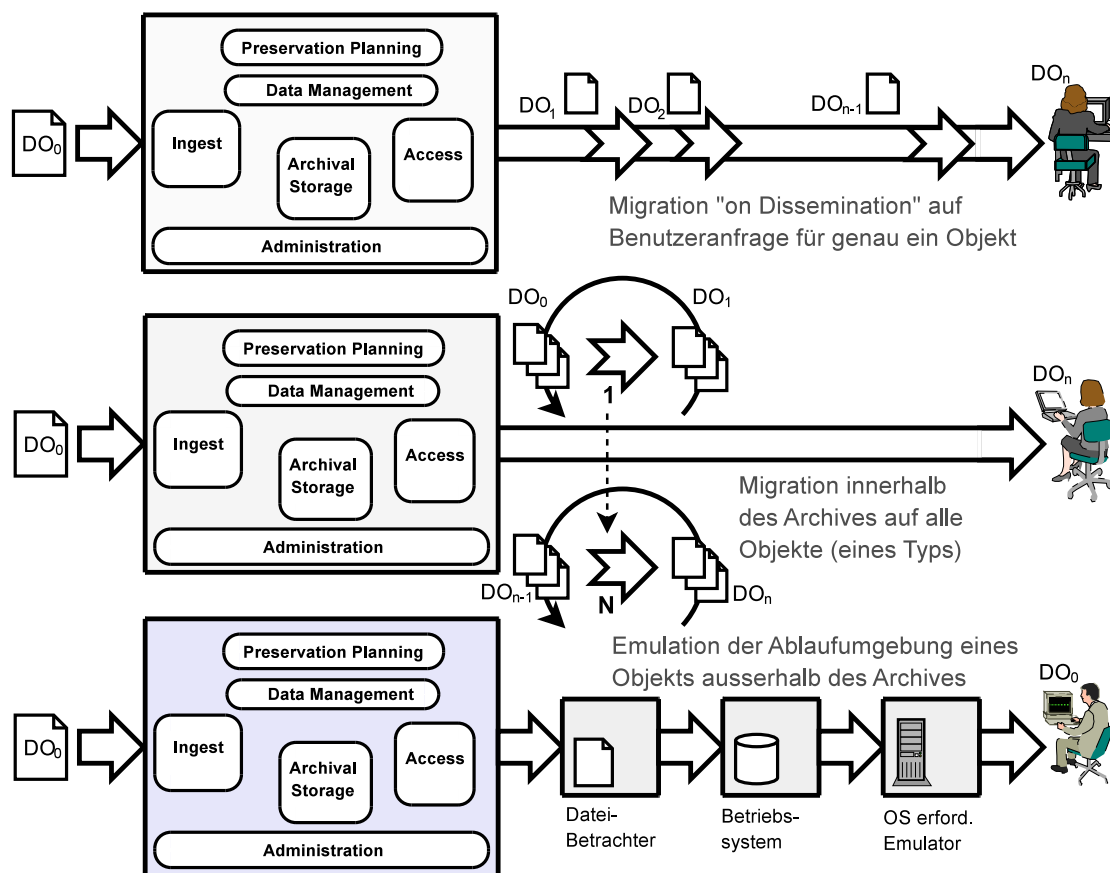
Die bisherigen Erfahrungen zeigen, dass sich offene Standards auf breiter Ebene durchsetzen und relativ stabil halten können, wie beispielsweise TCP/IP als Netzwerkprotokoll. Es bildet die Grundlage des weltweiten Internets und hat andere Netzwerkprotokolle, wie beispielsweise IPX/SPX oder Appletalk, komplett verdrängt. Problematischer sind Standards, die sich aufgrund von Monopolstellungen herausbilden. So weisen beispielsweise die Microsoft Dokumentenformate eine hohe Dynamik bei schlechter Dokumentation auf. Die Lage sieht etwas besser beim Portable Document Format (PDF)<sup>9</sup> aus, welches gleichfalls von nur einer Firma kontrolliert wird. Sicherlich zu begrüßen ist die zunehmende Verbreitung herstellernabhängiger Normstandards, wie die eXtensible Markup Language (XML). Diese haben zudem für Metadaten eine nicht unerhebliche Bedeutung [Dobratz und Tappenbeck 2002].

### 3.1.4 Migration und Emulation

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 Plattformen zugreifbar zu halten. Demgegenüber setzt Emulation an der Umgebung für das jeweilige Objekt an. Migration verändert also 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 (Abb. 3.2). Migration hat jedoch, wie in den weiteren Abschnitten dieses Kapitels gezeigt wird, einige Nachteile, die eine Anwendung bei bestimmten Objekttypen weitgehend verhindern.

---

<sup>9</sup>PDF-Referenz der Firma Adobe Version 1.7 oder 1.4 als Monographie [Adobe 2001]. Inzwischen wurde PDF als Standard bei der ISO vorgeschlagen. Für die Dokumentenlangzeitaufbewahrung hat Adobe den speziellen PDF-A-Standard geschaffen.



**Abbildung 3.2:** 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.

Jedoch hängt auch die Emulation von einigen Erfolgsbedingungen ab. Eine gute Diskussion der verschiedenen Strategien, ihrer Vor- und Nachteile findet sich in [Granger 2000] oder ebenso in [Hedstrom und Lampe 2001].

## 3.2 Statische und dynamische digitale Objekte

Mit sich verändernden Technologien, Anwendungen und anderer Computernutzung wandelt sich zwangsläufig die Art der digitalen Objekte. Im Laufe der Zeit kommen neue Objekttypen hinzu und andere werden ersetzt. Dieser Abschnitt nimmt die Auseinandersetzung von Unterkapitel 2.3.1 unter dem Aspekt der Langzeitarchivierbarkeit wieder auf. In einem länger existierenden digitalen Langzeitarchiv findet man daher AIP, die Objekte sehr unterschiedlichen Charakters enthalten.

### 3.2.1 Klassische Daten

Mit Daten des klassischen Typs seien digitale Objekte in Form virtueller Dokumente gemeint. Diese sind Objekte, die nicht mehr primär in Papierform erstellt werden, aber für

eine eventuelle Wiedergabe auf Papier gedacht sind. Hierzu zählt alles, was in Behörden, Unternehmen und Organisationen an Korrespondenz, Berichten, Abrechnungen, Strategiepapieren anfällt. Für diese Daten ist oft nur sicherzustellen, dass sie innerhalb ihrer gesetzlichen Aufbewahrungsfrist eingesehen werden können. Einige Daten von gesellschaftlichem oder geschichtlichem Langzeitinteresse sollten diese Fristen überdauern können.

Eines zeichnet die genannten Objekte jedoch aus: Sie lassen sich zumindest theoretisch zu jedem Zeitpunkt auf einen extradigitalen Datenträger, wie Papier oder Mikrofiche umkopieren. Audio- und Videostücke lassen sich auf analoge Datenträger kopieren. Konstruktionspläne können ausgedruckt werden, jedoch gehen typischerweise Informationsebenen und Zusammenhänge verloren. Die genannten "traditionellen Datenträger" benötigen geringen oder gar keinen technischen Aufwand, damit Endbenutzer die enthaltenen Informationen lesen und auswerten können. Dieses gilt nicht mehr uneingeschränkt für dynamische Daten und interaktive Objekte. Eine gute Darstellung der Vervielfältigung der Verbreitungswege, des Diskurses und den Einfluss auf die Art der Objekte und die daraus resultierenden Herausforderungen für Gedächtnisorganisationen liefert die Einleitung von [Mittler 2005].

### 3.2.2 Dynamische Daten, interaktive Objekte

Die soeben vorgestellten Kategorien digitaler Objekte haben zumindest theoretisch noch eine analoge Entsprechung. Ganz anders liegt der Fall bei dynamischen Objekten. Sie finden in der aktuellen Langzeitarchivierungsdebatte bisher noch geringe Beachtung. Bei diesen Objekten handelt es sich um ausführbare Programme oder Softwarebibliotheken, die überhaupt erst den sinnvollen Betrieb von Computern erlauben. Zu ihnen zählen:

- Betriebssysteme
- Anwendungsprogramme
- Computerspiele
- Interaktive Medien und Unterhaltung
- Datenbanken

Dynamische Daten zeichnen sich dadurch aus, dass sie außerhalb eines festgelegten digitalen Kontexts heraus 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. [Thibodeau 2002], S. 16 *"On the »preserve technology« end, one would place preserving artifacts of technology, such as computer games. Games are meant to be played. To play a computer game entails keeping the program that is needed to play the game operational or substituting an equivalent program, for example, through reverse engineering, if the original becomes obsolete."*

### 3.2.3 Betriebssysteme und Komponenten

Betriebssysteme zählen zu den Grundkomponenten eines jeden Rechners. Sie regeln die Interaktion mit der Hardware, kümmern sich um Ressourcenverwaltung und -zuteilung und erlauben die Steuerung durch den Endanwender. Sie sind im kompilierten Zustand angepasst an bestimmte Prozessoren, die Art der Speicheraufteilung und an bestimmte Peripheriegeräte zur Ein- und Ausgabe. Deshalb können sie nur auf einer bestimmten Rechnerarchitektur ablaufen. Da eine Reihe von Funktionen von verschiedenen Programmen benötigt werden, sind diese oft in sogenannte Bibliotheken ausgelagert. Programme, die nicht alle Funktionen enthalten, laden benötigte Komponenten aus den Bibliotheken zur Laufzeit nach. Bibliotheken und Programme hängen dementsprechend eng miteinander zusammen.

### 3.2.4 Anwendungsprogramme

Oberhalb der Betriebssystemebene<sup>10</sup> finden sich die Anwendungen. Diese sind Programme, die für bestimmte, spezialisierte Aufgaben erstellt wurden. Mit diesen Programmen generieren und bearbeiten 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 Betriebssystem übernimmt die Speicherung von Dateien auf Datenträgern üblicherweise angeordnet in Verzeichnissen. Zur Ausführung auf einer bestimmten Rechnerarchitektur werden Betriebssysteme und Applikationen aus dem sogenannten Quellcode in den passenden Binärcode übersetzt. Deshalb können Programme und Bibliotheken nicht beliebig zwischen verschiedenen Betriebssystemen verschoben werden.

### 3.2.5 Computerspiele

Darüber hinaus existiert eine ganze eigene Klasse von Anwendungsprogrammen, die zur reinen Unterhaltung geschaffen wurden: Computerspiele sind ein völlig neues Medium, welches im Freizeitverhalten der Menschen eine zunehmende Bedeutung erlangt. Computerspiele gibt es schon erstaunlich lange, ihre ersten Vertreter tauchten Anfang der 1970er Jahre auf.<sup>11</sup> Interessanterweise arbeiteten sie oft am oberen technischen Limit der aktuell verfügbaren Hardware. Für viele Nutzer bedeuteten Computerspiele die Einführung des Rechners in den Haushalt und waren Ausdruck für den schnellen technischen Fortschritt des ausklingenden zwanzigsten Jahrhunderts.

Der nur-digitale Hintergrund der Computerspiele bedeutete, dass nur die ablauffähigen Daten den Sinn des Programms erkennen ließen. Ist ein Spiel nicht mehr ausführbar, muss man es als komplett verloren betrachten, da noch nicht einmal Ergebnisse seiner Existenz, von Spielständen und Screenshots vielleicht einmal abgesehen, nachvollziehbar bleiben. Hier gilt zudem, dass Mitschnitte von Spielabläufen, je nach Art eines Computerspiels, nur sehr unvollständig Auskunft geben können und sich daher für wissenschaftliche Untersuchungen

---

<sup>10</sup>Für die schematische Darstellung der Funktionsweise eines Computers wird oft ein Schichtenmodell gewählt, siehe hierfür Abbildung 3.9.

<sup>11</sup>Vgl. hierzu [CSM 2008]. Der Bereich "Online Expo Spielmaschinen" startet mit dem Jahr 1972.



**Abbildung 3.3:** Beispiel eines beliebten Computerspiels ("Moorhuhn") als Flash-Applikation, die im Web-Browser abläuft. Schon an dieser abgedruckten Darstellung lässt sich nachvollziehen, dass eine rein statische Archivierungsform ungenügend ist.

in diesem Feld nur sehr bedingt eignen. Viele dieser Programme sind Zeichen ihrer Epoche und vergleichbar mit anderen Zeitzeugen.

### 3.2.6 Interaktive Medien und Unterhaltung

Mit der breiten Durchsetzung des Computers und der begleitenden Revolution auf dem Gebiet der Computernetze 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 Webseiten unterstützt von Multimedia-Formaten, wie Flash (Abb. 3.3) dominieren die großen Sites im Internet. Viele Webanwendungen führen den Hauptteil der Interaktion auf dem Server des jeweiligen Inhaltenanbieters aus. Viele Anwendungen funktionieren nur im Zusammenspiel der Server mit speziellen Clients.<sup>12</sup>

<sup>12</sup>Waren in der Anfangszeit HTTP-Clients, die Web-Browser, einfache Applikationen zum Darstellen von Text und Bildern, so sind sie inzwischen mächtige Anwendungen, die eine Vielfalt verschiedener Medienformate, wie Audio, Video, Java- und Java-Script-Applikationen ablaufen lassen können. Während sich Java als

Erweitert wird dieser Bereich durch Computerkunst oder die Nutzung des Mediums als Lernplattform. Auch hier gilt: Die Abläufe von E-Learning-Systemen oder einer Computerkunst-Installation einfach abzufilmen, wird in den meisten Fällen ziemlich unbefriedigend sein. Aus interaktiven Applikationen mit einer Vielzahl möglicher Navigationspfade wird so eine lineare Darstellung eingeschränkter Komplexität und Aussagekraft.

### 3.2.7 Datenbanken

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-Anwendungen, elektronische Fahrpläne diverser Verkehrsträger bis hin zu Content Management Systemen (CMS) moderner Internet-Auftritte von Firmen und Organisationen. Wenn von einer Datenbank, einer Tabellenkalkulation oder einer Konstruktion sehr verschiedene Ansichten ad-hoc erzeugt werden können, ist sehr schwer abzusehen, welche dieser Ansichten zu einem späteren Moment noch einmal benötigt werden. Unter Umständen hat man sich auf Teilmengen festgelegt, die von nachfolgenden Betrachtern als unzureichend oder irrelevant eingestuft werden. Gerade bei Konstruktionszeichnungen wichtiger langlebiger Erzeugnisse wie Flugzeugen, Schiffen oder Gebäuden besteht großes allgemeines Interesse eines zeitlich unbeschränkten Zugriffs.

## 3.3 Metadaten zur Beschreibung des Archivinhaltes

Metadatenformate sind im Zuge dieser Arbeit insofern von Bedeutung, als dass sie für die Erhebung und Speicherung zusätzlicher Informationen zu Emulatoren sowie Betriebssystemen und Programmen zur Wiederherstellung von Nutzungsumgebungen in Form von Zusatzinformationen und Handlungsanleitungen notwendig werden. Sie sollten in geeigneter Weise mit Bezug auf das beschriebene digitale Objekt abgelegt werden. Erste Überlegungen und Vorschläge finden sich bei [Rothenberg 1996].

Digitale Objekttypen unterscheiden sich wesentlich voneinander. Dabei sieht man einem Objekt nicht "von außen" an, um welchen Typ es sich handelt, welche Informationen in ihm abgelegt sind oder welche Aufgabe das Objekt im Verhältnis zu anderen Objekten erfüllt. Zusätzliche, näher beschreibende, katalogisierende Informationen bezeichnet man als Metadaten oder Metainformationen. Sie sind auch bei klassischen Archivalien, wie Büchern oder Kartenmaterial, nicht direkt mit dem jeweiligen Objekt verknüpft oder an dieses bindbar. Das hat mehrere Gründe:

- Metainformationen dienen der Auffindbarkeit von Objekten und sind deshalb suchbar, beispielsweise in Katalogsystemen abzulegen.

---

weitgehend plattform-unabhängige Programmiersprache für diesen Bereich durchgesetzt hat, ist die Vielfalt der bereitgestellten Medienformate deutlich größer. So versucht beispielsweise Microsoft mit "Silverlight" eine Alternative zu Adobe Flash zu etablieren.

The screenshot shows the Deutsche Bahn website interface in a Mozilla Firefox browser. The page is titled "Verbindungen - Ihre Auskunft - Mozilla Firefox". The address bar shows the URL "http://reiseauskunft.bahn.de/bin/query.exe/dn?ld=212.87&seqn". The page features a navigation bar with links like "Preise&Angebote", "Planen&Buchen", "Mobilität&Service", "Reisebüro", "Internat. Guests", and "Kon". Below the navigation bar, there is a section for "Ihre Angaben" (Your details) showing a journey from Freiburg(Breisgau) Hbf to Göttingen on Friday, 11.01.08, at 19:00. The page also displays a table of "Ihre Fahrtmöglichkeiten" (Your travel options) sorted by departure time. The table includes columns for Bahnhof/Haltstelle, Datum, Zeit, Dauer, Umst., Produkte, and Preis. The first option shows a journey from Freiburg(Breisgau) Hbf to Göttingen on Friday, 11.01.08, at 18:57, with a duration of 4:06, no transfers, and a price of 91,00 EUR. The second option shows a journey from Freiburg(Breisgau) Hbf to Göttingen on Saturday, 12.01.08, at 19:56, with a duration of 9:10, 3 transfers, and a price of 91,00 EUR. The third option shows a journey from Freiburg(Breisgau) Hbf to Göttingen on Saturday, 12.01.08, at 00:14, with a duration of 6:41, 1 transfer, and a price of 91,00 EUR. The page also includes a "Zur Buchung" (Book now) button for each option.

| Bahnhof/Haltstelle                  | Datum                        | Zeit                 | Dauer | Umst. | Produkte | Preis                                   |
|-------------------------------------|------------------------------|----------------------|-------|-------|----------|-----------------------------------------|
| Freiburg(Breisgau) Hbf<br>Göttingen | Fr, 11.01.08                 | ab 18:57<br>an 23:03 | 4:06  | 0     | ICE      | 91,00 EUR<br>Normalpreis<br>Zur Buchung |
| Freiburg(Breisgau) Hbf<br>Göttingen | Fr, 11.01.08<br>Sa, 12.01.08 | ab 19:56<br>an 05:06 | 9:10  | 3     | ICE, EN  | Preisauskunft nicht möglich             |
| Freiburg(Breisgau) Hbf<br>Göttingen | Sa, 12.01.08<br>Sa, 12.01.08 | ab 00:14<br>an 06:55 | 6:41  | 1     | ICE      | 91,00 EUR<br>Zur Buchung                |

**Abbildung 3.4:** Der Fahrplan der Deutschen Bahn ist ein gutes Beispiel einer großen Datenbank: Wo früher in Form von Kursbüchern Jahresfahrpläne archiviert wurden, ergeben sich an dieser Stelle komplett neue Herausforderungen.

- Sie können sinnvollerweise genutzt werden, Zugriffsrechte auf ein Objekt zu definieren. Diese lassen sich je nach Benutzergruppe unterschiedlich festlegen oder sich im Laufe der Zeit ändern. So erlöschen bestimmte Urheberrechte oder Patente nach Ablauf einer bestimmten Frist.
- Komplexe Objekte können aus einer Anzahl von einzeln identifizierbaren Objekten bestehen, die separat beschrieben werden sollen.

Metadaten spielen im Zusammenhang dieser Arbeit als Teil der Auswahlkriterien von Werkzeugen für die Langzeitarchivierung eine Rolle. Metadatenschemata sollten deshalb in der Lage sein, notwendige Zusatzinformationen für besagte Kriterien und ihre langfristige Verwendung aufzunehmen. Die Beschäftigung mit Metadaten erfolgt deshalb im Zusammenhang einer ganzen Reihe von Forschungsprojekten und Initiativen.<sup>13</sup>

<sup>13</sup>Siehe hierzu auch 2.6 im vorangegangenen Kapitel oder die Anwendung im DSpace [Tansley u. a. 2003], S. 88, 90, sowie die Überlegungen in [Rothenberg 2000], S. 26 ff.



### 3.3.1 Dublin Core Metadata Set

Der Bedarf an Metadaten besteht nicht erst seit der Beschäftigung mit Langzeitarchivierungsfragestellungen. Gerade mit dem explosionsartigen Wachstum online verfügbarer Informationen und Dokumente stieg die Nachfrage nach zusätzlichen, genormten Beschreibungen von digitalen Objekten deutlich an. So gründete sich bereits 1994, noch vor dem großen Durchbruch des World Wide Web (WWW), die Dublin Core Metadata Initiative am Rande einer WWW-Konferenz.

Diese Initiative mündete ein Jahr später in einer Konferenz zur Informationserschließung und -beschreibung. Sie tagte in Dublin im amerikanischen Bundesstaat Ohio. Deshalb gab man einfach der beschlossenen Grundmenge (Kernmenge) von Beschreibungstermen den Namen "Dublin Core Metadata". Diese Terme bezogen sich in erster Linie auf Web-Ressourcen, die kategorisiert und von den gerade aufkommenden stichwortbasierten Suchmaschinen besser gefunden werden sollten.

Die 15 Kernelemente, die um zusätzliche und weitergehend erklärende Felder erweitert werden konnten, erregten zügig das Interesse von Archivaren und Bibliothekaren. Deshalb kam es zu einer schnellen internationalen Verbreitung und einer Standardisierung<sup>14</sup> durch die ISO.

DCMI Metadata Terms sind sehr flexibel und ohne feste Reihung angelegt: Jedes Feld ist optional und darf auch häufiger auftauchen. Durch die zusätzlich erlaubten detaillierenden Felder lassen sich auf spezielle Bedürfnisse zugeschnittene Kategorisierungen erreichen.

**Eindeutige Kennung** Jedes Objekt erhält eine eindeutige ID *identifier*, welche seine Identifizierung nach einem zum Objekttyp passenden Katalog, wie URN, DOI oder klassischerweise ISBN/ISSN oder URL/PURL erlaubt. Gegebenenfalls wird der Identifier um ein Feld erweitert, wie das Objekt zitiert werden sollte.

**Inhaltsbeschreibung** Hierzu zählen die Felder:

- *title* legt den Titel des Objekts fest. Unter dieser Bezeichnung wird ein Objekt formell bekannt gemacht. Weitere Titelergänzungen, wie Übersetzungen, Originaltitel oder Abkürzungen gibt man alternativ zum formellen Titel (*alternative*) an.
- *description* liefert eine knappe Zusammenfassung des Inhalts. Hinzu können weitere Elemente, wie das Inhaltsverzeichnis oder Listen weiterer Objektbestandteile kommen, *tableOfContents*. Ebenso sind Verweise auf externe beschreibende Quellen möglich.
- *subject* entstammt optimalerweise einem formalen Klassifikationsschema. Anders als beim *title* stehen in diesem Feld suchtaugliche Schlagwörter, die dem Thema des Inhalts folgen.
- *coverage* dient der Eingrenzung von Bereichen, welches ein Objekt beschreibt oder wofür es gedacht ist. Dieses können zeitliche Beschränkung unter Angabe einer Periode, örtliche Beschreibungen oder Koordinaten sein.

---

<sup>14</sup>Die DCMI empfahl das "Dublin Core Metadata Element Set, Version 1.1" zur Standardisierung, welches als ISO 15836 registriert wurde.

**Technische Daten** dienen in erster Linie Hinweisen zur Verarbeitung oder der Darstellung von Objekten:

- *format* beschreibt das logische Objekt (Abschnitt 2.3) und sein Datenformat (Abschnitt 2.3.2) und gibt damit Hinweise, wie ein Objekt sichtbar gemacht werden kann. Ergänzende Informationen können Laufzeiten (*extent*) oder Medientypen (*medium*) enthalten.
- *language* bezieht sich auf die natürliche Sprache des Objektinhalts. Dieses gilt sicherlich nicht für alle Objekttypen. Jedoch existieren eine ganze Reihe von Objekten, wie Dokumente die in einer bestimmten Sprache abgefasst oder Applikationen, die lokalisiert sind. Zur Kennzeichnung sollte das Sprachkürzel nach ISO 639 und, wenn sinnvoll, ein Länderkürzel<sup>15</sup> nach DIN EN ISO 3166 eingesetzt werden.
- *type* beschreibt die Art oder Gattung eines Objekts, jedoch nicht sein konkretes Datenformat, welches im separaten Feld *format* abgelegt wird. Es sollten bestimmte Termini verwendet werden, von denen das "DCMI Type Vocabulary" eine Reihe vorschlägt. *collection* bezeichnet eine Sammlung von Teilobjekten, die mit eigenen Metadaten versehen sind. Hinzu können nähere Beschreibungen, wie *accrualPolicy* (Kriterien der Zusammenstellung), *accrualMethod* (Verfahren) oder *accrualPeriodicity* (Häufigkeit der Erweiterung) kommen. Weitere Termini sind:
  - Texte ("text") sind klassische statische Dokumente.
  - Audiomaterial ("sound") sind Musik- oder Sprachaufnahmen verschiedener Formen, wie unter anderem Audio-CDs.
  - Bildmaterial ("image") beschreibt die Klasse der Gemälde, Zeichnungen, verschiedene Typen von Fotografien, Drucke, Landkarten, Diagramme oder Noten. Diese erhalten alle als Zweittyp ("still image"). Zu den Vorgenannten kommen Filme, Videos, Animationen oder Fernsehsendungen hinzu, die mit einem weiteren Typenelement ("moving image") als bewegt gekennzeichnet werden.
  - Klassische, reale Objekte können als ("physical object") referenziert werden. Das könnte in der Langzeitarchivierung digitaler Objekte zur Beschreibung realer Hardware dienen.
  - Interaktive Objekte ("interactive resource", siehe vorherigen Abschnitt 3.2), welche vom Anwender Rückmeldungen erfordern, wie Flash-Animationen oder Web-Formulare, sind von Software abzugrenzen. Sie sind jedoch nicht ohne entsprechende Player oder Dienste ablauffähig.
  - Computerprogramme ("software") können in direkt ausführbarer Binärform oder als Quelltext abgelegt werden. Letzteres bietet sich für Open Source Software an und könnte unter Umständen eine spätere Neukompilation für aktuelle Referenzumgebungen erlauben.

<sup>15</sup>Siehe die jeweiligen ISO-Blätter. Der Standard ist durch das Internet Domain Name System sehr verbreitet.

- Datensätze ("dataset") sind beispielsweise wissenschaftliche Primärdaten aus Messungen oder Beobachtungen. Sie sollten in definierter Kodierung vorliegen. Datensätze nur durch eine Verarbeitung im Computer sinnvoll interpretierbar.
- Dienste ("service") sind beispielsweise Web-Services zur Interaktion von Programmen über Computernetzwerke hinweg. Sie involvieren Software an beiden Endpunkten der Verbindung.

**Vernetzung und Einordnung** von Objekten in Gesamtsysteme oder -zusammenhänge dient der Klassifizierung komplexerer und voneinander abhängiger Objekte und Sammlungen:

- *source* verweist auf ein Objekt (bezeichnet mit *identifier*), von dem das beschriebene Objekt in Teilen oder komplett abstammt. Ähnlich gelagert ist *relation*.
- *relation* beschreibt Beziehungen zwischen verschiedenen Objekten, welche unterschiedlich aussehen können: Wird ein weiteres Dokument referenziert, kann dieses per *references* erfolgen, wird es selbst von anderen referenziert, geschieht das mit dem Feld *isReferencedBy*. Weitere Referenzen können in beide Richtungen über das Datenformat bestehen (*hasFormat* oder *isFormatOf*). Zudem kann ein Objekt in einem anderen enthalten sein oder umgekehrt (*hasPart* oder *isPartOf*) oder ein anderes komplett ersetzen beziehungsweise selbst ersetzt werden (*replaces* umgekehrt *isReplacedBy*). Interessant sind für Verkettungen, beispielsweise die in Kapitel 6 beschriebenen View-Paths, die Felder *requires* oder für die andere Richtung *isRequiredBy*. Normen- oder Standardkonformität lässt sich im Feld (*ConformsTo*) hinterlegen.
- *audience* dient der Klassifizierung der Zielgruppe des Objekts.
- *instructionalMethod* gehört zu den eher speziellen Feldern, die nur für bestimmte Objekte, in diesem Fall Lehr- und Lernmaterialien, sinnvoll Anwendung finden.

**Lebenszyklus** gibt das Erstellungsdatum (*created*) des Objekts oder einen anderen charakteristischen Zeitpunkt wie die Veröffentlichung (*issued*) an. Alternativ lassen sich Zeitspannen definieren, die am besten der Notation JJJJ-MM-TT<sup>16</sup> folgen. Je nach Objekt können andere relevante Daten eingesetzt werden: vorgelegt (*dateSubmitted*), von der Archivaufnahme akzeptiert (*dateAccepted*). Weiterhin kann "steht bereit von bis" (*available*) oder "in Kraft getreten am, gültig bis" (*valid*) oder auch "rechtlich geschützt" (*dateCopyrighted*) markiert werden. Zulässig und sinnvoll ist in vielen Fällen auch das Datum der letzten Änderung zu vermerken.

**Rechte und Personen** Viele der in Bibliotheken und Archiven eingestellten digitalen Objekte sind mit bestimmten Rechten versehen, die mit Personen oder Institutionen im Zusammenhang stehen. Es ist deutlich vorzuziehen, Rechte per Metadaten über das Archivsystem zu verwalten, als Rechte direkt im Objekt zu verankern.<sup>17</sup>

<sup>16</sup>Jahr, gefolgt von Monat und Tag nach DIN ISO 8601.

<sup>17</sup>Vgl. diesbezügliche Diskussion in Abschnitt 7.11.

- *creator* benennt die letzte Person oder Institution, welche am Objekt Veränderungen vorgenommen hat. Dieses ist der verantwortliche Verfasser oder Urheber, der durch weitere Personen unterstützt gewesen sein könnte (siehe dazu *source*, *contributor* und *publisher*).
- *contributor* enthält Namen je einer weiteren Person, welche an der Erstellung, Bearbeitung des Objekts beteiligt waren.
- *publisher* bezeichnet die veröffentlichenden Instanz. Dieses sind traditionellerweise Verleger oder Herausgeber.
- *rights* sammelt Hinweise und Informationen zu Rechten, die mit dem Objekt verknüpft und zu beachten sind. Dieses kann in Form einer Direktangabe oder auch als Verweis (URI) auf Lizenzbedingungen (*license*) erfolgen. Näher spezifiziert werden kann dieses durch *rightsHolder*. Hierbei handelt es sich um den Namen einer Person oder Institution, die Verwertungsrechte am Objekt hat oder Eigentümer ist. Mit dem Feld *accessRights* bezeichnet man den Sicherheitsstatus und welche Personen(gruppen) Zugriff darauf erhalten dürfen.
- *provenance* macht Angaben zur Echtheit, Authentizität des Dokuments. Diese können in Form von Signaturen oder Hinweisen zur Veränderungsgeschichte vorliegen.

Eine Anwendung von Dublin Core Metadaten beschreiben [Ward 2003]. Sie sind Bestandteil von Dokumenten im standardisierten OpenDocument-Format. Eine weitere Beispielanwendung ist RSS 1.0.<sup>18</sup> Ein weiteres Beispiel der Anwendung und Anlehnung an Dublin Core findet sich in [Gabriel und Ribeiro 2001]. Hier wird ein flexibles und erweiterbares Metadaten-schema für Archive, Bibliotheken und Museen speziell für Multimedia-Objekte, die aus mehreren Teilobjekten bestehen können, vorgeschlagen. In klassischen HTML-Seiten lassen sich Metadaten nach Dublin Core Schema mittels allgemeinen "meta"-Element im Kopf des Dokuments definieren.

Ein Beispiel für den Kopf einer Web-Seite:

```
<head profile="http://dublincore.org/documents/dcq-html/">
  <title>Dublin Core</title>
  <link rel="schema.DC"          href="http://purl.org/dc/elements/1.1/" />
  <link rel="schema.DCTERMS"     href="http://purl.org/dc/terms/" />
  <meta name="DC.format"         scheme="DCTERMS.IMT" content="text/html" />
  <meta name="DC.type"           scheme="DCTERMS.DCMIType" content="Text" />
  <meta name="DC.publisher"      content="Publisher" />
  <meta name="DC.subject"        content="Dublin Core Metadaten-Elemente" />
  <meta name="DC.creator" lang="en" content="Name of Document Author" />
  <meta name="DC.creator" lang="de" content="Name des Autors des Dokuments" />
  <meta name="DCTERMS.license"   scheme="DCTERMS.URI" \
    content="http://www.gnu.org/copyleft/fdl.html" />
  <meta name="DCTERMS.rightsHolder" content="Rechteinhaber" />
```

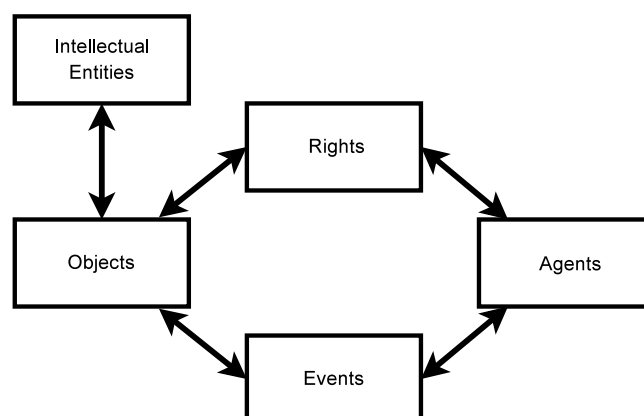
<sup>18</sup>Abkürzung für Really Simple Syndication – Konzept und Standard für die Verbreitung von Nachrichten im Internet.

```
<meta name="DCTERMS.modified" scheme="DCTERMS.W3CDTF" content="2008-01-18" />
</head>
```

Einen Vorschlag der Anwendung von Dublin Core zur Beschreibung von Computerspielen am Beispiel des Atari VCS 2500 und C64 macht [Huth 2004], S. 102 ff. in einer exemplarischen Studie.

### 3.3.2 PREMIS Metadata Set

Ein weiterer wichtiger Ansatz, der zunehmend an Bedeutung gewinnt, ist PREMIS [LoC 2008], ursprünglich initiiert von der US-amerikanischen Library of Congress. Am PREMIS-Projekt beteiligten sich Institutionen aus Australien, Neuseeland, den Vereinigten Staaten, Großbritannien, den Niederlanden und Deutschland. Ein zentrales Ziel bestand in der Identifizierung eines Kern-Sets von Metadaten sowie die Erstellung eines Data-Dictionary für diese Daten.



**Abbildung 3.5:** Gezeigt sind die Abhängigkeiten des grundlegenden PREMIS-Datenmodells nach [McCallum 2005].

PREMIS wurde bewusst einfach gehalten und verfolgt einen modularen Ansatz. Fünf Arten von Entitäten existieren im Modell: Objekte, Agents, Rechte, Ereignisse und die intellektuelle Entität selbst (Abb. 3.5). Inhaltlich beschreibende Metadaten eines Buch, einer Karte, Web-Site oder Ähnlichem, werden durch existierende Standards wie Dublin Core oder MODS (Metadata Object Description Standard) realisiert. Interessanterweise werden Rechte-Daten auf die Erlaubnis zu Erhaltungsmaßnahmen beschränkt, weil Zugriffs- oder Verbreitungsrechte für das Objekt nicht zum Kern der Archivierungsaktivitäten gerechnet werden. Beschränkungen gibt es ebenfalls für detaillierte technische Metadaten, Hardware- und Mediendokumentation, welche ebenfalls nicht eingeschlossen sind.

Eine weitergehende Ausführung zu Metadaten und ihren Formaten findet sich in den Kapiteln 5 und 6 in [Borhoff u. a. 2003]. Einen Überblick zu MARC 21 der Library of Congress gibt [McCallum 2005].

### 3.3.3 Formate für Metadaten

Bei der Ablage von Metadaten steht man vor einem ähnlichen Problem, wie beim Primärobjekt selbst - jedes Datenformat steht unter der Gefahr zu veralten. Da Metadaten zur späteren

Suche in Katalogsysteme eingespeist werden, liegt auf Kompatibilität und Interoperabilität ein wesentlicher Schwerpunkt. Die Autoren von [Borghoff u. a. 2003], S. 52 ff. schlagen als Format für Metadaten deshalb ASCII oder XML vor. Dublin Core Metadaten können zum Beispiel mit RDF/XML<sup>19</sup> dargestellt werden.

Ein Beispiel für ein XML-Schema für Metadaten ist der Metadata Encoding and Transmission Standard<sup>20</sup> (METS). Im März 2003 wurde die derzeit aktuelle Version 1.2 von der Library of Congress verabschiedet, die METS federführend verwaltet. Das XML-Format der Metadaten ist nicht durch METS festgelegt. Es kann je nach zugrunde liegendem Standard, wie beispielsweise Dublin Core, unterschiedlich sein. METS enthält Elemente zur Gruppierung von Objekten und ihre Verbindung mit deskriptiven und administrativen Metadaten.

METS eignet sich dazu, hierarchische oder anderweitig strukturierte Objekte zu kodieren, wobei es sich dabei um ein einzelnes Buch mit innerer Seiten- und Kapitelaufteilung, eine Buchreihe oder einen Film mit mehreren Sequenzen handeln kann. Zur Verknüpfung der digitalen Objekte mit ihren METS-Dateien findet eine Untermenge des XLink-Schemas Anwendung. Einige Anmerkungen zu METS und einem universellen Objektformat liefert [kopal 2008].<sup>21</sup>

## 3.4 Migration

Nach der Beschäftigung mit der allgemeinen Beschreibung und Einordnung von digitalen Objekten soll der weitere Verlauf des Kapitels die beiden zentralen Langzeitarchivierungsstrategien Migration und Emulation behandeln und sie am Ende miteinander vergleichen.

Migration im Kontext von Computern meint die Umwandlung eines digitalen Objekts in ein anderes unter Beibehaltung einer Reihe von Eigenschaften und unter Anwendung bestimmter Kriterien. Üblicherweise geht es hierbei um die Umwandlung von Daten 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 Daten zum jetzigen und möglichst auch für zukünftige Zeitpunkte sichern. Das Konzept der Migration von Computerdaten hat daher nicht allein mit digitaler Langzeitarchivierung zu tun. Allein schon 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 Migration oder spezieller Konvertierung im täglichen Umgang mit dem Computer durchaus gewohnt.

### 3.4.1 Begriffsbestimmung

Eine anerkannte Definition, die auch in [Borghoff u. a. 2003], Kapitel 3, benutzt wird, findet sich in [Hedstrom und Montgomery 1998]: *"Migration is the periodic transfer of digital materials from one hardware/software configuration to another, or from one generation of computer technology to a subsequent generation. The purpose of migration is to preserve*

<sup>19</sup>Siehe dazu - <http://ilrt.org/discovery/2001/09/rdf-xml-schema>.

<sup>20</sup>Vgl. offizielle Homepage unter <http://www.loc.gov/standards/mets>.

<sup>21</sup>Siehe hierzu - [http://kopal.langzeitarchivierung.de/downloads/kopal\\_Universelles\\_Objektformat.pdf](http://kopal.langzeitarchivierung.de/downloads/kopal_Universelles_Objektformat.pdf).

*the integrity of digital objects and to retain the ability for clients to retrieve, display, and otherwise use them in the face of constantly changing technology. Migration includes refreshing as a means of digital preservation but differs from it in the sense that it is not always possible to make an exact digital copy or replica of a database or other information object as hardware and software change and still maintain the compatibility of the object with the new generation of technology.*" Ebenso bietet [Thibodeau 2002] im Kapitel zur Migration Erklärungen an. Migration scheint auf den ersten Blick die naheliegendste und "natürlichste" Langzeitverfügbarkeitsstrategie für digitale Objekte zu sein. Eine gute Gesamtdarstellung für statische digitale Objekte bietet [Borghoff u. a. 2003], S. 59 ff.

Die Formatumwandlung kann dabei sowohl horizontal als auch vertikal bezogen auf eine gegebene Zeitebene stattfinden (Abb. 3.6).

**Horizontale Migration** Mit dem Konzept der horizontalen Migration hat jeder Computerbenutzer vermutlich schon einmal zu tun gehabt. In regelmäßigen Abständen bringen Softwareproduzenten neue Versionen ihrer Applikationen heraus. Ein Benutzer, der diese Produkte für seine tägliche Arbeit einsetzt, ist meistens dazu gezwungen, die Entwicklungen nachzuvollziehen. Da mit jeder neuen Version üblicherweise der Software neue Fähigkeiten<sup>22</sup> 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. Umgekehrt steht der Benutzer oft vor einem Problem: Die neuen Objekte lassen sich oft nicht mehr oder nur unter signifikanten Einbußen mit den älteren Versionen derselben Software öffnen.

Dabei gehen Benutzer oft implizit von der Annahme aus, dass die neuen Formate mindestens alle notwendigen Eigenschaften der früheren Formate besitzen und zusätzlich vielleicht weitere Vorteile bieten, wie beispielsweise Kompression.<sup>23</sup>

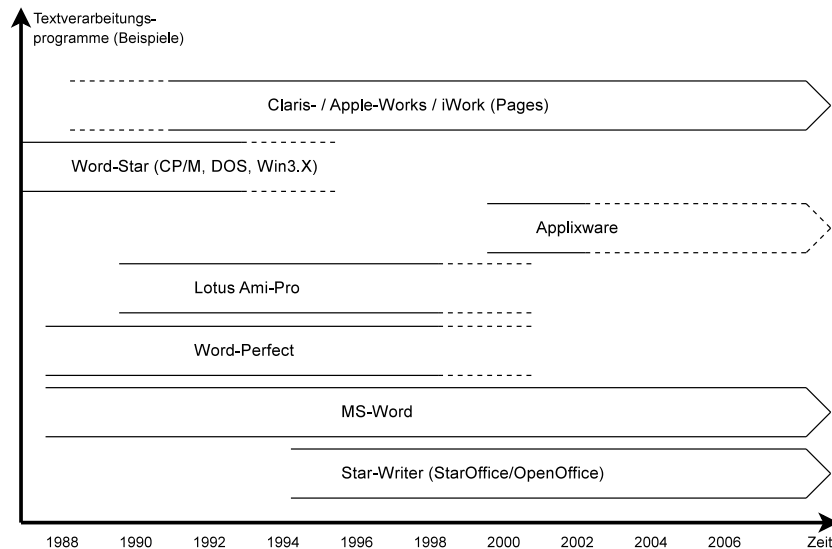
**Vertikale Migration** Vor ähnlichen Problemen stehen Nutzer verschiedener Softwareprodukte, die den gleichen Zweck erfüllen. Selten steht nur ein Produkt zur Verfügung. Anwendungen mit ähnlicher Funktionalität lassen sich nach ihrem Einsatzzweck klassifizieren, wobei sich Produkte in ihrer Funktionalität und in ihrem Bedienungsmodell durchaus erheblich unterscheiden können. Ein Beispiel wären die verschiedenen Applikationen zur Erstellung von Bildschirmpräsentationen, wie Microsoft Powerpoint, StarOffice Impress oder Presentation von Apple.

**Migrationsebenen** Das bisher Ausgeführte zeigt, dass es nicht eine einzige Migrationsstrategie gibt, sondern Migration eher als generelles Konzept, welches einen Korb von Strategien beinhaltet, verstanden werden muss. 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,

<sup>22</sup> Vgl. hierzu Abschnitt 2.3.2.

<sup>23</sup> Textverarbeitungsprogramme wie StarOffice oder OpenOffice, die ihre Daten XML-formatiert ablegen, sind dazu übergegangen ihre Daten aus Gründen der Platzeffizienz GZIP zu komprimieren.



**Abbildung 3.6:** Horizontale und vertikale Migration am Beispiel einiger Textverarbeitungsprogramme.

- der Transformation zwischen Datenformaten und
- der Wahl von und Migration zwischen Datenträgern.

### 3.4.2 Migrationsstrategien für Datenformate

Vielfach kommt man bei der Migration von Daten nicht um eine Änderung ihrer logischen Struktur herum. Das ist besonders dann der Fall, wenn der Einsatz einer neuen Software oder die Umstellung auf ein alternatives Softwareprodukt ansteht.

Üblicherweise stellt der Softwarehersteller eines Produktes bei einem Upgrade eine Importfunktion für die Vorgängerversionen zur Verfügung. Jedoch bezieht sich diese Funktionalität oft nur auf die direkten Vorgänger- oder bedingt Vorvorgängerversionen. Wenn weiter zurückliegende Versionen desselben Produktes nicht mehr unterstützt werden, bleibt für die Sicherung des langfristigen Zugriffs, ein Upgrade aller betroffenen Objekte nicht aus. Diese müssen dann unter Umständen erst auf die vorherige und anschließend die jeweils aktuelle Version transformiert werden.

Ein typisches Beispiel sind alte Textdokumente, die mit Microsoft Word erstellt wurden. So versagen oft die Import-Filter der aktuellen Produkte Word 2000 oder 2003 beim Einlesen von Dokumenten, die vor Word 97 erstellt wurden.

Dieses bringt wesentliche Implikationen mit sich: Es muss immer der gesamte Datenbestand, der mit einer Software erstellt wurde, auf das neue Release übertragen werden. Das bedeutet, dass der Umfang der zu migrierenden Daten mit jeder neuen Version zunimmt oder dass in regelmäßigen Abständen entschieden werden muss, welche Daten aus dem Prozess ausscheiden.

Bei einem Produktwechsel kann dann der Anwender darauf hoffen, dass der Hersteller des alten Produkts eine Reihe von Exportfiltern in Fremdformate vorgesehen hat. Da jedoch ein Hersteller wenig Interesse daran hat, den Wechsel zur Konkurrenz aktiv zu unterstützen,



wird man selten ein Format finden, welches alle Funktionen abbildet. Stattdessen wird man beispielsweise bei Textdokumenten auf einen gewissen gemeinsamen Nenner zurückgehen. Dieser gemeinsame Nenner basiert oft auf offenen Standards wie unformatierter ASCII oder UTF Text, formatiertem RTF, HTML oder XML.<sup>24</sup>

Wird jedoch die Dokumentenstruktur komplexer und besteht sie aus vielen Einzelkomponenten, so ist das Verhältnis dieser Komponenten untereinander von Interesse. Hier kommen oft sogenannte Verweise zum Einsatz, deren Ziele, auf die sie zeigen, nicht zwangsläufig persistent sein müssen.

Umgekehrt kann ein Produktwechsel auch aus Sicht eines Imports von Fremdformaten betrachtet werden. Das Problem stellt sich ähnlich, wenn es auch für den Produktanbieter attraktiv erscheint, gute Importfilter von Konkurrenzprodukten zur Verfügung zu stellen. Hier hebt sich die OpenOffice bzw. die StarOffice Suite positiv aus dem Feld heraus: Sie erlauben den Import der meisten Microsoft Word Formate und unterstützen dabei auch die sehr alten Versionen.

Gerade bei der Migration von Softwareprodukten, die das Ende ihres Produktzyklus erreicht haben, ist oft nur eine verlustbehaftete Migration möglich. Viel zitierte Beispiele sind der Formatverlust in Dokumenten. Wurde beispielsweise das Periodensystem der Elemente oder ein Gedicht mit einem bestimmten Zeichensatz erstellt, so kann die Struktur und damit oft der Sinn bei der Konversion verloren gehen. Ein anderes Beispiel sind "Zeichnungen" in ASCII-Grafik, wie sie oft in älteren Dokumentationen oder Lehrbüchern zum Einsatz kamen.

Die Migrationsproblematik hat eine ganz eigene Klasse von Softwareprodukten entstehen lassen. So gibt es gerade im Multimedia-Bereich eine große Auswahl an Programmen, die es erlauben, diverse Ausgangsformate in diverse Zielformate umzuwandeln. Oft wird dabei zu einer Vereinfachung gegriffen, um nicht die komplette Wandlungsmatrix von  $i$  Ausgangsformaten in  $j$  Zielformate abbilden zu müssen. Es wird ein leistungsfähiges möglichst verlustfreies Zwischenformat gewählt in welches zuerst das Ausgangsformat umgewandelt wird, bevor dann das Zwischenformat auf das Zielformat gewandelt wird. Das impliziert jedoch eine doppelte Migration.<sup>25</sup>

### 3.4.3 Regeneration und Migration von Datenträgern

Bisher wurde Migration unter dem Aspekt des Datenaustausches und der Datenkonversion zwischen verschiedenen Softwareprodukten betrachtet. Ein Bereich in dem Migrationsstrategien aufgehen oder sich kaum vermeiden lassen, sind Datenträger. Angaben zum Verfall lassen sich häufig schwer treffen, da sie vielfach weniger als ihre angegebene maximale Lebensdauer am Markt sind. Generell sollte man Herstellerangaben<sup>26</sup> nicht blind vertrauen. Die prognostizierte Lebensdauer liegt weit unter der von Papier oder Microfiches. Das liegt an der technologischen Entwicklung, welche immer neue Datenträger mit immer größeren Speicherkapazitäten und schnelleren Zugriffsverfahren auf den Markt bringt. Die Unterstützung

<sup>24</sup>Eine Beschreibung der genannten Datenformate findet sich beispielsweise bei PRONOM [TNA 2007].

<sup>25</sup>Ein Beispiel für diese Herangehensweise ist die PNM-Bibliothek unter Linux/Unix, welche vielfältige Umwandlungen von Grafikformaten ineinander erlaubt.

<sup>26</sup>Ein Beispiel ist die Mean Time between Failures (MTBF) für Festplatten. Diese Angabe bezeichnet einen statistischen Durchschnittswert für die Betriebsdauer eines Bauteils oder Gerätes, während dessen es störungsfrei funktioniert.

älterer Systeme, besonders derer, die sich am Markt nicht durchsetzen konnten und nur ein Nischendasein fristeten, wird oft nach wenigen Jahren eingestellt.

Die Erfahrungen zur langfristigen Haltbarkeit bestimmter Datenträger müssen vielfach erst noch gemacht werden, da die Technologien oft noch so neu sind, dass aufgrund bisher vergangenen zu kurzen Zeitspanne noch keine belastbaren Aussagen getroffen werden können. Dabei kommt es nicht nur auf den Datenträger selbst an, sondern auch auf ihn begleitende Systeme um die Daten auslesen zu können. Deshalb geht Rothenberg in [Rothenberg 1999] etwas provokativ aber durchaus begründet von einer mittleren technologischen Haltbarkeit von fünf Jahren aus.

Deshalb besteht eine häufige Strategie darin, weit vor dem angegebenen Haltbarkeitsdatum den Inhalt von Datenträgern umzukopieren. Diese Aufgabe übernimmt das Archivmanagement. Hierbei kann es sich durchaus um zeitliche Herausforderungen handeln, wenn ein Archiv eines bestimmten Umfangs umkopiert wird.

### 3.4.4 Probleme der Datentransformation

Generell sollte man festhalten, dass Migrationsprozesse, die Datentransformationen beinhalten, fast generell unter dem Verdacht der Verlustbehaftung<sup>27</sup> stehen. Finden mehrere Transformationsprozesse nacheinander statt, können sich Informationsverluste aufsummieren. Eine gute Beschreibung dieser Problematik und einen möglichen Lösungsvorschlag liefert [Mellor u. a. 2002].

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 Anbieter 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, der Pleite 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. Diese 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

<sup>27</sup>Es gibt Versuche diese Prozesse zu beschreiben und durch Bestimmung des Dateiinhalts und anschließenden Vergleich zu verifizieren, siehe hierzu die Forschungen an der Universität Köln <http://www.portal.uni-koeln.de/1224.html>.

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 sparsameren Transport zum Endanwender erlauben.

**Aufwand** In Anbetracht der beständig anwachsenden Datenmenge könnte eine Reduktion des Gesamtdatenvolumens angeraten erscheinen. Ein Aussortieren scheinbar obsoleter digitaler Objekte sollte weitgehend vermieden werden, da Vorhersagen, welches Objekt später noch einmal benötigt wird, sehr schwer zu treffen sind. Zudem könnte sich das Interesse nachfolgender Generationen komplett ändern, was ebenso schlecht vorausgesehen werden kann.

Der Aufwand von Transformationen ist damit nicht zu unterschätzen: Unabhängig von der aktuell vermuteten Relevanz bestimmter Objekte, ist immer der komplette Datenbestand 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 Zeiträumen erstrecken. Planungen solcher Prozesse sind Bestandteil der OAIS Management-Komponente.

**Technologischer Wandel** 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. Einen Überblick für Nichttechniker bietet [Rohde-Enslin 2004].

Wenn zur Reduktion der Datenmenge Kompressionsalgorithmen eingesetzt werden, sind in den meisten Fällen nur verlustfreie Verfahren zulässig. Hier muss jedoch sichergestellt sein, dass zu jedem Zeitpunkt Programme existieren, die vorgenommene Kompression wieder rückgängig machen können. Das bedeutet jedoch, dass neben den genannten Problemen eine weitere Ebene eingezogen wird. Im Laufe der Zeit gibt es immer wieder Verbesserungen der Algorithmen. Deshalb ist das LHA-Verfahren beispielsweise eher als historisch anzusehen, RAR- oder BZIP2-Archive<sup>28</sup> hingegen noch recht jung.

Digitalisierungs- und Kompressionsverfahren, welche mit Qualitätsverlust des Ergebnisses arbeiten, kommen vielfach bei Programmen zur Bearbeitung statischer Bilder, Audio- oder Videodaten zum Einsatz. Hier muss man verhindern, dass die beteiligten Applikationen die Daten in der verschlechterten Qualität ablegen oder muss dafür sorgen, dass das verlustfreie Original erhalten bleibt. Damit wäre aber das eventuell angestrebte Ziel einer Datenreduktion im Archiv ins Gegenteil verkehrt.

---

<sup>28</sup>Einige Quellen, wie [nestor 2008] raten dazu, Kompression generell zu vermeiden, da bei Datenträgerfehlern die Wahrscheinlichkeit von Totalverlusten steigt.

### 3.5 Emulation

Gegeben durch den Charakter der meisten bisher in Betracht gezogenen 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. Letztere stellt jedoch ein zentrales Moment vertrauenswürdiger Langzeitarchive dar.

Emulation setzt dabei nicht am digitalen Objekt selbst an, sondern beschäftigt 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 [Rothenberg 1995] und erweitert in [Rothenberg 1999]. Kompakte Überblicke geben zudem [Coy 2006], S. 58 ff. und [Borghoff u. a. 2003], Kapitel 4.

Computerhardware kann demnach durch Software nachgebildet werden, auch wenn dieses erstmal deutlich komplexer erscheint. So führt [Thibodeau 2002], S. 19 aus: *"Various approaches can be used to simplify the problem while still keeping data in their original encodings. The best-known approach is emulation. Emulation uses a special type of software, called an emulator, to translate instructions from original software to execute on new platforms. The old software is said to run "in emulation" on newer platforms. This method attempts to simplify digital preservation by eliminating the need to keep old hardware working. Emulators could work at different levels. They could be designed to translate application software to run on new operating systems, or they ..."*

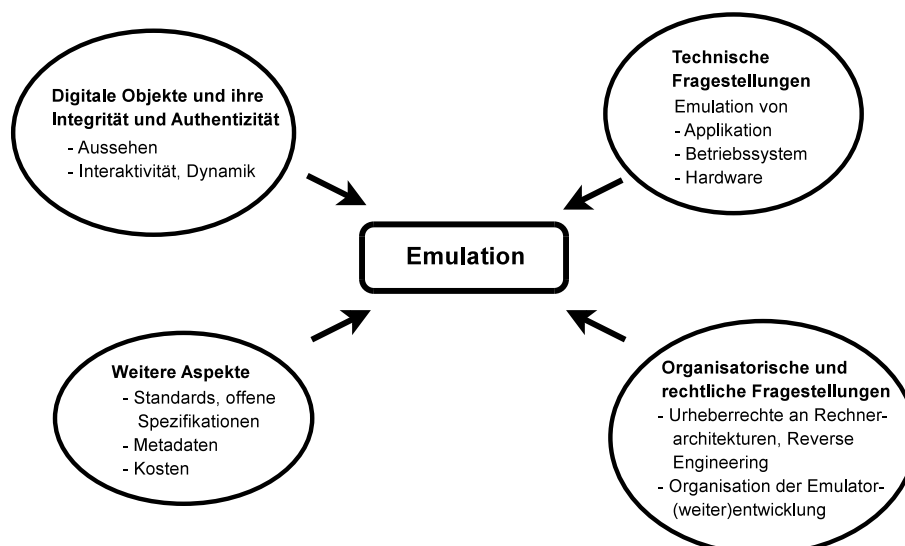


Abbildung 3.7: Fragestellungen beim Einsatz von Emulation nach [Granger 2000]

Emulation als Konzept ist nicht neu:

- Apple hat seinen Benutzern den Umstieg von Mac-OS 9 auf Mac-OS X durch einen

Emulator mit dem Namen "classic" erleichtert: Dadurch konnten die meisten alten Programme weiterhin unter dem neuen System ausgeführt werden.

- In einen ähnlichen Bereich fällt die DOS-Umgebung des Windows NT. Hier wird eine Umgebung nachgebildet, die die Ausführung älterer DOS-Programme erlaubt. Dazu werden sogar die populären Soundblaster-Hardwareschnittstellen emuliert. Die Ausführbarkeit gilt für neuere Microsoft Betriebssysteme nicht mehr uneingeschränkt: Viele alte Applikationen, die mit direkten Hardwarezugriffen arbeiten, können nicht mehr ausgeführt werden.
- Das Programm "Pose" emuliert beispielsweise die Palm-III-Hardware auf einer Linux X11-Plattform.<sup>29</sup> Das erlaubte Applikationsentwicklern ihre Software bequem in ihrer gewohnten Entwicklungsumgebung zu erstellen und zu testen und dadurch die Prozesse zu beschleunigen.
- Spielekonsolenhersteller steigern die Attraktivität ihrer neuen Generation von Konsolen, indem sie Titel der Vorgängergeneration weiter ablaufen lassen können. Da sich jedoch üblicherweise die Hardware wesentlich ändert, wie der Übergang von der Intel-Architektur der Xbox auf den PowerPC der Xbox360,<sup>30</sup> kommt Emulation zum Einsatz.

Emulatoren dienen der Erhaltung beziehungsweise Wiederherstellung alter Ablaufumgebungen. Dadurch erfüllen sie eine Verbindungsaufgabe zwischen der jeweils aktuellen Rechnerplattform oder einer Zwischenstufe und der zum Entstehungsmoment des Objekts gültigen Erstellungs- oder Nutzungsumgebung.

Bei *Emulatoren* handelt es sich um Computerprogramme, die in der Lage sind Ablaufumgebungen, wie Applikationen, Betriebssysteme oder Hardwareplattformen, in geeigneter Weise in Software nachzubilden, so dass ursprünglich für diese Ablaufumgebung erstellte Objekte weitgehend so arbeiten oder aussehen, wie in der Originalplattform oder -applikation. Emulatoren bilden damit die Schnittstelle, eine Art Brückenfunktion, 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.

**Emulation versus Virtualisierung** Eng verknüpft mit dem Konzept der Emulation ist die Virtualisierung. In den letzten fünf Jahren hat sich der Begriff Virtualisierung stark verbreitet. Er wird dabei oft mit dem Begriff der Emulation vermischt, da viele Virtualisierungsprodukte beide Merkmale aufweisen. Virtualisierung gewann ihre Bedeutung nicht als Strategie zur Langzeitarchivierung, sondern zur Verbesserung der Hardwareauslastung, Erhöhung der Softwaresicherheit und der Vereinfachung in Tests und Entwicklung.

Das Verfahren, mehrere verschiedene Betriebssysteme oder Instanzen auf einer einzigen Hardware parallel auszuführen, ist seit über 30 Jahren von der Mainframe-Architektur bekannt. Es ermöglicht eine effektive Ressourcennutzung und -verteilung zwischen vielen gleichzeitigen Anwendern. Inzwischen hält dieses Verfahren auch in die PC-Architektur Einzug.

<sup>29</sup>Siehe beispielsweise <http://packages.debian.org/source/etch/pose>. Der Emulator erhielt sogar einen passenden Rahmen im Look der damaligen Hardware.

<sup>30</sup>Die Sony Spielekonsole "PlayStation" gibt es inzwischen in der dritten Generation: In der ersten Hardwareversion der Konsole wurde die Rückwärtskompatibilität noch durch eine Hardwarelösung hergestellt, in der in Europa verkauften Variante und folgenden Revisionen setzt der Hersteller komplett auf Softwareemulation.

Emulation und Virtualisierung schaffen die Möglichkeit einer virtuellen Umgebung auf einer gegebenen Host-Plattform. Während jedoch die Emulation einer kompletten Umsetzung aller Hardwarekomponenten in Software realisiert, arbeitet die Virtualisierung stärker mit der zugrundeliegenden Host-Plattform zusammen. So werden beispielsweise zur Vermeidung von Overhead und damit einhergehendem Performance-Verlust CPU-Instruktionen direkt aus der virtualisierten Umgebung durchgereicht. Dadurch ist eine Verwendung eines bestimmten Virtualisierungssystems auf einer anderen Architektur ausgeschlossen.

Emulatoren können hingegen von einer Host-Architektur auf eine andere übertragen werden. Die Neuübersetzung des Emulators selbst muss natürlich trotzdem erfolgen.

Virtualisierung wird vielfach dazu eingesetzt, die Systemsicherheit durch Kapselung zu erhöhen und die Schnittstellen zu Hardwarekomponenten zu vereinheitlichen. Gerade letzteres kann wiederum für die Emulation in der Langzeitarchivierung von Interesse sein.

Eine Untermenge, die sogenannte "Paravirtualisierung", wurde in den letzten Jahren populär. Dieser Ansatz verhält sich sehr ähnlich zur klassischen Virtualisierungssoftware. Er versucht, statt "gefährliche" CPU-Instruktionen abzufangen und diese zu emulieren, direkt die Gastbetriebssysteme zu modifizieren, so dass diese direkt den Virtualisierer zur Ausführung aufrufen. Beispiele hierfür sind die Linux Konzepte UML (User Mode Linux) oder das mit viel Aufmerksamkeit bedachte XEN.<sup>31</sup>

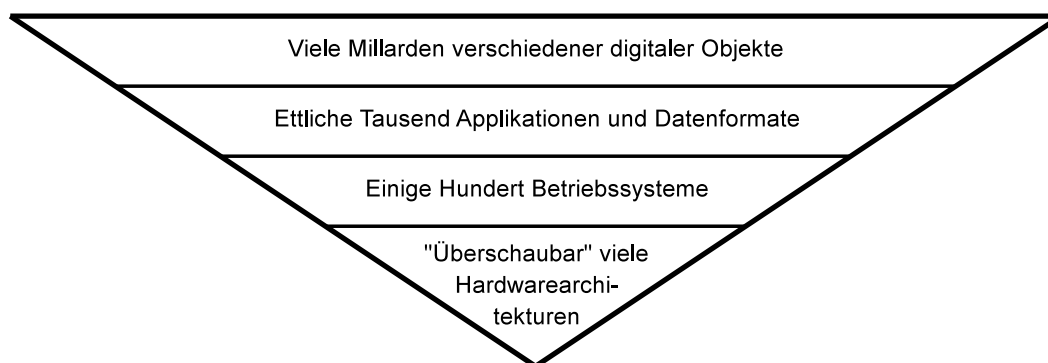
**Unvollständige Emulation** Ein wesentlicher Nachteil der Virtualisierung besteht in der Unsicherheit durch Obsoleszenz von Prozessorarchitekturen. Da aus Performancegründen CPU-Instruktionen direkt an die Hardware durch den Virtualisierer hindurchgereicht werden, sind diese beim Wechsel der darunterliegenden Hardwareschicht nicht mehr ausführbar. Trotzdem kann durch Virtualisierung recht effizient ein Teil der auftretenden Langzeitarchivierungsprobleme gelöst werden, wie in späteren Kapiteln gezeigt und im Anhang demonstriert wird. Hinzu kommt, dass komplette Hardwareemulatoren wegen der vollständigen CPU-Nachbildung komplexer und damit fehleranfälliger sind.

**Simulation** Teilweise taucht im Zusammenhang mit Emulation der Begriff der Simulation auf. Simulation versucht bestimmte physikalische Vorkommnisse oder Aktionen, wie das Klima der Erde oder die Kollision eines Fahrzeuges mit einem Hindernis, nachzubilden. Dabei versucht man herauszubekommen, wie sich das Klima unter bestimmten Annahmen verhält oder das Fahrzeug verformt, ohne es dabei tatsächlich zu tun. Emulation macht hingegen genau das, was auch in der physikalischen Welt passieren würde: Es wird ein Abbild eines Computers erzeugt und benutzt. Siehe hierzu auch [Verdegem und van der Hoeven 2006], S. 2. Wenn man einen Computer, beispielsweise ein mehr als zwanzig Jahre altes Gerät, wie den Commodore 64, als physikalische Gesamteinheit sieht, kann auch der Begriff der Simulation geeignet erscheinen. Deshalb taucht gerade bei Home-Computer-Emulatoren häufiger auch die Bezeichnung "Simulator" auf.

<sup>31</sup>Siehe hierzu [Barham u. a. 2003] oder auch [Thorns 2007], Kapitel 15.

### 3.5.1 Ansatzpunkte

Emulation heißt zuerst einfach erstmal nur die Erschaffung einer virtuellen Umgebung in einer gegebenen Ablaufumgebung. Das kann bedeuten, dass Software durch eine andere Software nachgebildet wird, ebenso wie Hardware in Software. Generell lässt sich feststellen, dass



**Abbildung 3.8:** Zahl der zu betrachtenden und behandelnden digitalen Objekte je nach gewählter Ebene des Emulationsansatzes.

für aktuelle Computerplattformen üblicherweise drei Ebenen für den Ansatz von Emulation identifiziert werden können, eine Einführung und einen Überblick hierzu geben [Granger 2000] oder der Abschnitt "Emulation levels" in [Verdegem und van der Hoeven 2006]:

- Emulation oder Ersatz von Applikationen durch andere, um so bestimmte Datenformate darzustellen oder auszuführen
- Emulation von Betriebssystemen, um so auf einem gegebenen Betriebssystem Applikationen auszuführen, die die Schnittstellen (API) eines anderen Betriebssystems erwarten
- Nachbildung eines komplettes Rechners einer bestimmten Architektur.

In die generellen Überlegungen sollte die Anzahl der jeweils zu betrachtenden Objekte einfließen (Abb. 3.8).

### 3.5.2 Nachbildung von Applikationen oder ihrer Funktionen

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 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. 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 lässt sich auf weitere Applikationen übertragen. Generell bedeutet das die Verwendung einer anderen Applikation als "Betrachter" als das ursprüngliche Programm – der *Editor*, mit dem die Datei erstellt wurde.

Aus Sicht der Langzeitarchivierung genügt üblicherweise ein Betrachter, der in der jeweils aktuellen Ablaufumgebung 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.

**Klasse der Anwendungsprogramme** Es existieren beispielsweise recht gute Importfilter für alte und sehr alte Microsoft Office Objekte für die aktuellen Star- oder OpenOffice Programme, die es erlauben, auch sehr frühe Texte, die mit einer der ersten Versionen von Word für Windows erstellt wurden, wieder einzulesen. Ebenso beherrschen diese Programme den Import der aktuellen Dokumente. Hier erkennen Anwender jedoch häufig schon die Grenzen der Prozedur: Nicht alle Eigenschaften werden zu 100% korrekt übertragen und geeignet dargestellt.

Noch schwieriger ist die Lage bei dynamischen Bestandteilen: Wurde das Textdokument mit einem oder mehreren Makros<sup>32</sup> versehen, so schlägt der Import an dieser Stelle oft fehl, da diese Funktion schlicht fehlt oder komplett anders implementiert wurde. Das liegt besonders daran, dass Datenformate proprietär sind. Damit kann es passieren, dass Teile der Datenformate nicht bekannt oder nur unvollständig verstanden sind.

**Spiele und Unterhaltung** Ein etwas anders gelagertes Feld sind Computerspiele. Viele dieser Applikationen schreiben zwar auch Dateien in Form von Zwischen- oder Spielständen. Der Informationsgehalt oder das Interesse diesen digitalen Objekten gegenüber ist sicherlich zu vernachlässigen. Hier geht es um die Applikation selbst. Dabei ist zu beobachten, dass sehr populäre Spiele der alten Spiele-Konsolen und Home-Computer der 1980er Jahre, wie beispielsweise Pac-Man für aktuelle Plattformen nachprogrammiert wurden. Hier haben sich die Programmierer am Original orientiert und versucht, dieses weitestgehend nachzuempfinden. Dieses gilt ebenso für Klassiker der PC-Spiele. So gibt es Umsetzungen, die sich von Microsofts "Civilization"<sup>33</sup> oder "Warlords"<sup>34</sup> inspirieren ließen. Auch wenn diese Titel nun oft unter einer Open-Source-Lizenz quelloffen zur Verfügung stehen, ist damit nicht automatisch deren Erhaltung über sehr lange Zeiträume sichergestellt.

<sup>32</sup>Makros sind Zusammenfassungen von ganzen Befehls- oder Handlungsfolgen, die unter einem einzigen einfach aufzurufenden Stichwort zusammengefasst werden. Makros benutzen dazu oft eingebettete Skriptsprachen der Applikation oder des Betriebssystems. Sie erfordern deshalb üblicherweise die Existenz spezieller Interpreter.

<sup>33</sup>FreeCiv Homepage - <http://www.freeciv.org>.

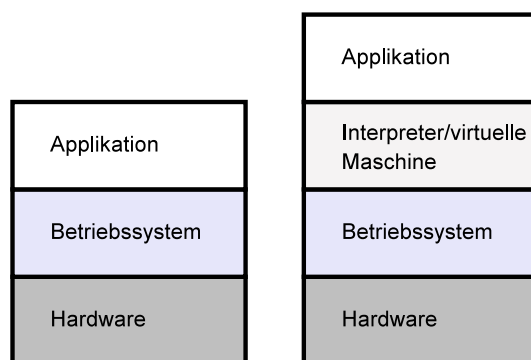
<sup>34</sup>Homepage des freien Clones - <http://freelords.sourceforge.net>. Im Augenblick verfügt das Projekt über eine sehr aktive Entwicklergemeinschaft, die das Spiel auch auf Java portiert hat.



Was für einfache oder sehr populäre Spiele noch funktioniert hat, wird mit den zunehmend komplexer werdenden Spielen nicht mehr möglich sein. Zudem ist im Laufe der Zeit die Zahl der Titel<sup>35</sup> nachgerade explodiert.

Eine recht interessante Möglichkeit, die zumindest für Klassen ähnlich angelegter Produkte gelten kann, ergibt sich aus der Trennung von Spiele-Engine<sup>36</sup> und Datenpaketen. So basiert die Serie von "Indiana-Jones"-Titeln aus dem Hause Lucas Arts auf einem gemeinsamen Ansatz. Hierfür wurde eine Art Emulator entwickelt, der die alten Spieledaten lesen und für aktuelle Benutzer-Interfaces darstellen kann. Auf diese Weise lassen sich über zehn Jahre alte Titel auf aktuellen Betriebssystemen und Rechnerarchitekturen ausführen.

**Maschinenunabhängige Byteformate oder Skriptsprachen** Versuche den Endanwender von inkompatiblen Software- und Hardwareschnittstellen abzuschirmen, existieren schon recht lange. Hierzu zählt die Einführung von Zwischenebenen als zusätzliche Abstraktionsschicht in Form von Skriptsprachen, wie Python, Ruby oder Perl (Abb. 3.9). Ähnliches findet man im Bereich des Web-Publishings<sup>37</sup> mit Actionscript für Flash oder Silverlight. Diese



**Abbildung 3.9:** Übliche Hardware- und Softwareschichten moderner Rechnerplattformen. Eine besondere Rolle spielen die maschinenunabhängigen Skriptsprachen und Byteformate.

Sprachen führen Anweisungen mittels eines Interpreters aus, der für verschiedene Rechnerplattformen übersetzt wurde. Ein weitergehenden Ansatz, der die Vereinfachung der Softwareentwicklung, Portierbarkeit und Performance miteinander verbinden soll, verfolgt die Programmiersprache Java. Der Java-Compiler erzeugt einen Byte-Code, der in der sogenannten Java Virtual Machine verschiedener Plattformen ausgeführt werden kann.

**Bewertung** Datenformate sind von der darunterliegenden Hardware-Architektur weitgehend unabhängig, so dass die Portierung der zur Betrachtung oder Ausführung angewendeten Programme bei Architekturwechseln ausreicht. Dieses setzt jedoch bei bestehenden Applikationen den Zugriff auf den Quellcode oder die Bereitschaft der Herstellerfirma<sup>38</sup> voraus.

<sup>35</sup>Schätzungen von Wirtschaftsagenturen gehen davon aus, dass in den nächsten Jahren der Umsatz mit Computerspielen den Umsatz in der Filmindustrie übersteigt [HZV 2007].

<sup>36</sup>Rendern der "Spielewelt" anhand gelieferter Daten, siehe hierzu den Abschnitt A.1.4 im Anhang.

<sup>37</sup>Diese Datenformate und Anwendungen sind besonders beim Web-Archiving zu berücksichtigen.

<sup>38</sup>Diese muss hierzu noch selbst existieren oder zumindest in einem anderen Unternehmen aufgegangen sein. Andernfalls kann es schwierig bis unmöglich werden, an den Quellcode von Software heranzukommen, da es keine vergleichbare gesetzliche Regelung, wie für Bücher, gibt.

Bei der Emulation älterer Datenformate ist man darauf angewiesen, dass jeder Softwarehersteller sämtliche seiner Formate auch sehr alter Versionen unterstützt oder diese so dokumentiert, dass sie von Dritten interpretiert werden können. Die bisherigen Erfahrungen geben wenig Anlass zur Hoffnung. Ein besonderes Problem ist dabei das Ausscheiden von Marktteilnehmern, wie dieses bei der Office-Suite der Firma Lotus zu beobachten war. Dieses Verfahren wird sich daher nur auf statische, gut dokumentierte Standardformate erfolgreich anwenden lassen.

Die Anwendung dieses Emulations-Levels bleibt trotzdem auf eine bestimmte Menge wohldefinierter, möglichst freier und offener Datenformate beschränkt. Die generelle Portierung und Emulation von Applikationen wird aufgrund der Fülle von Programmen, der oft fehlenden Information und Dokumentation und der Komplexität der Aufgabe nicht möglich sein.

### 3.5.3 Emulation des Betriebssystems oder dessen Schnittstellen

Eine Ebene darunter (Abb. 3.9) setzt die Emulation von Betriebssystemen beziehungsweise deren erwarteten Schnittstellen<sup>39</sup> an. Dieser Ansatz fußt auf der Idee, dass sich Programme beispielsweise unter Linux/X11 ausführen lassen, die ursprünglich für Microsoft Windows geschrieben wurden. Die Motivation liegt nicht primär in der Langzeitarchivierung begründet, sondern in dem Versuch, dem Zwang auszuweichen, für eine gewünschte Applikation gleich ein anderes Betriebssystem verwenden zu müssen. Besonders hervorheben kann man an dieser Stelle das Wine-Projekt<sup>40</sup> und seine kommerziellen Ableger, wie CodeWeavers. Das Projekt ReactOS geht noch weiter und programmiert das komplette Windows XP nach. Umgekehrt schafft Cygwin Posix-Schnittstellen für Windows.<sup>41</sup>

Etwas anders gelagert sind die Bemühungen der Betriebssystemhersteller, ihren neuen Systemen Schnittstellen für alte Software zur Verfügung zu stellen. Sie bieten so ihren Benutzern die Freiheit, auf ein neues Betriebssystem zu wechseln, ohne dafür sofort alle Applikationen austauschen zu müssen. Hierfür findet man oft den Begriff der "Soft Migration". Beispiele sind die im vorangegangenen Abschnitt genannten Emulationen von Betriebssystemschnittstellen seitens Microsofts und Apples.

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 gangbar. Da gerade im Bereich kommerzieller, proprietärer Software viele Schnittstellen nicht oder nur wenig dokumentiert sind, ist die Gefahr einer Inkompatibilität groß.<sup>42</sup>

Eine für die Langzeitarchivierung weniger interessante Variante findet sich mit einer kom-

<sup>39</sup>Die sogenannten Application Programming Interfaces - API. Einen generellen Überblick zu Betriebssystemen liefert [Stallings 1998].

<sup>40</sup>Vgl. hierzu die Projekt-Homepage - <http://www.winehq.org>. Der Windows-API-Emulator befindet sich in aktiver Entwicklung und erfreut sich einer großen Anwendergemeinde. Eine kommerzielle Erweiterung ist CodeWeavers - <http://www.codeweavers.com>.

<sup>41</sup>Vgl. zu diesen Projekten Abschnitt A.1.4.

<sup>42</sup>Programmierschnittstellen oder APIs werden geändert, es gibt Sicherheitsupdates und die Umstellung von Konzepten.

pletten Betriebssystemvirtualisierung. Hier setzen alle höheren Betriebssystemfunktionen auf einem gemeinsamen Kernel auf. Beispiele hierfür sind OpenVZ oder Linux VServer.<sup>43</sup>

Während es Projekten wie Wine zwar gelingt, etliche für die Microsoft-API erhältlichen Programme für Linux/X11 nutzbar zu machen, so ist hiermit im Sinne einer Langzeitverfügbarkeit wenig erreicht. Zwar schaut das Wine-Projekt schon auf eine vieljährige Entwicklungsgeschichte zurück, ist aber trotzdem nicht in der Lage, eine 100-prozentige Abbildung aller Schnittstellen zu leisten. Die Liste der unterstützten Applikationen liest sich durchaus beeindruckend, trotzdem gibt es immer wieder populäre Applikationen, die in einer Wine-Umgebung nicht ablaufen. Noch größere Probleme bestehen mit den Hardwareabstraktionsschichten wie DirectX. Hier sieht man den Nachteil proprietärer APIs: Ohne vollständige Information lässt sich kein im Sinne von Zuverlässigkeit kompletter Emulator schreiben. Zudem läuft ein solcher Emulator den aktuellen Entwicklungen hinterher, was vielleicht für die Langzeitarchivierung ein geringerer Nachteil ist.

Weiterhin kann ein Betriebssystememulator nicht von der darunterliegenden Hardware soweit abstrahieren, als dass darauf laufende Applikationen unabhängig von der darunter liegenden Rechnerarchitektur funktionieren. Damit scheidet die Emulation von Betriebssystemen als Konzept für den langfristigen Zugriff auf digitale Objekte aus.

### 3.5.4 Emulation der kompletten Hardwareplattform

Der Ansatz, die komplette Hardware einer bestimmten Rechnerplattform zu archivieren, nutzt eine noch tiefere Schicht (Abb. 3.9). So scheint es deutlich attraktiver, eine Hardwareplattform in Software nachzubilden:

- 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 Funktionenset einer Hardwareplattform ist überschaubar und oft deutlich geringer als das eines Betriebssystems oder einer Applikation.
- Es existieren deutlich weniger Hardwareplattformen als Betriebssysteme (Abb. 3.8).

Deshalb reicht die Idee, 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 unterschiedlichen Architekturen entwickelt haben. So sind üblicherweise alle Funktionen eines Prozessors umfassend dokumentiert, da nur in den seltensten Fällen CPU- und Betriebssystemhersteller identisch sind. Nur durch eine gute Dokumentation kann es dem Hardwarehersteller gelingen, Softwareentwickler von seinem Produkt zu

<sup>43</sup>Dieser Ansatz würde es erlauben, durchaus verschiedene Linux Varianten, die sich beispielsweise in der verwendeten C-Bibliothek unterscheiden, auf einem einzigen Kernel auszuführen. Zwingende Voraussetzung ist jedoch die Verwendung eines gemeinsamen ABI (Application Binary Interface). Informationen zu OpenVZ und VServer finden sich in [Thorns 2007] Kapitel 19, 20.

überzeugen. Diese Offenheit findet sich vielleicht nicht mehr in diesem Ausmaß für die Peripherie einer bestimmten Plattform wieder. Jedoch gibt es auch hier einen großen Umfang umfassend beschriebener Hardware, auf die in solchen Situationen bevorzugt zurückgegriffen werden sollte.

Auf diese Weise lassen sich komplette Ablaufumgebungen für digitale Objekte aufbewahren, ohne von komplexen Randbedingungen, wie APIs und schlecht dokumentierten Datenformaten, abhängig zu sein.

Hinzu kommt ein wesentlicher Vorteil für alle dynamischen digitalen Objekte. Funktionsaufrufe, die für eine komplett andere Hardwareplattform geschrieben wurden, stellen kein Problem dar, da sie komplett in Software nachgebildet werden. Diese Möglichkeit besteht in den erstgenannten Emulationsansätzen nicht.

### 3.5.5 Emulation im Einsatz

Die Betrachtungen von [Holdsworth und Wheatley 2001] und [Granger 2000] bieten grundsätzliche Überlegungen und Ideen für Ansatzpunkte.

Während die Emulation von Applikationen oder Dateiformaten für die tägliche Anwendung durchaus von einiger Bedeutung ist, so wird sie selten unter diesem Stichwort diskutiert. Der Ansatz der Betriebssystememulation spielt eine gewisse Rolle, erbringt jedoch keine Vorteile für den langfristigen Zugriff auf digitale Objekte. Wenn in jüngerer von Emulation und Langzeitarchivierung gesprochen wurde, so ist damit die Hardwareemulation gemeint. Sie konnte sich zudem im Umfeld der Softwareentwicklung durchsetzen: Hardwareemulation erlaubt Herstellern, die Entwicklung von Software für ihre neue Hardware schon weit vor dem Erscheinen anzustossen. Dieses ist beispielsweise im Bereich der Spielekonsolen zu beobachten. Andererseits erlaubt es Systemanbietern wie Hardware- und Softwareherstellern, einen leichteren Umstieg auf neue Plattformen zu realisieren und damit neue Konzepte ohne Rücksichtnahme auf Rückwärtskompatibilität durchzusetzen. Ältere Programme werden dann in einer emulierten Umgebung ausgeführt und stehen auf diese Weise dem Anwender weiterhin zur Verfügung. Unterstützt werden diese Entwicklungen durch die sich immer stärker etablierende Virtualisierung auf der PC-Architektur.

Nach etlichen Vorüberlegungen, wie sie [Rothenberg 2000] oder [Lorie 2002] vornehmen, gibt es nun eine gewisse Menge an Erkenntnissen, wie die des niederländischen DIAS-Projekts [van Diessen und Steenbakkers 2002]. Aktuelle Erfahrungen des Einsatzes der Emulationsstrategie finden sich in [van der Hoeven u. a. 2007].

## 3.6 Emulation und Migration im Vergleich

Bevor aufgrund ihres scheinbar naheliegenden Charakters auf Migration zur Archivierung digitaler Objekte gesetzt werden kann, ist diese Strategie in ihren möglichen Auswirkungen zu bewerten. Migration wirkt auf den verschiedenen Ebenen, in denen sie durchgeführt werden kann, eine Reihe kleinerer und gravierenderer Probleme auf.

Auch wenn die Emulation viele Probleme einer funktionalen Langzeitarchivierung elegant zu lösen hilft, so ist sie doch nicht in allen Fällen die erste Antwort auf diese Frage. Eine gute Diskussion der Fragen von Emulation und Migration gibt [Holdsworth und Wheatley 2004]

anhand einer Reihe von in Großbritannien durchgeführten Forschungsprojekten.<sup>44</sup> Einige un- oder noch nicht befriedigend gelöste Probleme im Zusammenhang mit Emulationsstrategien stellt [Sereiens und Hilf 2003] zusammen.

**Migration als Archivierungsstrategie** Migration erweist sich dort als vorteilhaft, wo permanent auf eine große Anzahl digitaler Objekte mit jeweils aktuellen Anwendungsprogrammen zugegriffen werden soll. Lernprozesse für den Umgang mit der notwendigen Software finden permanent statt. Deutlich schwieriger wäre schon die Vorstellung, dass Benutzer noch mit der Software der ersten Rechnergenerationen von vor zwanzig Jahren und mehr umgehen können.

Ebenfalls vorteilhaft kann sich die einfache Weiterverarbeitung migrierter Objekte erweisen. Sie lassen sich mit den aktuellen Werkzeugen verändern und leicht in bestehenden Systemen durchsuchen oder publizieren. Damit einhergehen können Verbesserungen gegenüber dem Originalformat. Man muss sich nur einmal vorstellen, welche Möglichkeiten des Layouts und der Dokumentauszeichnung gegenüber den ersten einfachen Textbearbeitungs- und Textsatzsystemen bestehen. Beispiele sind Multimedia-Dokumente wie Präsentationen oder Querverbindungen mit Verweisen.

Dokumente, die permanent in Benutzung stehen, werden quasi automatisch in ihrer Qualität überwacht, womit Änderungen bei der Datentransformation schon früh auffallen sollten und eine eventuell notwendige Reaktion zeitnah erfolgen kann. Migration erweist sich als unproblematisch, wenn die Authentizität der Objekte sichergestellt ist (Abb. 3.10).

Zumindest bei häufig verwendeten Dokumenttypen werden sich mit hoher Wahrscheinlichkeit Standards etablieren. Für diese Standards ist der Bedarf an Konversionswerkzeugen entsprechend hoch, so dass von einer guten Verfügbarkeit ausgegangen werden kann. Die eben geschilderten Vorteile betreffen häufig jedoch nur eine Teilmenge des Gesamtdatenbestandes. Zudem weichen die Anforderungen von Bibliothekaren und Archivaren an digitale Objekte stark von denen der "normalen" Benutzer ab. Bei Migration handelt es sich nicht um eine einheitliche Methode zum generellen Umgang mit digitalen Objekten. Sie umfasst ein ganzes Strategiebündel. Die Einzelstrategien hängen sehr stark von den beteiligten Quell- und Zielformaten ab und lassen sich damit kaum generalisieren.

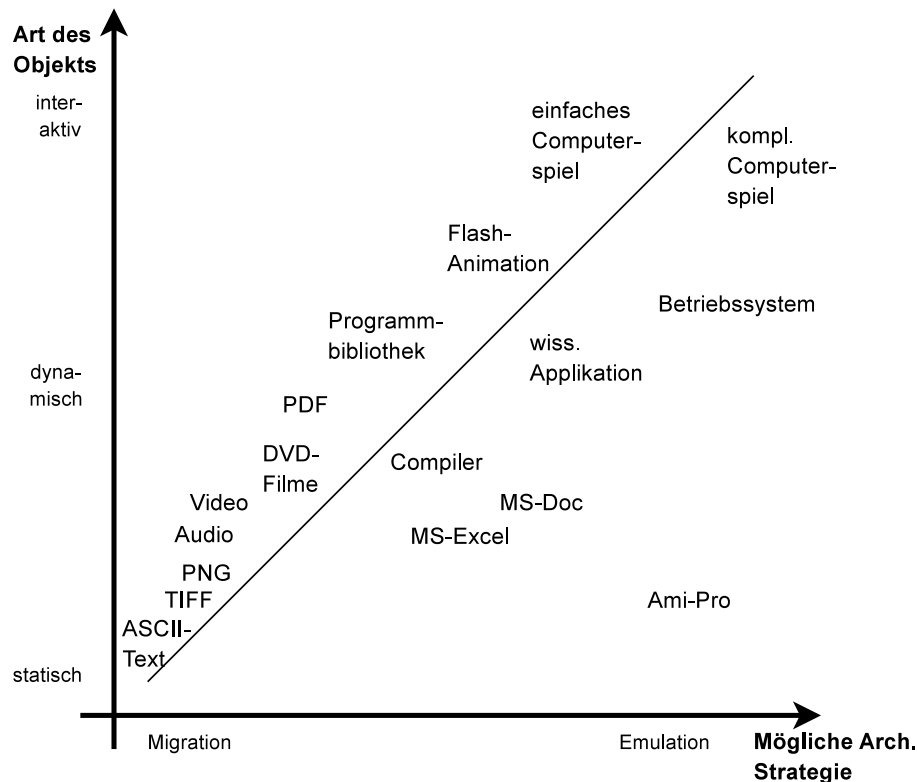
Sehr kritisch wird die Migration als Langzeitarchivierungsstrategie in [Rothenberg 1999], S. 13 sowie [Rothenberg 2002], S. 13 ff. beleuchtet.

**Migration von digitalen Objekten primären Interesses** Kriterien, wie Suchbarkeit, Betrachtung in aktuellen Computern und Weiterverarbeitbarkeit, sprechen für die Migration von digitalen Objekten in ihre jeweils passende Repräsentation. Wenn dabei gleichzeitig die Originale aufgehoben werden, so kann in emulierten Umgebungen jederzeit die Authentizität des migrierten Objekts festgestellt werden. Ebenso bietet die Emulation die Möglichkeit erfolgte Migrationsschritte erneut nachzuvollziehen.

Jedoch bleiben etliche digitale Objekte von vornherein von der Migration ausgeschlossen: Alle dynamischen Objekte, wie Programme, Bibliotheken und Betriebssysteme, deren Quellcode nicht offen liegt, können nicht auf die jeweils aktuelle Plattform übertragen werden. Vielfach ist das Interesse an diesen Objekten zu klein, als dass es sich ökonomisch lohnt, hier

---

<sup>44</sup>Vgl. Abschnitt 2.6.



**Abbildung 3.10:** Vorschlag einer Einordnung verschiedener digitaler Objekte in Abhängigkeit ihres Typs in eine Langzeitarchivierungs-Strategiematrix.

eine Softwaremigration anzustoßen. Zudem sind die Abhängigkeiten von Programmen und Bibliotheken vom Betriebssystem und einer darunterliegenden bestimmten Hardwareschicht zu komplex, als dass sich dieses Unterfangen auf eine Vielzahl digitaler Objekte anwenden ließe.

**Migration von Emulatoren und ihrer Hilfsmittel** Migration kann zudem für Emulatoren eine nicht zu vernachlässigende Option darstellen, wie im Abschnitt 4.8 bereits ausgeführt wurde. Gerade dadurch, dass sich an der emulierten Umgebung aufgrund ihres historischen Charakters nichts mehr ändert, kann es einfach sein, einen Emulator auf eine neue Umgebung zu portieren, mithin zu migrieren. Hierbei handelt es sich häufig nur um Änderungen im Benutzer-Interface, die nicht zwingend eine komplette Neuprogrammierung eines Emulators erfordern müssen. Ändert sich die Host-Plattform aufgrund technischen Fortschritts, so haben nur vollständige Emulatoren eine Chance auf Migration. Andernfalls müsste eine komplette Emulation der vormals nur virtualisierten Komponenten hinzugefügt werden.

Ebenfalls von Interesse ist die Migration von Hilfsprogrammen, die beispielsweise zum Lesen oder Umwandeln der verschiedenen Containerformate der eingesetzten Emulatoren verwendet werden. Dieses gilt ebenso für Programme zum Einhängen von Festplatten-, ISO- oder Disketten-Images in das Dateisystem der gerade verwendeten Plattform. Diese Fragestellungen werden im späteren Abschnitt 7.7.2 behandelt.

**Menge der zu berücksichtigenden Objekte** Migration setzt am digitalen Objekt selbst an. Es handelt sich um einen periodisch notwendigen Prozess. Die Wiederholung einer notwendigen Anpassung kann man zwar durch die Wahl von Standardformaten reduzieren, auszuschließen ist sie jedoch nicht. Bei sehr großen Objektzahlen kommen damit nur automatisch ablaufende Migrationstechniken in Frage, besonders da die Menge der Daten stetig zunimmt.

Bei der Wahl einer geeigneten Emulationsebene lässt sich die Zahl der zu beachtenden digitalen Objekte deutlich besser überschauen (Abb. 3.8). Die Menge der Rechnerarchitekturen, besonders wenn man sich auf die populären und für die Erstellung der meisten Objekte am häufigsten eingesetzten Plattformen beschränkt, ist im Verhältnis zur Zahl der Objekte verschwindend klein.

Die Grenzen der Emulation zeigen sich derzeit für komplexe Netzwerkanwendungen, die eine ganze Reihe verschiedener, miteinander vernetzter Maschinen involvieren. Hierzu zählen beispielsweise große Online-Spiele. Ebenfalls stoßen Emulationsstrategien an ihre Grenzen, wo um spezielle, nicht dokumentierte Hardware, wie Dongles zum Schutz von Software, geht. Emulatoren können zwar das Aussehen alter Benutzer-Interfaces abbilden. Ihre Begrenzung liegt jedoch in den Aspekten der eigentlichen Hardware: Aussehen, bestimmtes Verhalten von Peripheriegeräten oder die Haptik lassen sich nicht übertragen.





## 4 Emulatoren - Recherche und Auswahlkriterien

Hardwareemulatoren können, wie im letzten Kapitel ausgeführt wurde, 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 Ablaufumgebungen für diese Objekte vorbei. Wie bereits dargestellt, eröffnet ein Hardwaremuseum keine wirklich zuverlässige Langfristperspektive, denn die Archivierung von alten Computern wird langfristig aus vielen Gründen kostenintensiv und zunehmend risikoreich. Deshalb bietet eine virtuelle Sammlung vergangener und sich im Prozess der Ablösung befindender Rechnerarchitekturen eine sinnvolle Alternative mit etlichen Vorteilen.

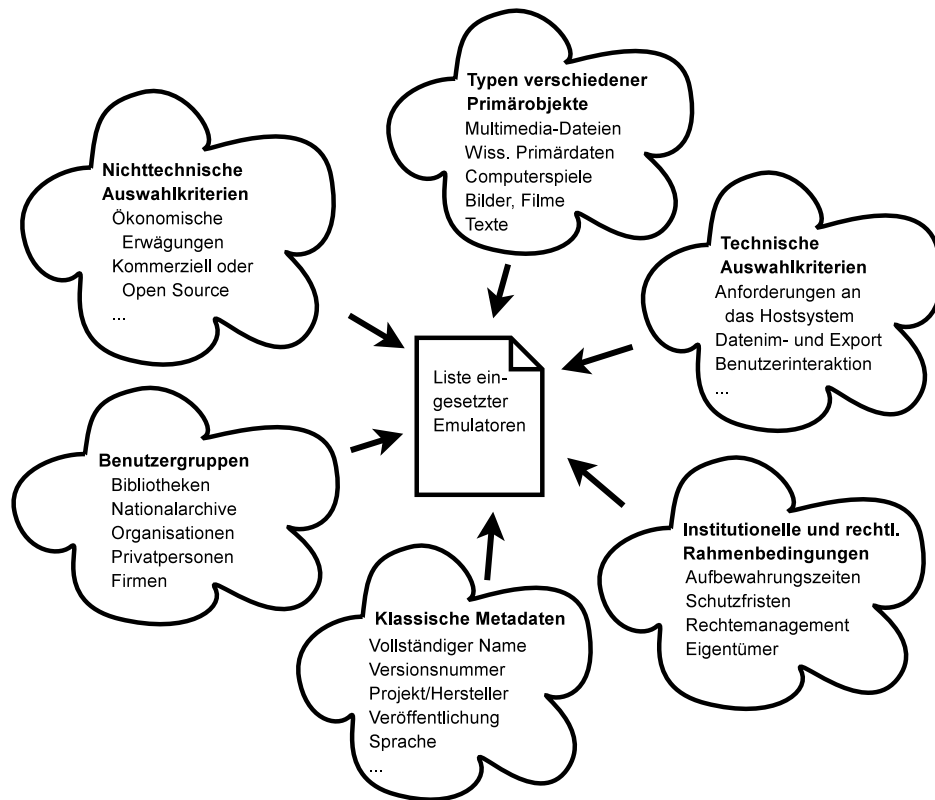
Emulatoren für die unterschiedlichen Computersysteme werden daher zu einem zentralen Bestandteil eines zu schaffenden Softwarearchivs. Dieses Softwarearchiv kann seinerseits wieder einen Bestandteil eines OAIS-basierten Langzeitarchivs bilden. Emulatoren sind ebenso wie die eigentlich interessierenden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

Die Aufgabe dieses Kapitels besteht darin, eingangs zu zeigen, dass bereits eine große Menge an Hardwareemulatoren im Laufe der letzten 30 Jahre entstanden ist. Hierzu erfolgte eine Recherche nach verfügbaren Emulatoren. Soweit es möglich erschien, wurden kursorische Tests vielversprechender Ansätze auf deren Ausführbarkeit auf einer möglichen Host-Plattform durchgeführt. Hierbei wurde die Intensität und Vollständigkeit der Überprüfung auf generelle Funktionsfähigkeit bewusst niedrig gehalten. 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. Da diese ebenfalls aus Sicht einer Archivaufnahme von Bedeutung sind, führen sie die Liste der Kriterien an.

Weiterhin beschäftigt sich dieses Kapitel mit der Aufstellung zusätzlicher Auswahlkriterien, die helfen sollen:

- als wichtig erachtete Klassen digitaler Objekte zu bedienen,
- institutionelle Rahmenbedingungen in Betracht zu ziehen,
- relevante Benutzergruppen und ihre (vermuteten) Bedürfnisse zu identifizieren,
- technische und nicht-technische Charakterisierungen vorzunehmen und die Vollständigkeit der Nachbildung abzuschätzen,

- Anhaltspunkte zur ökonomischen Bewertung zu erhalten und
- die Langfristorientierung abzuschätzen.



**Abbildung 4.1:** Gruppen von Auswahlkriterien für die Liste der ins Archiv aufzunehmenden oder wünschenswerten Emulatoren.

Die Emulatoren selbst sind Applikationen, die auf der jeweils aktuellen Rechnerplattform des Archivnutzers oder in einer geeignet definierten Bezugs- oder Referenzumgebung ablaufen. Diese Umgebungen sind Bestandteil der Betrachtungen in Kapitel 6. Hierbei greifen sie auf eine Reihe von Ressourcen zu, die vom *Host-System* zur Verfügung gestellt werden. Umgekehrt müssen sie die Nutzungsumgebung für das jeweilige Zielobjekt möglichst vollständig und umfassend nachbilden.

Die Auswahl der inzwischen kommerziell erhältlichen sowie 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.

Nicht alle Auswahlkriterien lassen sich auf alle Rechnerplattformen und ihre Emulatoren anwenden. 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 die genannte Home Computer Firmware, ist frei verfügbar. Für die X86-Architektur liegt der Fall mit der Verfügbarkeit eines Open-Source-System- und Grafikkarten-BIOS einfacher.

Ein wichtiges, jedoch schwer abzuschätzendes, weil in die Zukunft reichendes Kriterium ist die langfristige Perspektive: Die Emulatoren sind wie jedes Primärobjekt von Veralterung betroffen. Um nicht dadurch die Les- oder Zugreifbarkeit der interessierenden Objekte zu gefährden, sind geeignete Maßnahmen zur Langzeiterhaltbarkeit dieser Software zu ergreifen. Inzwischen existiert eine Reihe von Ansätzen, die Migration, Modularisierung oder Kapselung zur Erreichung dieses Ziels nutzen. Diese sind teilweise schon im Einsatz oder befinden sich in Erforschung und Entwicklung.

Aus Sicht der Langzeitarchivierung besteht zudem eine Divergenz: Der überwiegende Anteil von Emulatoren und Virtualisierern wurde aus ganz anderen als Langzeitarchivierungsgründen erstellt. Sie sind heutzutage Standardwerkzeuge in der Softwareentwicklung. Nichtsdestotrotz eignen sich viele der im folgenden vorgestellten Werkzeuge für eine Teilmenge möglicher Langzeitarchivierungsaufgaben.

## 4.1 Recherche verfügbarer Emulatoren

Das Konzept der Hardwareemulation existiert erstaunlich lange. Die Idee, Rechnerarchitekturen in Software nachzubilden, hat viele Gründe, die von der Bereitstellung einer Entwicklungsplattform für andere Gerätetypen bis zum Enthusiasmus gegenüber speziellen Maschinen reicht. Die dafür notwendige Software lief üblicherweise als Programm außerhalb des privilegierten Kernel-Kontexts und war damit mit jeder für die jeweilige Plattform erhältlichen oder erstellten Applikationen vergleichbar. Deshalb standen am Anfang dieser Arbeit Untersuchungen, welche Programme bereits vorhanden sind oder an welchen in dieser Richtung gelagerten Projekten gearbeitet wird (Abschnitt A.1).

Ziel war es, einen ersten Überblick zu gewinnen, um abschätzen zu können, für welche Rechnerarchitekturen relevante Emulatoren, damit sind produktiv nutzbare oder sich dahin entwickelnde Programme und Projekte gemeint, zur Verfügung stehen. Dabei wurden für die Host-Systeme anfangs keine Beschränkungen angenommen, da sich für im Sinne der Langzeitarchivierung gute Werkzeuge<sup>1</sup> auch in geschachtelten Umgebungen einsetzen lassen.

Die erste Recherche nach verfügbaren Emulatoren wurde bewusst weit gefasst. Ausgewertet wurden in erster Linie Internet-Quellen. Diese haben gegenüber Büchern oder gedruckten Periodika den wesentlichen Vorteil, dass sie einerseits die jeweils aktuellsten Informationen bereithalten und andererseits oft gleichzeitig die Möglichkeit des Bezugs der ausführbaren Programme oder der Quelltexte erlauben. Generell lässt sich bei dieser Art der Recherche bereits auf eine Relevanz schließen: Wenige Referenzen bei Projekten oder Produkten lässt auf zu geringe Kundenzahl oder sehr kleine User-Communities schließen. Das birgt häufig die Gefahr einer fehlenden Nachhaltigkeit.

Parallel dazu erfolgte eine kursorische Recherche in den einschlägigen (deutschsprachigen) Computer-Magazinen,<sup>2</sup> um zu verhindern, dass in der Menge der oft unstrukturierten

<sup>1</sup>Die Zahl der relevanten Referenzplattformen wird klein sein (Abschnitt 6.5), lässt sich aber durch Emulator-Stacking erweitern, siehe hierzu Abschnitt 4.7.

<sup>2</sup>ct, iX aus dem Heise-Verlag inclusive Online-Auftritt mit [HZV 2007], seriös mit dem höchsten Verbrei-

Informationen des Internets wichtige Emulatorprojekte übersehen werden. Sie ergab keine zusätzlichen Erkenntnisse. Stattdessen lieferte sie Anhaltspunkte und Testberichte oft vielversprechender Ansätze auf deren Ausführbarkeit auf einer Host-Plattform und deren generellen Funktionsfähigkeit.

Die nachstehenden Untersuchungen nehmen zuerst wenig Rücksicht auf die Relevanz der einzelnen Plattformen. Diese hängt stark von der Zielgruppe (Abschnitt 4.5) und ihren langzeitzuarchivierenden digitalen Objekten ab.

## 4.2 Klassen von Rechnerplattformen

Zur Einordnung der Recherche-Ergebnisse wurden bestimmte Rechnerplattformen in Klassen zusammengefasst. Diese Klassenbildung orientiert sich am Zeitstrahl der Computergeschichte, wie sie beispielsweise in [CHM 2008] gut dargestellt wird.<sup>3</sup>

Eine Klassenbildung könnte auch anders aussehen: Eine Kategorisierung nach Anwendung nehmen die Autoren von [van der Hoeven und van Wijngaarden 2005] vor. Sie ordnen beispielsweise nach dem Einsatz in bestimmten Nutzungsumgebungen und teilen diese in "Game Computer Platforms", "Historic Computing", "Business Efficiency" und "Cross Platform Emulation" auf.

### 4.2.1 Frühe Großrechner

Da Mainframes, sogenannte Großrechner, und später verschiedene Unix-Workstations die Anfänge der breiteren kommerziellen Computernutzung dominierten (Abb. 4.2), gibt es ebenfalls in diesem Bereich Bemühungen, sie durch Emulation nachzubilden. Die Zahl der damit befassten Personen und auf diesen Maschinen erstellten digitalen Objekte ist verglichen mit modernen Plattformen noch sehr gering. Diese Maschinen sind daher häufiger als Primärobjekte in Museen und Ausstellungen, denn als Werkzeuge zur Rekonstruktion von Ablaufumgebungen interessant. Spätere, leistungsfähigere Systeme, die zur Steuerung von Abläufen oder Berechnungen eingesetzt wurden, spielten in Firmen und im Großforschungssektor eine nicht unwesentliche Rolle.

### 4.2.2 Der sogenannte Industriestandard: X86

Mit Beginn der 1990er Jahre hat sich die PC-Architektur als eine Art gemeinsamer Standard durchgesetzt und neue Technologien, Ideen und Designs hervorgebracht. Architekturen, die einmal als Standard definiert wurden, um eine Kompatibilität von Code und Programmen zu erreichen – eine wesentliche Voraussetzung für Planungssicherheit und Investitionsschutz – werden ab einem bestimmten Punkt der Weiterentwicklung selbst zum Hindernis. Die Beispiele des PentiumPro-Prozessors und des Itaniums zeigen, dass Rückwärtskompatibilität nur

---

tungsgrad, Linux-Magazin. Die Leserschaft deckt die Spannbreite des ambitionierten privaten Computernutzers bis zum professionellen EDV-Fachmann ab.

<sup>3</sup>Die in einer Plattform eingesetzte CPU kann als ein Entscheidungskriterium genommen werden, siehe dazu beispielsweise <http://www.computerhistory.org/microprocessors>. Die Geschichte der Spielecomputer findet man in [CSM 2008] beginnend mit dem Jahr 1972.



**Abbildung 4.2:** Das Rechenzentrum der Universität Freiburg betrieb in den 1970, 80er Jahren einen Großrechner der Firma UNIVAC.

in bestimmten Grenzen sinnvoll realisierbar ist. Mit dem AMD Opteron-Prozessor wurde ein besserer Wanderer zwischen der alten 32 bit und der neuen 64 bit-Welt eingeführt - Die Entwicklungsrichtung ist klar: Der 32 bit-Industriestandard-PC erleidet dasselbe Schicksal wie die Mainframes und Home-Computer und tritt in den nächsten Jahren den Weg in die IT-Geschichte an.

Jedoch wurde auch mit der PC-Architektur die Vielfalt der Rechnerplattformen trotz Standardisierung und teilweiser Monopolisierung nicht aus der Welt geschafft. Nicht alle Anwendungen und Aufgaben lassen sich mit ihr erledigen: Die Generation der transportablen Geräte, wie Mobiltelefone, Organizer und PocketPC erfordern andere Prozessoren und schlanke Betriebssysteme.

Die Popularität des X86-PC hat jedoch ihre Schattenseiten. Anders als vorangegangene Architekturen ist die Palette bisher produzierter Hardware extrem breit. Dieses bringt einige Herausforderungen an X86-Emulatoren und Virtualisierer mit sich.

### 4.2.3 Weitere Plattformen

Im Bereich der Video- und Computerspiele hat sich eine ziemliche Breite verschiedener Plattformen entwickelt. Aus Software- oder Hardwaresicht lassen sich folgende Kategorisierungen treffen:

- "Klassische" Computerspiele bezeichnen Programme, die auf einem typischen Universalgerät abliefen, das nicht primär zum Spielen geschaffen wurde. Dieses beginnt mit

einfachen textbasierten Programmen auf der Textkonsole von Großrechnern, erlebte die erste Blüte auf den Home-Computern und reicht bis zum Multimedia-PC der heutigen Zeit.

- Arcade-Spiele transportierten die neue Technologie in die Spielhallen: Sie sind typischerweise mit Automaten zu spielen, die gegen Geldeinwurf benutzt werden konnten.
- Sogenannte Videospiele sind nach ihrem Ausgabegerät, typischerweise dem heimischen Fernseher, benannt. Die Plattform heißt üblicherweise "Konsole", womit ein nur zum Spielen gedachter Computer bezeichnet wird. Diese Geräteklasse wurde für eine gute Audioausgabe und ansprechende grafische Darstellung auf dem Fernseher optimiert.
- Mit der Miniaturisierung kamen Ende der 1990er Jahre tragbare Videospiele hinzu. Diese Geräteklasse bringt die Aus- und Eingabeeinheiten und den Rechner in einem einzigen kompakten, portablen Gehäuse unter. Dieser Bereich dehnt sich verstärkt auf Mobiltelefone und PDAs aus.

Für den überwiegenden Anteil der Universalcomputer liegen Emulatoren vor, welche die Spiele dieser Geräteklassen ausführen können. Das ist für spezielle Videospiele-Konsolen und Taschengeräte oft schwieriger. Hier existieren für einige Geräte frei verfügbare oder kommerziell erhältliche Entwicklungsumgebungen und Emulatoren.<sup>4</sup>

#### 4.2.4 Home-Computer

Während die auf der X86 erstellten oder in dieser Umgebung darstellbaren digitalen statischen und dynamischen Objekte in der weiten Überzahl sind, gibt es sicherlich einige Objekte der Home-Computer-Ära, die für die Langzeitarchivierung relevant sind. Für Home-Computer spielen hingegen in erster Linie dynamische Objekte eine Rolle. Hierzu zählen die Firmwares der einzelnen Maschinen, erste grafische Benutzeroberflächen und Applikationen. Bei letzteren stehen sicherlich die Computerspiele im Vordergrund. Spiele dominierten sicherlich die Gründe zur Beschaffung solcher Maschinen. Der Markt dieser Geräteklasse war stark segmentiert und nicht durch wenige Teilnehmer dominiert. Daraus resultierte eine ziemliche Breite verschiedener Hardwaredesigns.

#### 4.2.5 Spielplattformen

Computer wurden schon sehr früh als reine Unterhaltungsgeräte eingesetzt. Einerseits wurden Geräte für den Endverbraucher verkauft, die anders als Home-Computer nur mit wenigen Elementen zur Steuerung bestückt waren und je nach System entweder fest eingebaute Software oder durch Wechsel eines Datenträgers unterschiedliche Spiele ablaufen lassen konnten. Andererseits gab es für Jahrmärkte und Spielhallen Systeme, die für den Einsatz im öffentlichen Raum produziert wurden. Sie verfügten dann über Münzeinwurf und ein oder mehrere fest installierte Spiele, die über wenige Schalter oder Joysticks gesteuert wurden.

---

<sup>4</sup>So liefert die Entwicklungsumgebung für das Symbian-OS, welches auf Mobiltelefonen und PDAs anzu-treffen ist, neben den notwendigen Bibliotheken und Compilern auch einen Emulator mit.

Mit der zunehmenden Miniaturisierung der elektronischen Hardwarekomponenten dehnte sich der Bereich der Spielmaschinen in den mobilen Markt aus und brachte eine Geräteklasse wie den GameBoy hervor. Parallel dazu entwickelten sich die sogenannten Spielekonsolen weiter und eroberten einen immer breiteren Markt. Die Maschinen, wie PlayStation oder GameCube wurden üblicherweise an den Fernseher angeschlossen und mit speziellen Controllern bedient. Bei dem größten Teil der digitalen Objekte dieses Bereichs handelt es sich deshalb um Computerspiele. Solche Objekte werden beispielsweise vom Computerspiele-Museum in Berlin<sup>5</sup> archiviert.

## 4.3 Erste Iteration der Emulatorensuche

Für die erste Recherche wurde nur eine oberflächliche Abschätzung der Relevanz vorgenommen. Die grobe Klassifikation in die im folgenden vorgestellten Gruppen erfolgte anhand der CPU oder der Plattform allgemein:

- Der Prozessor ist das wesentliche Distinktionsmerkmal der Klasse der Intel-kompatiblen X86-Computer. Software, die für Betriebssysteme dieser Rechnerklasse geschrieben wurde, ist abgesehen von Einschränkungen aufgrund der technologischen Fortentwicklung auf allen Maschinen binärkompatibel.
- Auf der anderen Seite wurde die gleiche CPU, wie die Motorola 68000er Linie in sehr verschiedenen Maschinen verbaut – angefangen von der Klasse der Home-Computer über frühere Apple Macintosh Maschinen bis hin zu bestimmten Unix-Workstations von SUN Microsystems oder Geräten wie Zyxel-Modems. In diesem Fall erscheint es sinnvoller, entlang der Plattform zu unterscheiden und Amiga, Apple Macintosh und Unix-Workstations in eigene Kategorien einzusortieren.

Diese Klassifikationen sind nicht statisch und könnten durch neuere Entwicklungen im Bereich der Emulatoren überholt werden, wenn beispielsweise ein modularer Motorola-68000-Emulator eine Brücke über verschiedene Rechnerarchitekturen und Hersteller hinweg schlägt.

Einer der Ausgangspunkte der Recherche war das Google-Directory zu Emulatoren,<sup>6</sup> welches ebenfalls eine gewisse Klassifikation anhand von Rechnerplattformen vornimmt. Für den Überblick wurde mit einem minimalen Set an Metadaten<sup>7</sup> gearbeitet, welches lediglich:

- den Namen des Produkts oder des Projekts,
- eine Online-Referenz und
- eine knappe Erstbewertung

<sup>5</sup>Siehe Homepage [CSM 2008]. Geschichte der Spielecomputer ebd. im Bereich "Online-Expo / Spielmaschinen".

<sup>6</sup><http://www.google.com/Top/Computers/Emulators> fasst das Thema Emulation deutlich weiter als reine Hardwareemulation. So werden zusätzlich Emulatoren für einzelne Hardwarekomponenten, wie Laufwerke oder Betriebssystememulatoren, siehe hierzu Abschnitt 3.5.3, aufgeführt. Eine weitere Quelle und aktuelle Linksammlung bietet [EMUU 2008].

<sup>7</sup>Siehe hierzu den späteren Abschnitt 4.4 dieses Kapitels.

angibt.

Wegen der inzwischen großen Auswahl an Emulatoren und Virtualisierern für verschiedene Plattformen kann es zudem interessant sein, die Austauschbarkeit von Ein- und Ausgabedateien sowie die Migrierbarkeit von Containerformaten zu betrachten.

### 4.3.1 X86-Computer

Das letzte Jahrzehnt vor der Jahrtausendwende war durch den sogenannten Personal Computer (PC) geprägt. Basierend auf einer Intel-CPU<sup>8</sup> wurde diese Architektur zum Quasi-industriestandard. Das bedeutete, dass die große Vielfalt verschiedener Unix-Plattformen, Mainframe-Architekturen und Consumer-Produkte durch eine einheitliche, einigermaßen standardisierte, dabei offene Architektur<sup>9</sup> zunehmend an den Rand gedrängt und abgelöst wurde. Diese Entwicklung schuf die Grundlage für den Erfolg von Microsoft, das den Markt der X86-Betriebssysteme weitgehend bestimmen und monopolisieren konnte. Aus diesem Grund ist die Auswahl an Emulatoren und Virtualisierern inzwischen recht vielfältig.

**Bochs** ist ein Open-Source-Projekt zur Emulation der 32 bit-PC-Architektur. Es existiert seit Ende der 1990er Jahre und erfreut sich einer recht großen Benutzergemeinschaft. Dieser Emulator steht jedoch etwas im Schatten des erfolgreicheren QEMU.

**Dioscuri** wurde primär für die funktionale Langzeitarchivierung an der Königlichen Bibliothek und dem Nationalarchiv der Niederlande<sup>10</sup> entworfen und wird im Rahmen des PLANETS Projekts weiterentwickelt [PLANETS 2008]. Dieser Java-basierte Emulator kann bereits eine 8086-CPU emulieren und damit DOS ausführen.

**DOSBOX** kann man als Hybrid eines Hardware- und Betriebssystememulators betrachten. Dieses Open-Source-Werkzeug,<sup>11</sup> welches unter aktiver Weiterentwicklung steht, bietet in erster Linie die Möglichkeit, Spiele für DOS auf Linux Plattformen ausführen zu können.

**DOSEMU** gehört zu den älteren Virtualisierern. Dieses Projekt<sup>12</sup> ist Open Source Software und existiert bereits seit Ende der 1990er Jahre, wird aber immer noch weiterentwickelt und an aktuelle Host-Betriebssysteme angepasst. Bei diesem Emulator handelt es sich wie bei DOSBOX eher um einen Hybrid, wobei das einzig unterstützte Zielbetriebssystem DOS (anders als bei der vorgenannten DOSBOX) zusätzlich installiert werden muss.

<sup>8</sup>Oder entsprechenden (pin)kompatiblen Prozessoren weiterer Hersteller, wie AMD, Cyrix.

<sup>9</sup>Siehe [CHM 2008], <http://www.computerhistory.org/timeline/?category=cmptr>. IBM reagierte mit dem eigenen PC auf den Markterfolg des Apple II und anderer Maschinen und läutete damit die PC-Ära ein. So brachte beispielsweise Compaq schon 1983 einen IBM-kompatiblen Rechner auf den Markt, der in der Lage war, mit exakt der gleichen Software wie die IBM-PC zu arbeiten.

<sup>10</sup>Homepage des Emulatorprojekts - <http://dioscuri.sourceforge.net>, [DIOSCURI 2008].

<sup>11</sup>Eine nicht uninteressante Besonderheit ist, dass kein separates DOS mehr installiert werden muss. Homepage - <http://dosbox.sourceforge.net>.

<sup>12</sup>Homepage des DOSEMU-Projekts - <http://dosemu.sourceforge.net>.



**FAUmachine** ist eine spezielle virtuelle Maschine, die PC Hardware simulieren kann. Sie läuft als normaler User-Prozess auf Linux Hosts mit X86 Architektur, kann aber analog zu QEMU durch einen Kernelpatch beschleunigt werden. Das Projekt wurde für Soft- und Hardwareverifikation<sup>13</sup> erstellt und ist daher nicht direkt mit anderen Projekten und Produkten vergleichbar. Es ist Open Source und kann daher als Vorlage dienen, wie sich bestimmte Peripherie in Software nachbilden lässt.

**JPC** der Java PC Emulator,<sup>14</sup> welcher an der Universität Oxford entwickelt und Mitte 2007 als Open Source freigegeben wurde, läuft als Java Applet in Java Virtual Machines ab Version 1.5. Es gibt bereits die Unterstützung des Real-Mode der CPU und es kann DOS und darauf laufende Applikationen ausführen.

**Parallels Workstation** stammt von SWsoft.<sup>15</sup> Hierbei handelt es sich um einen eher kleineren Anbieter im Virtualisierungsmarkt, der jedoch als erstes seine Produkte für die drei Host-Plattformen Windows, Linux<sup>16</sup> und Mac-OS X<sup>17</sup> anbot. Der Funktionsumfang liegt ähnlich wie bei der VMware Workstation oder Virtual PC von Microsoft.

**Plex86** zählte zu den ersten, im Sinne von Open Source freien X86-Emulationsprojekten.<sup>18</sup> Es ist selbst seit Ende 2003 nicht mehr aktiv, hat jedoch anderen Open-Source-Emulatorprojekten das BIOS geliehen.

**QEMU** gehört zu den derzeit sicherlich populärsten Open-Source-Emulatoren.<sup>19</sup> Dieser Emulator kann mit speziellen Virtualisierungsmodulen auf X86-Hosts ebenfalls als Virtualisierer eingesetzt werden. Er steht für weitere als lediglich die üblichen Linux und Windows Host-Plattformen zur Verfügung.

**Virtual PC für PPC** zählte zu den ersten kommerziellen PC-Emulatoren, welches von der Firma Connectix<sup>20</sup> entwickelt wurde. Mit diesem Programm ließen sich PC-Betriebssysteme in einer emulierten Umgebung auf der Apple PowerPC-Architektur betreiben. Mit der Aufgabe dieser Architektur seitens Apple wird dieses Produkt nicht mehr weiterentwickelt. Es unterstützt offiziell lediglich DOS bis Windows 98 Nutzungsumgebungen. Zudem gibt es eine ganze Anzahl alternativer Produkte, die nicht auf eine (obsolete) Rechnerplattform beschränkt sind.

<sup>13</sup> Projekt-Homepage - <http://www3.informatik.uni-erlangen.de/Research/FAUmachine>, im Zuge des DBench Projects EU-gefördert. Die letzte stabile Version wurde Anfang 2006 veröffentlicht.

<sup>14</sup> Homepage und Online-Demonstration <http://www.physics.ox.ac.uk/jpc>.

<sup>15</sup> Produkt-Homepage - <http://www.parallels.com/en/products/workstation>.

<sup>16</sup> Dieses Produkt wird als "Workstation" vermarktet und liegt in Version 2.2 vor.

<sup>17</sup> Wird als "Desktop" Version 3.0 mit 3D-Grafikunterstützung beworben.

<sup>18</sup> The new Plex86 x86 Virtual Machine Project - <http://plex86.sourceforge.net>.

<sup>19</sup> Homepage <http://fabrice.bellard.free.fr/qemu>. Der Emulator kann neben der PC-Architektur eine ganze Reihe weiterer Rechnerarchitekturen emulieren.

<sup>20</sup> Diese Firma hatte sehr verschiedene Produkte, wie beispielsweise auch Webcams im Programm. Sie verkaufte Virtual PC an Microsoft.

Da die X86-Architektur immer noch aktuell und auch für die nächsten Jahre nicht mit ihrer Ablösung zu rechnen ist, spielen neben reinen Emulatoren verschiedene Virtualisierer<sup>21</sup> für die Langzeitarchivierung eine Rolle. Kommerziell gesehen, sind sie für diese Architektur bedeutend wichtiger.

**VirtualBox** ist ein recht junges X86-Virtualisierungsprodukt, welches in einer Open-Source-Basisversion und einer erweiterten, kommerziellen Variante von der Firma Innotek<sup>22</sup> vertrieben wird. Das Produkt kann in seiner Funktionalität gut mit VMware und Parallels Workstation oder Virtual PC verglichen werden. Es lässt sich auf Linux, Windows und Mac-OS X Host-Plattformen einsetzen.

**Virtual PC** führt das ursprünglich von Connectix entwickelte Produkt unter der Regie von Microsoft<sup>23</sup> für die Windows-Host-Plattform fort. Die erste Version für 32 bit-Hosts erschien 2004, eine Serverversion 2005, eine neue wenig renovierte Desktop-Revision Anfang 2007. Letztere wurde in einer 32 und 64 bit-Fassung herausgebracht, die nun Windows Vista als Host-System unterstützt. Es handelt sich dabei um einen X86-Virtualisierer, der offiziell eine Liste älterer Microsoft Betriebssysteme ablaufen lassen kann. Damit ist sowohl die Palette der Host-Systeme, als auch die der Nutzungsumgebungen vergleichsweise klein, womit dieses Produkt angesichts etlicher Alternativen für die Langzeitarchivierung weniger relevant und nur für spezielle Nutzergruppen mit eingeschränkten Anforderungen sinnvoll einsetzbar scheint. Es steht unter einer proprietären Lizenz, ist jedoch kostenfrei in seiner Nutzung.

**VMware** ist der Name einer Firma,<sup>24</sup> die seit Ende der 1990er Jahre sehr erfolgreich kommerzielle Virtualisierungssoftware für die PC-Architektur vermarktet. Das Produktspektrum wurde von der zuerst vorgestellten Workstation im Laufe der Zeit um weitere Produkte, wie den VMware-Server oder -Player, mit ähnlicher Funktionalität ergänzt. Mögliche Hostplattformen sind moderne Linux- oder Windows-Varianten und mit dem neuen Produkt "Fusion" zusätzlich neuere Apple Mac-OS X für X86-Maschinen.

### 4.3.2 Die ersten Consumer-Geräte: Home-Computer

Anders als die Entwicklung der X86-Architektur kann die Ära der Home-Computer als abgeschlossen betrachtet werden, ihre Hochphase<sup>25</sup> lag in der Zeit von Mitte der 1980er Jahre bis Anfang der 1990er. Bis auf einige Ausnahmen<sup>26</sup> werden keinerlei neue Programme für oder digitale Objekte mit diesen Maschinen mehr erstellt.

<sup>21</sup>Die Virtualisierer emulieren üblicherweise die Peripherie und reichen nur CPU-Kommandos direkt an das Host-System weiter, siehe auch Abschnitt 3.5.

<sup>22</sup>Produkt-Homepage - <http://www.innotek.de/virtualbox>.

<sup>23</sup>Produkt-Homepage - <http://www.microsoft.com/windows/products/winfamily/virtualpc/default.msp>.

<sup>24</sup>Homepage der Firma VMware - <http://www.vmware.com>, die dedizierte Produkt-Homepage findet sich im Bereich "Products".

<sup>25</sup>Siehe [CHM 2008], speziell <http://www.computerhistory.org/timeline/?category=cmptr> von 1979 an.

<sup>26</sup>In der sogenannten Demo-Szene, siehe beispielsweise Breakpoint 2007 - <http://breakpoint.untergrund.net>, werden Echtzeitkunstwerke für diverse Computerplattformen, auch noch für alte Home-Computer, geschrieben.

Eine erste Iteration der Suche nach bereits existierenden oder referenzierten Emulatoren erfolgte im Rahmen von Vorarbeiten bereits im Jahre 2004, siehe hierzu Abschnitt A.1 des Anhangs. Im folgenden sind nur eine Auswahl von Emulatoren aufgeführt, die aktuell eine gewisse Relevanz besitzen und über aktive Benutzergemeinschaften verfügen. Ein guter Startpunkt der Suche ist [EMUU 2008], welches seit 1996 online steht und sich mit dem Thema Emulatoren befasst.<sup>27</sup>

**ARAnyM** ist eine virtuelle Maschine für eine größere Anzahl verschiedener Plattformen, welche die Atari ST/TT/Falcon nachbildet. Mit ihrer Hilfe können TOS/GEM Anwendungen auf den meisten heutigen Betriebssystemen ausgeführt werden. Ziel des Open-Source-Projekts besteht darin, eine Umgebung zu schaffen, in der *"... TOS/GEM applications could continue to live forever"*.<sup>28</sup>

**Arnold** zählt zu den besten Amstrad CPC Emulatoren für Linux Plattformen. Er kann die Hardware der Amstrad Modelle CPC 464/664/6128, 464+/6128+ und die des VEB Mikroelektronik KC Compact nachbilden. Weitere Informationen zu diesem Open-Source-Emulator liefert die Projektseite.<sup>29</sup>

**Hatari/WinSTon** ist ein Atari ST Emulator für die Plattformen Linux und Windows. Der Emulator benötigt zum Ausführen von Applikationen ein installiertes TOS-Image. Er eignet sich in erster Linie zum Ausführen von Spielen.

**JaC64** ist ein Java-basierter C64 Emulator,<sup>30</sup> der im Browser als Applet ausgeführt werden kann. Dieses erlaubt eine breitere Palette an Referenzumgebungen.

**MESS** ist ein modularer Emulator für eine Vielzahl verschiedener Home-Computer-Typen. Viele Komponenten der einzelnen Plattformen, wie CPUs oder IO-Bausteine ähneln sich, so dass sie nicht für jeden Typ neu programmiert werden müssen. Dieser Emulator steht für eine Reihe von Referenzplattformen zur Verfügung.<sup>31</sup>

**UAE** der Ubiquitous Amiga Emulator steht für eine große Anzahl von Plattformen zur Verfügung.<sup>32</sup> Er bildet die Geräte Commodore Amiga 500/1000/2000 fast vollständig ab.

### 4.3.3 Konsolenspiele und Arcade-Emulatoren

In eine ganz eigene Kategorie gehören digitale Unterhaltungsgeräte. So kamen schon zu einem sehr frühen Zeitpunkt computerbasierte Spielgeräte auf, die auf Jahrmärkten oder

<sup>27</sup>Dabei hat die Seite eher Meta- und Portalcharakter. Sie pflegt eine aktuelle Sammlung externer Ressourcen. Ein deutschsprachiges Pendant findet sich unter <http://www.aep-emu.de>, [AEP 2008].

<sup>28</sup>Homepage - <http://www.arany.org>.

<sup>29</sup>Homepage - <http://arnold.emuunlim.com>, letzter Eintrag von 2004.

<sup>30</sup>Homepage - <http://www.jac64.com>, aktuellste Version von Mai 2007.

<sup>31</sup>Homepage - <http://www.mess.org>, Versionen für Windows, verschiedene Unixes und Mac-OS X.

<sup>32</sup>Homepage - <http://uae.coresystems.de>, Versionen für diverse Unixes, DOS, Windows, Mac-OS, RiscOS, BeOS und NextStep sind von verschiedenen Quellen verfügbar.

in Spielhallen aufgestellt wurden und gegen den Einwurf von Geld für eine gewisse Zeit oder Anzahl von Runden öffentlich genutzt werden konnten. Diese Geräteklasse ist eher für technische Museen oder Ausstellungen zur Altagskultur, denn klassische Bibliotheken oder Archive von Interesse.<sup>33</sup> Sie teilten oft die Grundplattform mit anderen, bereits vorhandenen Geräten, wie den Home-Computern.

Mit der zunehmenden Integration der elektronischen Komponenten wurden ultra-portable Geräte, wie der GameBoy Mitte der 1990er oder die PlayStation Portable um 2005 populär.

**MAME** ist ein Arcade-Spiele-Emulator für ältere Systeme, der oft im Zusammenhang mit MESS genannt wird. Dieses liegt an der gemeinsamen technischen Plattform beider Geräteklassen.<sup>34</sup>

**PCSX** emuliert die Sony PlayStation. Dieser Emulator ist Open Source und bildet alle Komponenten der Sony PlayStation für reguläre PC-Hardware nach. Es realisiert ein ausgefeiltes Plugin System, das leicht erweiterbar ist und weitgehend konfiguriert werden kann. Um kommerzielle Spiele ausführen zu können, benötigt man ein originales Sony PlayStation BIOS ROM.

**SNES9x** ist ein portabler, Freeware Super Nintendo Entertainment System (SNES) Emulator für Linux- und Windows-Plattformen. Mit ihm gelingt es, die meisten Spiele, die für SNES und Super Famicom Nintendo Game Systems geschrieben wurden, auf einem PC auszuführen.<sup>35</sup>

**GNUBOY** Der GameBoy Color Emulator<sup>36</sup> bildet den Color GameBoy, ein populäres Mobilgerät, in Software nach. Er ist in C programmiert und nutzt für die Ausgaben auf unix-artigen Systemen die X11 oder Framebuffer-Konsole. Für Windows und DOS wird die SDL-Schnittstelle,<sup>37</sup> wie von vielen anderen Emulatoren auch, genutzt.

**vNES** ist ein Nintendo Entertainment System (NES) Emulator, der in Java programmiert wurde.<sup>38</sup> Dieses erlaubt eine einfache Cross-plattform- und Online-Nutzung.

#### 4.3.4 Emulatoren der diversen Apple Rechner

Ende der 1980er und Anfang der 1990er Jahre spielten diverse Rechner der Computerfirma Apple im Marktgeschehen eine Rolle. Für diese Maschinen, die für eher kreative Tätigkeiten dem X86-PC vorgezogen wurden, gab es eine Reihe von Softwareprodukten, die nur

<sup>33</sup>Siehe Abschnitt A.1.3 für eine etwas breitere Darstellung und weitere Referenzen.

<sup>34</sup>Projekt-Homepage - <http://mamedev.org>.

<sup>35</sup>Projekt-Homepage - <http://www.snes9x.com>. "snes9express" ist ein grafisches Frontend, das die Benutzung des Emulators stark erleichtert.

<sup>36</sup>Siehe für einen Überblick und diverse Portierungen [AEP 2008], speziell - [http://www.aep-emu.de/Emus-file-emus\\_detail-id-947.html](http://www.aep-emu.de/Emus-file-emus_detail-id-947.html).

<sup>37</sup>Siehe hierzu Abschnitt 7.7.1.

<sup>38</sup>Homepage - <http://www.virtualnes.com>.

für diese Plattform verfügbar waren. Mit dem zwischenzeitlich sinkenden Marktanteil dieser Rechnerplattform gingen die Softwarehersteller dazu über, ihre Produkte auch für die X86-Rechnerarchitektur mit Microsoft Betriebssystem zu erstellen. Deshalb ist diese Plattform eher dann interessant, wenn Software nur für Apple Rechner verfügbar ist. Die aktuellen Apple Computer basieren nun auf der X86-Architektur. Bisher existiert jedoch noch kein kommerzieller Emulator, in dem ein Mac-OS X für X86 ausgeführt werden kann.<sup>39</sup>

**PearPC** kann die neuere Apple PowerPC Architektur emulieren. In diesem Emulator können Nutzungsumgebungen auf Basis des neueren Mac-OS X nachgebildet werden. Das Projekt<sup>40</sup> ist Open Source, leider scheint die Entwicklung eingeschlafen zu sein, da die aktuellste Version mit der Release-Nummer 0.4.0 vom 20. Dezember 2005 stammt. Das Programm lässt sich nicht sehr intuitiv verwenden und bereitet ziemlichen Aufwand, die "Panther" Version des Mac-OS X zu installieren. Eine gute Alternative besteht mit dem Multiplattformemulator QEMU.<sup>41</sup>

**Mini vMac** emuliert einen Macintosh Plus, einen der frühesten Macintosh Computer, der zwischen 1986 und 1990 verkauft wurde. Das letzte Update erfolgte Ende 2006.

**Basilisk II** ist ein Open Source Motorola 68000 Macintosh Emulator. Mit diesem kann entsprechende Mac-OS Software ausgeführt werden. Wie bei den meisten anderen Emulatoren auch, wird eine Lizenz für das Betriebssystem sowie ein ROM-Abbild der originalen Maschine benötigt. Der Emulator selbst steht unter der GNU General Public License (GPL).

### 4.3.5 Emulatorensuche zu sonstigen Architekturen

Durch den ausreichenden zeitlichen und technologischen Abstand gelingt es, die ehemals für kleine Computer unvorstellbare Rechenleistung und Speicherkapazität der Großrechner auf heutigen Systemen innerhalb eines Emulators ablaufen zu lassen. Eine größere Herausforderung bei diesen Systemen liegt eher in der geeigneten Emulation der Speichermedien, Lochkarte und -streifen.

**Hercules** ist ein Open Source Emulator<sup>42</sup> für die IBM Mainframe S/360, S/370, ESA/390 und z/Architecture Systeme. Er kann auf Linux, Mac-OS und Windows Host-Plattformen eingesetzt werden. Die derzeit aktuelle Version trägt die Bezeichnung 3.05 und kann gegenüber ihren Vorgängern auch die Z-Series emulieren. Wie schon beim realen Vorbild handelt es sich beim Emulator um ein recht komplexes System, welches sinnvoll nur mit umfangreichem Hintergrundwissen betrieben werden kann. Anders als bei Home-Computern oder X86-PCs

<sup>39</sup>Die Unterschiede zum klassischen PC sind nicht groß, jedoch setzt Apple auf das neuere EFI, welches von Microsoft Vista erst in späteren Versionen unterstützt werden soll.

<sup>40</sup>Siehe hierzu die Projekt-Homepage unter <http://pearpc.sourceforge.net>.

<sup>41</sup>Siehe hierzu Abschnitt 5.6.1.

<sup>42</sup>Siehe hierzu die Projekt-Homepage - <http://www.hercules-390.org>. Die Lizenz ist dort mit der Q Public License angegeben.

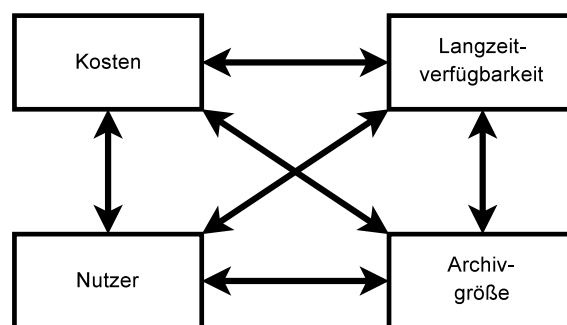
war die Zahl der direkten Anwender an diesen Plattformen deutlich beschränkter. Für virtuelle Netzwerkverbindungen nutzt Hercules das TUN/TAP-Interface des Host-Systems, welches auch durch etliche andere Emulatoren eingebunden wird. Virtuelle Festplatten sind Dateien im Host-System. Die Steuerung des Emulators erfolgt mittels einer Konsole und wäre damit für Automatisierungen zugänglich.

**The Computer History Simulation Project** ist ein Emulator, der eine ganze Reihe früherer Großrechnersysteme in einem Projekt<sup>43</sup> zusammenfasst. Ein Überblick zu den unterstützten Systemen findet sich in der Rechercheliste im Anhang, Abschnitt A.1.

Neben den genannten Emulatoren existiert eine Vielzahl weiterer, spezieller Programme für bestimmte Anwendungen. Einige interessante Ansätze sind im Anhang (Abschnitt A.1.4) der Vollständigkeit halber kurz aufgeführt.

## 4.4 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 Ablaufumgebung<sup>44</sup> steigert, andererseits dies die Kosten des Archivbetriebs in die Höhe treibt. Metadaten und die Auswahlkriterien von Emulatoren



**Abbildung 4.3:** 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.

oder Virtualisierern 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 (Kapitel 7) 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. Charakteristika von

<sup>43</sup>Weitere Informationen liefert die Projekt-Homepage [CHSP 2007] oder auch [Burnet und Supnik 1996].

<sup>44</sup>Siehe hierzu auch das spätere Kapitel 6 zu verschiedenen View-Paths.

Metadaten<sup>45</sup> bilden den ersten Teil der im folgenden präsentierten Auswahlkriterien. Meistens können sie nicht alle Kriterien sinnvoll abdecken. Oder die Aufnahme sämtlicher denkbarer Auswahlkriterien würde die gängigen Metadatenschemata unnötig aufblähen und die Kosten ihrer Pflege erhöhen.

Emulatoren präsentieren sich auf der ausführenden Host-Plattform als normale Applikation, die einerseits bestimmte Anforderungen an ihre Umgebung stellt, andererseits der jeweiligen Benutzergruppe eine Schnittstelle zur Erstellungsumgebung des gewünschten Objekts liefert.

#### 4.4.1 Auswahlkriterien für die digitale Langzeitarchivierung

Die klassischen Metadaten digitaler Objekte allein genügen nicht, um anhand ihrer eine qualifizierte Auswahl von Emulatoren oder Virtualisierern für die Langzeitarchivierung zu treffen. 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 Kapitel 6 eingeführt werden, erreichen.

Die Liste der Auswahlkriterien lässt sich in nachstehende Kategorien vorgruppieren:

- Allgemeine Metadaten, wie Name, Hersteller oder Projekt, Homepage im Internet und Historie der letzten Änderungen, Abhängigkeiten von der Host-Plattform
- Benutzergruppen,<sup>46</sup> Darstellung einzelner Objekte, Fähigkeiten im Umgang mit Objekten, Frequenzen des Zugriffs auf Objekte in ihren Langzeitarchiven. So kann es unterschiedliche Ziele zum Einsatz von Emulation geben: Wenn es darauf ankommt ein bestimmtes Quark Express Dokument wieder sichtbar zu machen, braucht man unter Umständen dafür nicht unbedingt einen Apple-Macintosh-Emulator, wenn das Programm auch in einer Variante für Microsoft Windows zur Verfügung stand und dieser Weg bereits im Archiv gesichert ist. Interessiert jedoch das Aussehen einer populären Applikation zu einem bestimmten Zeitpunkt in ihrer damals typischen Umgebung, kann es von Interesse sein dieses "Look and Feel" durch einen zusätzlichen View-Path abzubilden.<sup>47</sup>
- 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 Host-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

<sup>45</sup>Vgl. hierzu das Dublin Core Set von Metadaten aus 3.3.1.

<sup>46</sup>Institutionen und Gedächtnisorganisationen haben unterschiedliche Ansprüche an Fristen der Objektarchivierung.

<sup>47</sup>In diesen Fällen wird eine Applikation oder ein Betriebssystem aus Sicht des Archivbetriebs vom sekundären Objekt zum primären.

die Hardwareparameter der Emulation eines Amiga 500 deutlich beschränkter als die große Varianz der X86-PCs.

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

#### 4.4.2 Generelle Informationen

Generelle Informationen unterscheiden sich vom Charakter her nicht von klassischen Archivdaten: So wie man den Autor, Titel und Verlag eines Buches ablegt, sollten ebensolche Basisdaten zu den Emulatoren erhoben werden. Sie bilden das Grundset der Metadaten.

- Name - Name oder Bezeichnung des Programms.
- Version - Wie von Büchern bekannt, erleben populäre Programme typischerweise mehr als eine Auflage. Damit verschiedene Ausgaben, die sich oft durch Funktionsumfang und Anforderung an ihre Umgebung deutlich unterscheiden, sich sinnvoll auseinanderhalten lassen, sollte eine Versionsnummer am besten in Kombination mit dem Erscheinungsdatum vermerkt werden.<sup>48</sup>
- Beschreibung - Sie erfolgt in kurzer, menschen-lesbarer Form, um eine schnelle Orientierung beim Suchen zu erlauben.
- Ersteller - Die Firma oder das Projekt sollte diesen Datensatz enthalten. Zusätzlich könnten Informationen zur Webseite (Lokation) beziehungsweise Kontaktdaten abgelegt werden.
- Lokation - Ortsangabe des Herstellers oder des Projekts, typischerweise im Internet durch Angabe eines URI (Uniform Resource Identifier). Diese Ortsangabe bietet typischerweise die Möglichkeit zum Bezug der Software in elektronischer Form (Download) oder einen Verweis auf geeignete Quellen.
- Benutzerschnittstelle - Dieses Feld liefert überblicksmäßig Angaben zur Steuerung eines Programms. So lassen sich einige Emulatoren nur über die Kommandozeile oder nur über ein grafisches Benutzer-Interface und Konfigurationsdateien in ihrer Ausführung steuern. Dieses könnte für eine spätere automatisierte Erstellung von View-Paths von Interesse sein.
- Sprache - Das passende Benutzer-Interfaces hilft späteren Anwendern eines Programms mit diesem umzugehen. Diese sollten in der Lage sein, das Werkzeug steuern und Fehler- sowie Hinweismeldungen interpretieren zu können. Moderne Programme bieten oft die Unterstützung der Lokalisierung in etliche Landessprachen an.

<sup>48</sup>Gerade bei Open-Source-Emulatoren gibt es viele Zwischen-Releases, die oft nicht mit einer speziellen Bezeichnung versehen sind. Teilweise findet man hier die Revisionsnummer der Versionsverwaltung als Orientierung.



- Anleitungen - Traditionellerweise werden mit in Boxen verkauften Programmen Benutzerhandbücher ausgeliefert. Dieses geschieht inzwischen zunehmend in elektronischer Form, was die Archivierung erleichtert. Diese Handbücher sollten Hilfestellung bei der Installation und der Benutzung des Programms bieten. Dieses wird mit zunehmendem Abstand zum Erscheinungsdatum umso wichtiger.

## 4.5 Benutzergruppen und institutionelle Rahmenbedingungen

Die Anforderungen an Emulatoren oder Virtualisierer können je nach Benutzergruppe oder Institution sehr verschieden ausfallen. Dadurch können durchaus andere Ergebnisse der Tool-Auswahl resultieren.

Während in der Gruppe der Privat- und Hobbyanwender eher persönliche Interessen und Vorlieben Archivierungsziele bestimmen, so sind die meisten Institutionen, wie kommerzielle Unternehmungen, Behörden und Administrationen sowie Bibliotheken, Archive und Forschungseinrichtungen in Regelungswerke eingebunden. Diese teilweise in Gesetze gegossenen Vorgaben bestimmen wesentlich über die Ausrichtung des jeweiligen digitalen Langzeitarchivs mit.

So bestimmt das "Gesetz über die Deutsche Nationalbibliothek" den Sammelauftrag der größten deutschen Bibliothek<sup>49</sup> und erweitert ihn um Internet-Veröffentlichungen. Einen generellen Überblick zu rechtlichen Fragen im Umfeld der Langzeitarchivierung geben [Beger 2005] oder auch [Goebel und Scheller 2004], die im Auftrag von Nestor tätig wurden.

### 4.5.1 Archive

Nationale und allgemeine Archive staatlicher oder öffentlicher Institutionen haben die Aufgabe, ausgewählte oder sämtliche Objekte des Geschäftsverkehrs ihrer beauftragenden Institutionen für bestimmte gesetzlich befristete Zeiträume oder unbefristet aufzuheben. Bei diesem Auftrag geht es üblicherweise um ein sehr eng definiertes Feld verschiedener Datentypen, da oft mit standardisierten Austauschverfahren innerhalb der Institutionen gearbeitet wird. Vielfach können Archive in gewissem Umfang administrativ auf ihren Input einwirken. Der überwiegende Anteil der Daten wird aus statischem Material bestehen, welches sich für automatische Migrationsprozesse eignet oder für solche vorbereitet wurde.

Eine wichtige Aufgabe dieser Archive wird es daher sein, die Authentizität ihrer Archivobjekte zu jedem Zeitpunkt nachweisen zu können.<sup>50</sup>

### 4.5.2 National-, Universitäts- und Allgemeinbibliotheken

Bibliotheken obliegt seit je her die Aufgabe eine breite Palette an Objekten des kulturellen, wissenschaftlichen und wirtschaftlichen Schaffens, meistens in Form von Texten und Bildern,

<sup>49</sup>Siehe [DNB 2006] oder auch die Pressemitteilung zur Neufassung - [http://www.ddb.de/aktuell/presse/-pressemitt\\_dnbg\\_neu.htm](http://www.ddb.de/aktuell/presse/-pressemitt_dnbg_neu.htm).

<sup>50</sup>Siehe hierzu auch die Untersuchungen von [van Diessen und van der Werf-Davelaar 2002] im Rahmen des DIAS-Projekts.

zu sammeln. Mit der digitalen Revolution werden die gedruckten Werke, die die letzten Jahrhunderte dominierten, durch eine zunehmende Anzahl digitaler Objekte ergänzt.

Während viele Neuerscheinungen teilweise nur noch digital erfolgen oder in digitaler Form wegen des großen Recherchevorteils beschafft werden, kommen weitere Objekte hinzu, die aus der Digitalisierung bereits vorhandener Bestände stammen.<sup>51</sup>

### 4.5.3 Forschungseinrichtungen

Ein großer Teil der Forschung der letzten 50 Jahre fand unter Einsatz von Computern statt und wird sich in Zukunft eher noch ausweiten. Im Zuge der verschiedenen Arbeiten erfolgte eine Erhebung oder Generierung von Daten, die in später veröffentlichte Forschungsergebnisse einfließen. Jedoch wurde lange Zeit nur die Arbeit selbst, aber selten die gewonnenen wissenschaftlichen Primärdaten archiviert. Hierdurch gingen teilweise wertvolle Daten verloren, die für spätere Forschungsprojekte hätten relevant sein können. Ebenso lassen sich eingereichte und veröffentlichte Ergebnisse nur schwer überprüfen oder reproduzieren, wenn die Primärdaten nicht mehr vorliegen.

Nicht umsonst hat die DFG die Forderung aufgestellt, dass eine mindestens zehnjährige Frist zur Aufbewahrung von Messdaten und wissenschaftlichen Erzeugnissen gelten soll. Auf diese Weise soll nicht nur dem wissenschaftlichen Betrug vorgebeugt, sondern wertvolle Daten für eine weitere Verwendung aufbewahrt werden. Die MPG beteiligt sich zudem an einem Forschungsprojekt zur Langzeitarchivierung wissenschaftlicher Primärdaten.<sup>52</sup>

### 4.5.4 Computer-, Kunst- und technische Museen

Durchaus sehr andere Anforderungen als Archive und Bibliotheken haben Institutionen zur Bewahrung von Alltagskultur, Geschichte oder Kunstwerken. Sie stellen Rechnerplattformen als Dinge des täglichen Lebens aus, die in einer bestimmten Epoche relevant waren. Dabei geht es einerseits um den optischen und haptischen Eindruck der ursprünglichen Hardware, aber auch um das Aussehen und die Funktion der jeweils verwendeten Software.

Digitale Objekte von Interesse sind für diese Institutionen Applikationen und Betriebssysteme, aber nicht so sehr die üblichen Primärobjekte der bibliothekarischen Langzeitarchive. Trotzdem kann es auch für Museen Aufgabe oder Verpflichtung sein, bei der "Archäologie" unbekannter digitaler Objekte zu helfen und diese wieder sichtbar zu machen.

Vielfach lebt das Interesse an einem Museum oder die Begeisterung für eine Ausstellung vom Mitmachfaktor. Emulatoren der ausgestellten Computer sollten deshalb über einfach zu bedienende Benutzer-Interfaces verfügen.

Da zu den Computern eine ganze Reihe verschiedener Peripherie dazugehört, besteht sicherlich teilweise das Bedürfnis, diese auch an aktueller Hardware betreiben zu können. Dann sind serielle, parallele oder USB-Schnittstellen eines Emulators zum Host-System von Interesse. Jedoch gilt auch hier, dass sich aktuelle Hardware und Museumsstücke immer weiter auseinanderbewegen werden und dann irgendwann auch keine geeigneten gemeinsamen

<sup>51</sup>Siehe hierzu auch Abschnitt 3.1.2.

<sup>52</sup>PARSE.Insight - Permanent Access to the Records of Science [APA 2008], Mitglied sind beispielsweise die Deutsche Nationalbibliothek [DNB 2008] und die Max Planck Gesellschaft (MPG).

Schnittstellen mehr besitzen. Ebenso gilt dieses für den Ausfall der Hardware, der im Fall der Rechner selbst durch den Einsatz von Emulatoren ausgeglichen werden kann. Dieses ist für Peripherie wenig realistisch.

Durch den Verzicht darauf, ein großes Hardwaremuseum permanent betriebsfähig zu halten, lassen sich mithilfe des Emulatoreinsatzes Kosten reduzieren. Diese könnten mit anwachsendem Museumsbestand irgendwann weder personell noch finanziell geleistet werden.<sup>53</sup>

### 4.5.5 Unternehmen und Organisationen

Für Unternehmen und diverse Institutionen gelten für alle Geschäftsvorgänge Aufbewahrungsrichtlinien unterschiedlicher Zeitdauer. Sie sind verpflichtet, von wichtigen Vorgängen, Dokumenten, Geschäfts- und anderen Berichten sowie Korrespondenzen die Originaldokumente oder Kopien aufzubewahren und geeignet abzulegen. Teilweise heben Firmen ihre geschäftskritischen Daten auch deutlich länger als die gesetzlich vorgeschriebene Periode auf. Wegen der im Verhältnis zu anderen Nutzern beschränkten Zeithorizonte genügen in vielen Fällen sicherlich einfachere Langzeitarchivierungsstrategien. So könnte es bereits ausreichen, mit den derzeitigen Virtualisierungstechniken die Historie der hauptsächlich eingesetzten X86-Architektur auf einem einzigen Host-System mit Linux beziehungsweise Windows XP oder Vista zusammenzufassen.



**Abbildung 4.4:** DOS-Spiel (Doom von ID-Games) mit funktionierender Soundausgabe aus dem Jahr 1993 in der DOSBOX unter Linux/X11 mit einem 2.6er Kernel.

<sup>53</sup>Vgl. hierzu den Abschnitt 2.5 und [Rothenberg 2002], S. 12.

### 4.5.6 Privatpersonen

Für Privatpersonen stellt sich die Lage der Langzeitbewahrung digitaler Objekte noch einmal anders dar: Sie stehen vor dem wachsenden Problem, wie sie ihre digital aufgenommenen Bilder und Filme, sowie ihr privates Medienarchiv aus kommerziellen Inhalten langfristig zugreifbar halten können. Hierzu zählen Fotos, private und gekaufte Videos, CD- oder MP3-Sammlungen. Daneben existieren in vielen Haushalten Sammlungen privater Emails oder Textdokumente. Viele Computernutzer möchten zudem Computerspiele ihrer früheren Systeme (Abb. 4.4, S. 95) oder Game-Konsolen weiterhin verwenden können. Dabei sollte der spätere Zugriff auf den größten Teil dieser Objekte auch für weniger technik-affine Personengruppen möglich sein.

Während für die persönlichen Daten privater Computernutzer keine Aufbewahrungsrichtlinien bestehen, sieht dieses im Feld der elektronischen Kommunikation anders aus. So werden inzwischen viele Rechnungen von diversen Dienstleistungsanbietern nur noch in elektronischer Form verschickt. Verwaltungsakte, wie die Abgabe der eigenen Steuererklärung, sind zunehmend in elektronischer Form erwünscht. In diesem Bereich ergeben sich neue Problem- und Fragestellungen.

## 4.6 Nichttechnische Auswahlkriterien

Für das spätere Archivmanagement spielen eine ganze Reihe nicht-technischer Kriterien eine Rolle. Hierzu zählen die Art der Software, existierende Benutzergemeinschaften, Reife, Produktvorgeschichte und Weiterentwicklung sowie Kostenaspekte. Gerade letztere können sich nicht unerheblich auf den längerfristigen Betrieb eines Archivs auswirken.

### 4.6.1 Kommerzielle oder Open Source Software

Software lässt sich aus Sicht der Anwender grob in zwei Kategorien einteilen, wobei diverse Zwischenformen zu finden sind. Auf der einen Seite werden kommerzielle Produkte von Firmen oder freischaffenden Programmierern angeboten, andererseits existieren Gemeinschaften von Softwareentwicklern, die gemeinsam über einen längeren Zeitraum durchaus mit wechselnden Teilnehmern an einem Projekt arbeiten.

**Kommerzielle Software** zeichnet sich üblicherweise dadurch aus, dass sie von einer Firma auf dem Markt, verknüpft mit bestimmten Leistungen, angeboten wird. Hierzu zählen beispielsweise Vereinbarungen zu Hilfestellungen und Fehlerbehebung. Durch Support-Verträge können mit einer Langzeitarchivierung beauftragte Institutionen versuchen, eine langfristige Absicherung zu erreichen, jedoch gibt es bisher wenig Beispiele für Software, für die eine Unterstützung in Zeiträumen angeboten wurde, der den geplanten Perioden einer Langzeitarchivierung entspricht.

Applikationen erfahren eine regelmäßige Produktpflege beispielsweise durch Updates, die einen Weiterbetrieb der Software in sich wandelnden Host-Plattformen erlauben. Jedoch kommt jede Software an das Ende ihres Produktzyklus, sei es durch Obsoleszenz, Ablösung

durch ein Nachfolgeprodukt oder Einstellung, weil es sich nicht mehr lohnt oder Konkurs beziehungsweise Verkauf eines Anbieters.

Kommerzielle Software wird mittlerweile durch immer aufwändigere Verfahren vor unerlaubter Vervielfältigung und Verbreitung<sup>54</sup> geprägt. Dieses kann ein Schutz am Installationsdatenträger sein oder die Notwendigkeit von Netzwerkverbindungen zum Hersteller implizieren, der bei jedem Start des Produkts die Gültigkeit der Lizenz überprüft. Oder es werden zeitlich beschränkte Laufzeit- oder Evaluationslizenzen angeboten. Solche Lizenzen können das langfristige Funktionieren von Software gefährden. Denn wenn aus den bereits genannten Gründen der ursprüngliche Hersteller nicht mehr zur Verfügung steht, kann es passieren, dass das Produkt nicht aufgrund fehlender geeigneter Ablaufumgebungen, sondern schlicht wegen des Kopierschutzes nicht mehr nutzbar wird. Vorhersagen zu einem solchen Verhalten von Programmen sind oft schwer zu machen, da sich viele Hersteller bezüglich der Verfahren zur Absicherung ihrer Lizenzen bedeckt halten. Besonders in die Diskussion geraten ist in diesem Zusammenhang das Digital Rights Management (DRM).<sup>55</sup> Für die Verwendung in einem Langzeitarchiv spielt die Freiheit der eingesetzten Software von objektgebundenen Lizenzsicherungen und DRM eine nicht unerhebliche Rolle für eine risikoärmere Anwendung.

Darüber hinaus ist die Art der Lizenz nicht unwesentlich. Es existieren verschiedene Modelle, die sich auf Benutzerzahlen, Institutionen, Organisationen oder zeitliche Befristungen beziehen können.

Der Quellcode kommerzieller Applikationen ist in den wenigsten Fällen vom Hersteller erhältlich, da dieser in der Nichtfreigabe den besten Schutz vor unerlaubter Vervielfältigung oder unauthorisierter Nutzung sieht.

**Open Source Software** repräsentiert ein alternatives Lizenzierungsmodell. Anders als bei kommerzieller Software kann auf den Quellcode jederzeit zugegriffen und dieser verändert und verbessert werden. Formale Support-Kontrakte lassen sich zu Open-Source-Produkten selten mit den Erstellern der Software abschließen, jedoch entwickelt sich inzwischen eine vielfältige Servicelandschaft. Während kommerzielle Applikationen oft durch einen klaren Produktzyklus gekennzeichnet sind, lassen sich Vorhersagen zur langfristigen Verfügbarkeit und Weiterentwicklung bei Open-Source-Projekten schwerer treffen. Andererseits können sie durch eigenen Einsatz durchaus mitgesteuert und dadurch qualitativ hochwertiger und besser werden.

Unabhängig davon kann es für Institutionen interessant sein, ihre ursprünglich proprietär entwickelten Programme unter einer Open-Source-Lizenz bereitzustellen. Wenn sich für das Projekt eine interessierte Benutzergemeinschaft findet, lassen sich die Kosten der Weiterentwicklung senken und das Anwendungswissen verbreitern. Ein Beispiel hierfür ist der Java-basierte X86-Emulator "Dioscuri".<sup>56</sup> Offene Lizenzen helfen bei der Standardisierung, da keine abschreckenden Kosten für eine Benutzung entstehen und sinnvolle Erweiterungen und Ergänzungen von vielen Seiten vorgenommen werden können. Gerade für Organisationen ohne Vermarktungsinteresse übertreffen die Nebenkosten einer Kommerzialisierung oft die erzielbaren Einnahmen. Zudem kollidieren diskriminierungsfreie Standards und Verfahren weniger mit dem öffentlichen Auftrag von Gedächtnisorganisationen.

<sup>54</sup>Ausführliche Anmerkungen zu dieser Problematik und Konsequenzen erörtert Abschnitt 7.11.

<sup>55</sup>Ebd.

<sup>56</sup>Siehe [van der Hoeven u. a. 2005], die Weiterentwicklung erfolgt im Zuge des PLANETS-Projekts [PLANETS 2008].

**Abwägungen und Konsequenzen** Vorhersagen zur Langzeitverfügbarkeit für beide Arten von Software lassen sich mangels zu kurzer Perioden bisher nur schwer machen. Jedoch zeichnet sich gerade im Bereich der kommerziellen Software ein sich mit zunehmendem Alter verschärfender Widerspruch aus ökonomischem Erfolg und der langfristigen Erhaltung eines Produkts ab. In dem Augenblick wo ein Hersteller das Interesse an seinem Produkt verliert oder die Weiterentwicklung oder Anpassung an diesem aus wirtschaftlichen Zwängen heraus einstellen muss, gibt es keine automatische Folgeregelung. Das Produkt und sein Quellcode befinden sich im Besitz des Herstellers und können nicht durch Dritte übernommen werden.

Bei Open Source liegt die Fragestellung anders. Auch in diesem Bereich werden viele Projekte aus Zeitmangel, Verlust des Interesses seitens der Entwickler oder Nutzer oder durch äußere Einflüsse, wie die Änderung der Rechnerplattform, aufgegeben. Jedoch liegt der Quellcode weiterhin vor und bei entsprechendem Bedarf kann unter Inkaufnahme gesteigerter Kosten der Weiterbetrieb der Software gesichert werden – entweder langfristig oder bis eine geeignete Nachfolgelösung gefunden werden kann.

Für beide Softwarearten hilft ein Blick auf die Hersteller-, Anbieter oder Projekt-Homepage, um einen Eindruck zu bekommen, wie lange die Software schon verfügbar ist, welche Reife sie dabei erlangt hat, in welchen Abständen neue Versionen zur Verfügung gestellt wurden und ob es eine aktive Anwendergemeinde gibt. Für die Einschätzung der Nutzbarkeit eines Werkzeugs spielt die Qualität der Produktunterstützung eine Rolle: Wie lange existiert es schon, gibt es regelmäßige Updates oder zumindest Bugfixes, wird es an neue Umgebungen (zeitnah) angepasst, wenn diese erscheinen? Wie sieht es mit der generellen Qualität aus, wird das Produkt von einer großen Firma oder Organisation angeboten und betreut und wie ist deren Zuverlässigkeit? Sind sie langfristig oder eher sehr kurzfristig orientiert?

#### 4.6.2 Weitere Kriterien

Darüber hinausgehende Auswahlkriterien lassen sich anhand der folgenden Liste zusammenstellen. Viele dieser Punkte können schwer absolut sondern eher nur empirisch mit einem gewissen Zeitaufwand bestimmt werden.

- **Tauglichkeit** - Es ist zu klären, ob ein Emulator lediglich eine ganz bestimmte Nutzungsumgebung oder eine ganze Reihe verschiedener Hardwareplattformen genutzt werden kann.
- **Nutzbarkeit** - Wie intuitiv lässt sich eine Software bedienen, für welche Einsatzszenarien ist sie geeignet, wo liegen ihre Schwächen und Nachteile, worin ihre Beschränkungen in der bestehenden Implementation. Wie leicht kann sie installiert werden?
- **Zuverlässigkeit** - Wie gut ist die Stabilität des Programms, wie häufig treten Fehler, welchen Schweregrads, auf?
- **Performance** - Sie kann ein Entscheidungskriterium sein, wenn beispielsweise mit vielen parallelen Anfragen oder gleichzeitigen Benutzern auf einem System gerechnet wird. Je effektiver eine Software arbeitet, desto geringer sind die Kosten der vorzuhaltenden Referenzplattform.

- Portabilität - Diese hängt zum Teil mit der Art der Software zusammen, wie sie im vorangegangenen Abschnitt erörtert wurde. Hier spielen jedoch ebenso Fragen der Entwicklungsumgebung aus Programmiersprache, Kompilern und notwendigen APIs und Softwarebibliotheken eine Rolle. Sie ist eine der Erfolgsbedingungen des Einsatzes von Emulatoren für die Langzeitarchivierung.
- Programmiersprache - Sie bestimmt wesentlich über die langfristige Erhaltbarkeit mit und kann deshalb auch als eigenes Kriterium gegenüber dem letzten Punkt aufgestellt werden.
- Relevanz, Marktdurchdringung - Beide können wichtige Kriterien sein, wenn man davon ausgeht, dass verbreitetere Produkte über größere Anwendergemeinschaften und damit besseres und umfangreicheres Wissen verfügen.

### 4.6.3 Ökonomische Erwägungen

Es lassen sich eine ganze Reihe direkter und indirekter Kostenfaktoren beim Einsatz von Emulations- oder Virtualisierungssoftware identifizieren. Mit ihnen muss sich das Management eines Archivs auseinandersetzen. Zu den direkten Kosten zählen die Aufwendungen für die Anschaffung und regelmässige Ausgaben für Wartung, Support und Updates. Einige Lizenzmodelle unterscheiden nach der Anzahl der parallelen Benutzer. Die direkten Kosten multiplizieren sich dann mit der Zahl der parallel gepflegten Installationen. Eine größere Menge verschiedener Emulatoren oder Virtualisierer erhöht die Wahrscheinlichkeit des Langzeitzugriffs durch Vervielfachung verfügbarer View-Paths, lässt dabei jedoch die Gesamtkosten<sup>57</sup> des Archivs steigen.

Zu den direkten Kosten addieren sich die indirekten Kosten, beispielsweise für den Personalaufwand und vorzunehmende Anpassungen oder Migrationen an den einzelnen Softwareprodukten. Hierüber bestimmen die geplanten Zeiträume der Aufbewahrung der Primärobjekte. Dabei fällt nicht so sehr ihre Menge an sich ins Gewicht, sondern die Anzahl der unterschiedlichen Objekttypen.

## 4.7 Technische Auswahlkriterien

Emulatoren sollen die sich zunehmend öffnende Lücke zwischen Primärobjekten samt ihren Ablaufumgebungen und der jeweils aktuellen Referenzplattform überbrücken. Sie laufen als Applikationen auf einem Host-System und stellen als solche Anforderungen an ihre Umgebung. Umgekehrt generieren Applikationen sowie eventuell zu deren Ablauf notwendige Betriebssysteme ihrerseits Bedingungen an ihre Umgebung, die ein Emulator bereitstellen muss.

### 4.7.1 Anforderungen an das Host-System

Der Emulator oder Virtualisierer als Applikation im Host-System stellt bestimmte Bedingungen an Betriebssystem und Rechnerarchitektur. Diese Anforderungen können bereits teilweise in den Metadaten zum digitalen Objekt "Emulator" kodiert sein. Zu ihnen zählen:

<sup>57</sup>Vgl. hierzu den Abschnitt 6.4.1 zu erweiterten und verallgemeinerten Erwägungen.

- Betriebssystem - mindestens bestimmt durch Name und Version.
- Bibliotheken - stellen diverse APIs bereit, ebenso wie Schnittstellen zu Benutzer-Interfaces, textbasiert oder für grafisch orientierte Ausgaben, Anforderungen an Systembibliotheken und weitere Softwarekomponenten, wie beispielsweise Hilfsprogramme im Host-System.
- Hardwareanforderungen - Rechnerarchitektur mit Angaben zu minimalem Speicherbedarf, optimale Speicherausstattung, Mindestanforderungen an die CPU - bezogen auf Geschwindigkeit und Funktionsumfang, Anforderungen zu Festspeicherbedarf.
- Infrastruktur - Wege des Datenaustausches, Möglichkeit mehrere Instanzen eines Emulators ablaufen zu lassen.
- Konfiguration - Einrichtung und Steuerung des Emulators aus Sicht des Host-Systems durch APIs, Konfigurationsdateien, Skripting-Fähigkeiten.
- Sicherheit - Nutzungsumgebungen sollten strikt abgeschottet von der Referenzumgebung laufen.

#### 4.7.2 Datenim- und -export, Interaktion mit dem Host-System

Nur in Ausnahmefällen tritt der Emulator dem Nutzer als Primärobjekt entgegen, weshalb ein Datenaustausch mit der Referenzumgebung entfällt. Für den Normalfall müssen Wege des Transports der Zielobjekte in die Emulationsumgebung realisiert werden. Zudem werden Situationen auftreten, in denen ein umgekehrter Datenaustausch, beispielsweise von modifizierten Zielobjekten zurück in das Host-System, erforderlich wird. Verschiedene Varianten können je nach Emulator und Host-System zur Verfügung stehen:

- Containerdateien als primäre virtuelle Datenträger lassen sich auch ohne gestarteten Emulator direkt mit Werkzeugen im Host-System bearbeiten.
- Virtuelle Wechseldatenträger, wie Diskettenabbilder sowie ISO-Images können mit Werkzeugen im Host-System vorbereitet und später an den startende oder laufende Emulatoren gebunden werden.
- Austausch mittels virtueller Netzwerke zwischen Gast und Host. Ein Spezialfall können sogenannte gemeinsam genutzte Verzeichnisse "Shared Folders" sein. Sollte ein Datentransport über Weitverkehrsnetze realisiert werden, müssen Überlegungen und Vorkehrungen zur Datensicherheit vorhanden sein oder geeignet getroffen werden.
- Datenaustausch per "Copy & Paste" durch spezielle APIs zwischen Emulator beziehungsweise Virtualisierer und Host.
- Datenaustausch mittels serieller, paralleler oder USB-Schnittstellen wird vermutlich nur für sehr wenige Szenarien interessant sein.



Für die überwiegenden Datenaustauschkanäle sind spezielle Werkzeuge und eine Sammlung von Treibern, beispielsweise für Netzwerkprotokolle oder Dateisysteme, erforderlich. Ein Teil dieser zusätzlichen Objekte gehört üblicherweise nicht zum Software- oder Lieferumfang eines Emulators oder Virtualisierers. Sie müssen gesondert berücksichtigt werden. Die Notwendigkeit oder Abhängigkeit von weiteren Sekundärobjekten sollte sich am besten in den Metadaten zum Emulator widerspiegeln.

### 4.7.3 Anforderungen seitens der Nutzungsumgebung

Die wiederherzustellende Ablaufumgebung kann je nach Rechnerarchitektur bestimmte Anforderungen an die Emulation stellen. So sollten geforderte Bildschirmauflösungen, Farbtiefen oder 3D-Fähigkeit realisierbar sein.

Ebenso muss für bestimmte Funktionalität eine passende Hardware bereitgestellt werden. In den meisten Fällen wird es sich dabei um rein virtuelle Hardware, wie Netzwerkkarten oder Datenträger handeln. Während bei frühen Rechnerarchitekturen die Hardwareauswahl sehr beschränkt war und wenig Varianz aufwies, liegt dieses bei der X86-Architektur anders. Deshalb sind für diese Architektur sinnvollerweise weitere Angaben zu machen, die die Weiterentwicklungen beispielsweise seitens der Betriebssysteme widerspiegeln:

- Mainboard, Speicher, BIOS und Systemuhr sind die Grundkomponenten einer Rechnerarchitektur neben der CPU und fassen die Eigenschaften der Hardware mit ihrer zum Betrieb notwendigen Firmware zusammen. Diese Komponenten sind für frühe Architekturen sehr fix. Deren Firmware (von Betriebssystemen kann bei vielen Home-Computern kaum gesprochen werden) oder Betriebssysteme sind für die jeweilige Hardware kompiliert. Die X86-Architektur erfordert seitens des Betriebssystems meist zusätzliche Software, wie Hardwaretreiber. Der im Emulator bereitgestellte Speicher für die Nutzungsumgebung muss zusätzlich zum Bedarf des Emulators selbst bereitgestellt werden. Viele Anwendungen benötigen eine weitgehend genaue Systemuhr oder einen exakten Prozessortakt für eine korrekte Funktion. Andernfalls stimmt beispielsweise die Spielgeschwindigkeit oder die Aktualisierung von Bildschirminhalten nicht mehr mit dem Original überein.
- Tastatur und Maus gehören zu den zentralen Elementen der Benutzereingabe. Geräte wie Joysticks oder Grafiktablets sind ebenso denkbar, jedoch deutlich seltener geeignet in die Nutzungsumgebung abzubilden. Für die Tastatureingabe kommt es auf die geeignete Übersetzung der Tasten-Codes für die Nutzungsumgebung an. Dabei kann es passieren, dass bestimmte Tasten-Codes bereits im Host-System abgefangen werden, da sie dort eine besondere Bedeutung haben. Dann muss eine geeignete Ersatzeingabe bereitstehen. Dieses gilt ebenso für Rechnerplattformen, die mit zusätzlichen Modifikatortasten gearbeitet haben, welche im Host-System nicht vorhanden sind.<sup>58</sup> Der Fall liegt bei Mauseingaben einfacher, jedoch kann die Zahl der Tasten der Host-Maus kleiner oder anders belegt sein, als für die Nutzungsumgebung erforderlich oder vom Anwender erwartet.

---

<sup>58</sup>Siehe zu den speziellen Herausforderungen der Benutzerinteraktion Abschnitt 6.5.4.

- **Datenträger** der jeweils unterstützten Form. Hierzu zählen Disketten, Festplatten, CD-ROMs, DVDs oder Bandlaufwerke, die eine virtuelle Entsprechung benötigen, damit die Nutzungsumgebung auf die jeweils erwartete Peripherie zum Lesen und Speichern ihrer Daten trifft. Die genannten Laufwerke können je nach Plattform über unterschiedliche Busse, wie beispielsweise SCSI oder IDE/ATAPI angeschlossen sein.
- **Grafikadapter.** Die meisten Endbenutzersysteme verfügten über Bildschirmausgabesysteme, die in geeigneter Form nachgebildet werden müssen. Hierbei erwartet die jeweilige Nutzungsumgebung bestimmte Ausgabestandards, wie Hercules, CGA, EGA oder VGA, verschiedene Bildschirmauflösungen, Farbtiefen oder auch die Unterstützung von 3D-Funktionen, wie OpenGL.
- **Audiogeräte.** Viele Rechnerplattformen implementieren Verfahren, Audiosignale aufzunehmen und wiederzugeben. Dabei spielt die Sampling-Rate der Daten und, ob diese in Mono-, Stereo- oder Mehrkanalton vorliegen, eine Rolle.
- **Parallel-, Seriell-, USB-Ports** zum Anschluss von Peripheriegeräten. Dieses kann in einem Hardwaremuseum von Interesse sein, ist aber für digitale Langzeitarchive zunehmend nachrangig.
- **Netzwerk.** Es wird überwiegend als virtueller Ethernet-Adapter bereitgestellt. Diese Technologie wurde Ende der 1970er Jahre entwickelt und in immer neuen Standards für höhere Bandbreiten angepasst. Auf Basis dieses hardwarenahen Standards können eine ganze Reihe von höherschichtigen Netzwerkprotokollen wie TCP/IP, NetBIOS oder IPX/SPX arbeiten.

**Vollständigkeit und Exaktheit der Emulation** Virtualisierer versuchen zur Beschleunigung "Abkürzungen" zu verwenden, so dass keine reale Hardware nachgebildet wird, sondern lediglich möglichst dünne Softwareschichten zum Datenaustausch<sup>59</sup> dienen. Sie sind daher per-se unvollständig. Dieses erfordert jedoch vom Betriebssystem der Nutzungsumgebung oftmals besondere Hardwaretreiber, die vielfach für ältere Systeme nicht zur Verfügung stehen. Deshalb sollte ein Emulator im Sinne der Simulation realer Hardware möglichst vollständig sein, so dass keinerlei nachträgliche Anpassungen der Nutzungsumgebung notwendig sind.

Ein Emulator sollte den Anforderungen einer Nutzungsumgebung weitgehend folgen können oder im Fehlerfall eine entsprechende Meldung generieren. So könnte eine Applikation eine bestimmte Bildschirmauflösung, Farbtiefe oder 3D-Funktionalität erwarten. Die Nichterfüllung eines Punkts generiert üblicherweise eine Meldung seitens des Betriebssystems der Nutzungsumgebung. Diese geeignet in die Referenzumgebung zu übertragen erfordert entsprechende Schnittstellen der Software.

**Unterstützte Betriebssysteme von Nutzungsumgebungen** Aus den vorgenannten Parametern ergibt sich die konkrete Unterstützung bestimmter Betriebssysteme und Applikationen einer Nutzungsumgebung. Diese Aggregation wäre einerseits für Entscheidungs-

<sup>59</sup>Direkte Nutzung der Host-CPU ohne den aufwändigen Umweg über deren Nachbildung in Software - siehe hierzu auch 3.5) oder VMware SVGA II Grafikadapter.

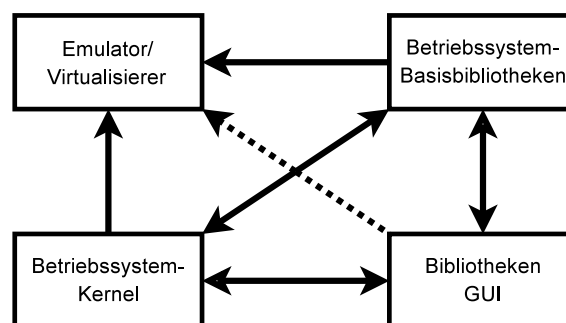
systeme, siehe dazu das spätere Kapitel 6 zu "View-Paths", andererseits für Objektregister<sup>60</sup> interessant. Als Beispiel sei eine Auswahl an Betriebssystemen der X86-Architektur aufgeführt:

- DOS - Microsoft, Digital Research (DR-), IBM-, Free-DOS
- Windows (2.X), 3.1/3.11
- Windows 95/98/ME; Windows NT 3.5/4.0; Windows 2000
- Windows XP; Windows 2003; Windows Vista; Windows 2008
- Linux Kernel: pre 2.2, 2.2, 2.4, 2.6; C-Bibliothek GLIBv1, GLIBv2.X; Linux Executable Format AOUT, ELF
- Free, Net, Open BSD
- Weitere Betriebssysteme für X86 wie BeOS, OS/2, Solaris for X86. Dabei stellt OS/2 größere Herausforderungen an Emulatoren wegen der exzessiven Nutzung bestimmter privilegierter CPU-Funktionen.

Für die Relevanz der einzelnen Betriebssysteme ist zu unterscheiden, ob sie als Primär- oder Sekundärobjekte von Interesse sind.

## 4.8 Langzeitverfügbarkeit von Emulatoren

Emulatoren sind selbst nichts anderes als dynamische digitale Objekte. Für sie gelten die identischen Problemstellungen wie für die Primärobjekte, die ein Archivbenutzer betrachten möchte. Deshalb sind Überlegungen zur Perpetuierung der Emulatoren für zukünftige Benutzung eine zentrale Komponente. Diese Problemstellung wäre nach dem OAIS-Referenzmodell beim Archivmanagement anzusiedeln. Emulatoren müssen daher immer wieder für die jeweils



**Abbildung 4.5:** Ein Emulator ist auf diverse Schnittstellen zum Betriebssystem der Host-Plattform, bestimmte Basis- oder auch GUI-Bibliotheken angewiesen.

aktuellen Hardware- und Betriebssystemkombinationen angepasst werden. Bei kommerziellen Produkten trägt dafür üblicherweise der Hersteller Rechnung, jedoch ist dessen Fokus

<sup>60</sup>Format Registries wie PRONOM oder JHOVE, siehe auch Abschnitt 6.1.

oft ein anderer: Die Produkte orientieren sich an modernen Betriebssystemen. Damit ergeben sich einige Herausforderungen: Würde die Weiterentwicklung der virtuellen Hardware bei gleichzeitiger Weiterentwicklung der Schnittstellen zum Host-Betriebssystem, eingestellt, würde sich für alle nun historischen Betriebssysteme optimalerweise nichts mehr ändern. Das hätte den Vorteil, dass von Seiten der Hersteller keine Treiberentwicklung für Mainboard, Netzwerk-, Grafik- und Soundkarte mehr erfolgen müsste. Ohne diese Anpassungen ließe sich der Virtualisierer jedoch zunehmend schwerer vermarkten.

Emulatoren interagieren als Software für eine bestimmte Rechnerplattform eng mit den Schnittstellen des Betriebssystems und einer Reihe von Basis- und erweiterten Bibliotheken für bestimmte Aufgaben, wie die Benutzerinteraktion. In der Entwicklung und Verwendung von Emulatoren lassen sich deshalb Abhängigkeiten (Abb. 4.5) identifizieren, die über ihren Einsatz entscheiden:

- Weiterentwicklung der Betriebssystem-Kernel mit Änderungen des Application Binary Interface (ABI)
- Weiterentwicklung der Basisbibliotheken, wie zentrale C-Bibliothek
- Weiterentwicklung der grafischen Benutzer-Interfaces (GUI), wie die Einführung von 3D-Desktops
- Weiterentwicklung der Emulatoren

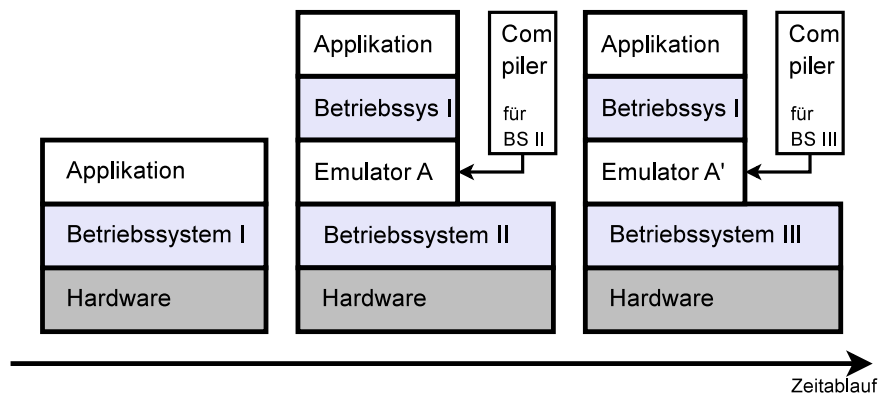
Die bisherigen Forschungen zur Emulation in der Langzeitarchivierung haben die folgenden vier Strategien zur Verfügbarkeit von Emulatoren auf der jeweils aktuellen Rechnerplattform identifiziert:

- Emulator-Migration
- Emulator-Stacking oder -Chaining
- Universal Virtual Machine (UVM)
- Modulare Emulatoren

Einen guten Überblick bieten hier [van der Hoeven und van Wijngaarden 2005] oder der Abschnitt "Preserving the emulator" in [Verdegem und van der Hoeven 2006].

### 4.8.1 Softwaremigration

Dieser Ansatz 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



**Abbildung 4.6:** Die Migration des Emulators bedeutet die Anpassung der Schnittstellen zur Host-Plattform und die Verfügbarkeit geeigneter Compiler (vgl. [Verdegem und van der Hoeven 2006]).

das aktuelle User-Interface dieser Plattform gehen. Langlebige Programmiersprachen wie beispielsweise C sollten eine Übersetzung mit dem jeweils aktuellen Compiler erlauben (Abb. 4.6). Beides setzt jedoch eine gewisse Stabilität der Paradigmen, sowohl der Host-Betriebssysteme und User-Interfaces (Abb. 4.5) als auch der Programmiersprachen voraus:

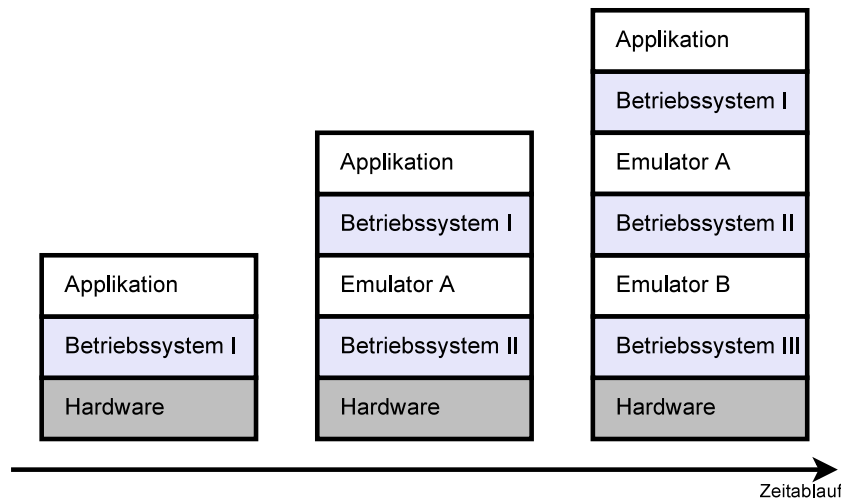
- Der Emulator bildet die Zielplattform komplett nach und verzichtet auf direkte Nutzung von Funktionen der Host-Plattform (bei Virtualisierung mindestens der CPU). Andernfalls ist der Emulator nicht von Architekturwechseln unabhängig.
- Der Quellcode des Emulators steht zur Verfügung, so dass eine Neuübersetzung bei Plattformwechseln jederzeit möglich ist.
- Der Emulator ist in einer Programmiersprache geschrieben, für die ein Compiler der jeweils aktuellen Rechnerplattform zur Verfügung steht, oder
- Der Emulator liegt in einer Programmiersprache vor, die ohne Verlust von Funktionalität auf eine Nachfolgerin oder komplett andere übersetzt werden kann. Voraussetzung hierfür ist eine umfassende Dokumentation.
- Der Code des Emulators ist so modular angelegt, dass Änderungen im Paradigma des User-Interfaces mit nicht zu hohem Aufwand vollzogen werden können. Die konsequente Anwendung dieses Ansatzes führt zur Emulation Virtual Machine (EVM).

Der Vorteil liegt in der Verwendung nur einer Emulationsschicht. Dies bedeutet, dass die Interaktion und der Datenaustausch mit dem Zielsystem direkt erfolgen kann (Abb. 4.6). Das Anpassungsproblem an sich wandelnde Host-Plattformen kann durch im Folgenden aufgeführte Vorgehensweisen versucht werden zu umgehen.

#### 4.8.2 Geschachtelte Emulation

Die üblichen Emulatoren sind mehr oder weniger in ihrer Ausführung von der darunter liegenden Host-Plattform abhängig. Ausnahmen, die jedoch bisher eine geringe Rolle spielen,

sind Java-basierte Emulatoren<sup>61</sup> oder die später vorgestellte EVM. Vorteil und Grund für die-



**Abbildung 4.7:** Eine Möglichkeit, Emulatoren nicht an eine neue Host-Plattform anpassen zu müssen, besteht in der gestapelten beziehungsweise ineinander geschachtelten Emulation (vgl. [van der Hoeven und van Wijngaarden 2005]).

se Herangehensweise besteht in der spezifischen Anpassung an die Host-Plattform, welches üblicherweise für eine gute Performance der Zielumgebung und eine komfortable Integration in die vorgegebene Benutzungsumgebung sorgt.

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, wie Abbildung 4.7 abstrakt zeigt.

### 4.8.3 Universal Virtual Machine

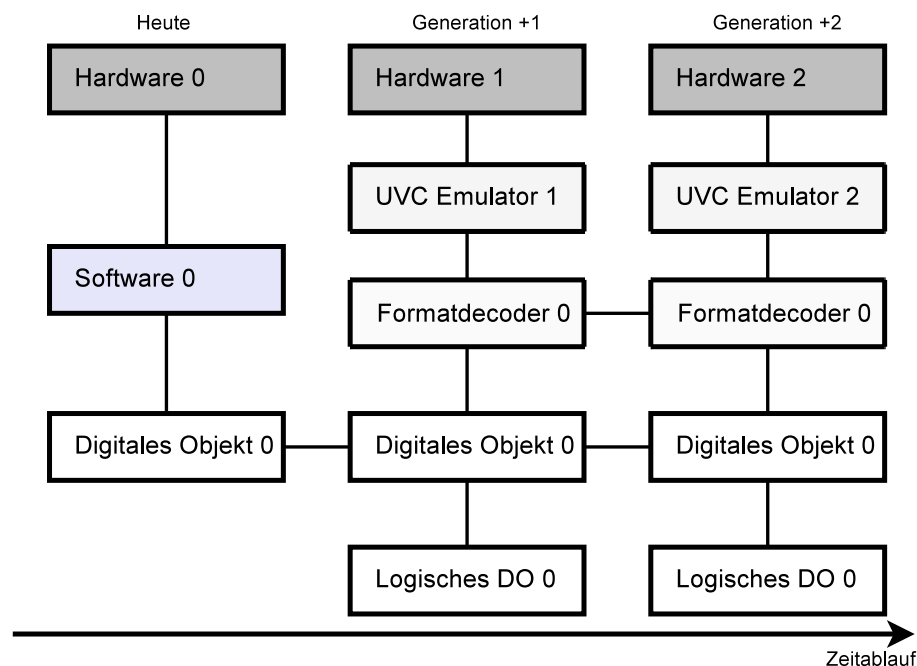
Im Idealfall sollte ein Emulator unabhängig von einer bestimmten Plattform oder Periode der Computerentwicklung sein. Hierzu führte Lorie eine zusätzliche Abstraktionsschicht zwischen Emulator und Host-Plattform in die Diskussion ein.<sup>62</sup> Dieses Konzept erhielt unter anderem die Bezeichnung "Emulation Virtual Machine" (EVM). Basierend auf einem Emulationsinterpreter und einer Emulatorspezifikation sollte es möglich sein, Emulatoren zu schreiben, die auf einer EVM ausführbar sind.

Diese speziellen Emulatoren sollten dann stabil über Perioden hinweg sein und sich auf unterschiedlichen Host-Plattformen ausführen lassen. Eine Variation dieses Ansatzes schlugen van Diessen, Lorie mit dem Universal Virtual Computer (UVC) vor [van Diessen und Lorie 2005]. Eine Anwendung des UVC zumindest auf bestimmte Bildformate wurde im Zuge des DIAS-Projekts von IBM und der Königlichen Bibliothek der Niederlande untersucht.<sup>63</sup> Hierfür wurde der UVC als ein einfacher "General Purpose Computer" implementiert, der sich auf heutige und zukünftige Rechnerarchitekturen portieren lassen soll. Programme, die in einer speziellen UVC-Sprache erstellt wurden, können dann plattformunabhängig im UVC ausgeführt

<sup>61</sup>Vgl. hierzu Abschnitt 5.6.3 für X86-Emulatoren.

<sup>62</sup>Vgl. hierzu [Lorie 2001].

<sup>63</sup>Siehe hierzu [Lorie 2002] oder vgl. die Darstellung in [van der Hoeven u. a. 2005].



**Abbildung 4.8:** Die Darstellung des Funktionsmodell des UVC im Zeitablauf (vgl. hierzu den Aufsatz [Sereiens und Hilf 2003], S. 42 ff).

werden. Jedoch ist dieser Ansatz auf statische Objekte beschränkt, die keinerlei dynamische Komponenten bis hin zur direkten Benutzerinteraktion enthalten.

#### 4.8.4 Modulare Emulationsansätze

Als Weiterentwicklung des EVM-Gedanken werden modulare Emulationsansätze vorgeschlagen und prototypisch entwickelt [Verdegem und van der Hoeven 2006]. Die Autoren des Artikels [van der Hoeven u. a. 2005] definieren modulare Emulation wie folgt: *"Emulation of a hardware environment by emulating the components of the hardware architecture as individual emulators and interconnecting them in order to create a full emulation process. In this, each distinct module is a small emulator that reproduces the functional behaviour of its related hardware component, forming part of the total emulation process."*

Dabei versucht sich der modulare Ansatz in starker Nähe zur klassischen Basishardwarearchitektur des "Von Neumann" Rechners aufzuhalten, um eine hohe Flexibilität durch Nachnutzbarkeit bereits vorhandener Komponenten in neuen Emulatoren zu erreichen. Dieses entspricht durchaus der Realität heutiger Architekturen, die beispielsweise gemeinsame Standards für Peripheriebusse verwenden. Das konzeptionelle Modell definiert hierzu:

- Einen modularen Emulator
- Die Emulatorspezifikation
- Die Universal Virtual Machine
- Eine Component Library

- Verschiedene Controller

Zur Ausführung der Nutzungsumgebung auf der UVM wird ein Emulator benötigt. Dieser hat die Aufgabe, die Ausgangshardware für die Nutzungsumgebung bereitzustellen. Er sollte wegen der dynamischen Randbedingungen modular angelegt sein, wobei jedes Modul eine bestimmte Hardwarekomponente in Software für die UVM nachbildet. Durch Zusammenstellung der je nach gewünschtem Vorlagesystem enthaltenen Komponenten entsteht die jeweils erforderliche Plattform. Hierbei können jüngere Architekturen oft schon auf existierende Module zurückgreifen und es müssen nur die jeweils neuen technologischen Entwicklungen implementiert werden.<sup>64</sup>

---

<sup>64</sup>Eine teilweise Umsetzung dieses Konzepts findet man beispielsweise in MESS oder QEMU wieder, die im folgenden Kapitel beschrieben werden.



## 5 Hardwareemulatoren für den Langzeitzugriff

Emulatoren sind Untersuchungsgegenstand dieser Arbeit und eine der zentralen Komponenten des später zu diskutierenden Softwarearchivs zur Erhaltung von Nutzungsumgebungen. Sie übernehmen die Brückenfunktion zwischen den Erstellungsumgebungen aus der Entstehungszeit der digitalen Primärobjekte und der jeweils aktuellen Referenzplattform. Die nachfolgenden Untersuchungen und Betrachtungen erfolgten im Zuge der Mitarbeit in PLANETS. Sie beschäftigen sich dabei, eingebettet in das Teilprojekt "Preservation Action", mit den Erfolgsbedingungen und Auswahlkriterien für Emulatoren.

Das vorangegangene Kapitel behandelte Hardwareemulatoren und ihre Auswahlkriterien für die digitale Langzeitarchivierung abstrakt. In den folgenden Abschnitten sollen die generellen Überlegungen exemplarisch für zwei Hardwarekategorien konkretisiert werden. Am Beispiel von Home-Computern und der umfangreichen Klasse der X86-PC werden konkrete Applikationen für die Hardwareemulation im Einzelnen vorgestellt: Home-Computer waren die ersten weit verbreiteten Geräte und sie gehören seit 15 Jahren der IT-Geschichte an. Es gab viele verschiedene Plattformen, was eine gewisse Herausforderung an die Emulatoren und das Archivmanagement stellt. Die Zahl der unterschiedlichen Programme für diverse Host-Systeme ist vielfach hoch. Deshalb interessieren besonders modulare Emulatoren, die mehr als eine Plattform abbilden können oder welche, die als Java-Programme sich für eine Reihe von Referenzumgebungen eignen. Ein aus Forschungssicht interessanter Aspekt bestimmt die Klasse der Home-Computer: Der Zuwachs an digitalen Objekten kann als weitgehend abgeschlossen und daher unveränderlich betrachtet werden.

An ganz anderer Stelle im Spektrum der Rechnerarchitekturen stehen die X86-PCs, die auf eine bereits 20-jährige Geschichte zurückblickt und für die es wohl die meisten Betriebssysteme und Programme gibt, mit denen digitale Primärobjekte erzeugt wurden.

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. An sich keine neue Entwicklung, befördert sie jedoch die Idee und das Wissen um Hardwareemulation erheblich. Sie ist deshalb von den reinen Emulatoren abzugrenzen und wird deshalb in einem eigenen Abschnitt behandelt. Neben den genannten Beispielarchitek-

turen existieren weitere nicht unwesentliche Geräteklassen, die jedoch wegen der Größe des Gesamtgebiets lediglich kurz im Anhang (Abschnitt A.1) angerissen werden, da andernfalls ihre ausführliche Darstellung den Umfang dieser Betrachtung sprengen würde.

Für die vorgestellten Emulatoren und Virtualisierer der X86-Architektur sind im Folgenden Versionsnummern oder Veröffentlichungszeitpunkte angegeben. Diese dienen der Orientierung und als Bezugspunkte, da im Zuge von Weiterentwicklungen sich Eigenschaften einzelner Werkzeuge durchaus signifikant ändern können. Die einzelnen Aspekte illustriert das Beispiel der knapp 10-jährigen Entwicklungsgeschichte der "VMware Workstation".

## 5.1 Emulation von Home-Computern

Die Popularität zu ihrer Zeit und die vergleichsweise Einfachheit der Maschinen hat eine große Anzahl von Emulatorprojekten<sup>1</sup> hervorgebracht. Die ersten entstanden bereits in der Blütezeit dieser Architekturen und waren unter anderem für die PC-Plattform mit DOS oder Apple Macintosh geschrieben worden. Im Laufe der Zeit kamen immer wieder neue Emulatoren hinzu, die an die sich ändernden Host-Plattformen und veränderte Paradigmen, wie das Aufkommen der plattformabstrahierenden Programmiersprache Java, angepasst wurden.

Die hohe Anzahl der verschiedenen Home-Computer schlug sich in einer dementsprechenden Zahl separater Emulatoren nieder. Alle diese geeignet aufzubewahren, kann die damit befassten Archive überfordern, insbesondere da die Zahl der jeweiligen Primärobjekte im Verhältnis zu heutigen Plattformen eher niedrig liegt. Interessant könnten diese Emulatoren für die experimentelle Überprüfung einer Ineinanderschachtelung als Langzeitstrategie sein.<sup>2</sup>

Aus diesen Gründen liegt das Augenmerk im Weiteren auf modularen Ansätzen, welche die Zahl der Einzelemulatoren gering halten, und auf Java-basierter Emulation, die auf einer größeren Anzahl Referenzumgebungen inklusive weit verbreiteter Web-Browser ablauffähig ist. Da im Bereich der Home-Computer einerseits Spiele dominieren und andererseits die verschiedenen Maschinen an sich im Mittelpunkt stehen, sind Emulatoren für Home-Computer hauptsächlich für Privatanwender, Technische Museen oder Ausstellungen, wie sie beispielsweise vom Berliner Computerspiele-Museum<sup>3</sup> organisiert werden, von Interesse.

### 5.1.1 MESS

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 PIO und SIO Bausteine hinzu,<sup>4</sup> die sich zwischen den verschiedenen Maschinen nicht wesentlich unterschieden. Deshalb verfolgt MESS einen modularen Ansatz,<sup>5</sup> der eine universelle Basismaschine aufsetzt, die für jedes System, sei es Spielekonsole, Taschenrechner, Handheld oder

<sup>1</sup>Vgl. hierzu die (zwangsläufig unvollständige) Aufstellung in A.1.

<sup>2</sup>Vgl. hierzu den Abschnitt 4.8.2. Versuche dieser Richtung wurden erfolgreich mit einer Reihe von C64 und ZX Spectrum Emulatoren in VMware und QEMU unternommen.

<sup>3</sup>Siehe hierzu den Bereich "Ausstellungen" auf [CSM 2008].

<sup>4</sup>Einen Überblick zu Prozessoren und Peripherie bieten [CHM 2008] oder [CSM 2008].

<sup>5</sup>Siehe für ähnliche Konzepte auch die Abschnitte 4.8.3 und 4.8.4 am Ende des vorangegangenen Kapitels.

Bürocomputer, eine Beschreibung erhält. Ausgehend von dieser greift es auf vorgefertigte Emulationscodeblöcke zurück.

Das Akronym MESS steht für Multiple Emulator Super System.<sup>6</sup> Es greift auf die Codebasis des Multiple Arcade Machine Emulators (MAME) zurück, der verschiedene Spielautomaten auf ähnliche Art emuliert.

**Unterstützte Nutzungsumgebungen** MESS kann inzwischen mehr als 400 verschiedene Geräte, unterschiedliche Revisionen und Clones von diesen nachbilden. Die vom Emulator nachgebildeten Systeme funktionieren nicht alle komplett. Gerade Exoten wie Lisa oder BeBox sind oft noch nicht perfekt. Die unvollständige und permanent erweiterte Liste umfasst: 3DO

ABC 80, 800C/HR, 800M/HR, 802, 806; Acorn Electron; Adventure Vision; Amiga 500, CDTV; Amstrad/Schneider 464+, 6128+, CPC6128, CPC664, KC Compact; APF Imagination Machine, M-1000; Apple I, II(gs), IIc,e und III (inklusive diverser Erweiterungen); Aquarius, Arcadia 2001; Atari 400, 800, 2600, 5200, 7800, Jaguar; Atom

Bally Professional Arcade; BBC Micro Model A/B/B+, Master; BeBox Dual603-66, Dual603-133

C65 / C64DX; CBM20xx Series (Basic 1); Channel F; Chess Champion MK II; ColecoAdam, ColecoVision; Colour Genie EG2000; Commodore 16/116/128/232/264 ,+4; Commodore 30xx, 40xx, 80xx, Commodore 64, C64GS, CBM4064/PET64/Educator64, 64/VC64/VIC64, Commodore B128-40/Pet-II/P500, B128-80LP/610, Max (Ultimax/VC10), SP9000/MMF 9000; Compis; Concept; CPS Changer (Street Fighter ZERO)

DAI Personal Computer

Enterprise 128

Famicom (Twin); FunVision

Galaksija, GameBoy (Pocket, Color); Genesis (Megadrive), Geneve 9640

IBM PC 10/27, Amstrad PC1512 (Version 1, 2), Amstrad PC1640/PC6400, Amstrad PC20, EURO PC, IBM PC 08/16/82, IBM PC/XT (CGA), PC (CGA, MDA), PC/XP (VGA), PC200 Professional Series, Tandy 1000HX, IBM PC/AT (CGA), PC/AT 386 (CGA), PC/AT 486 (CGA), PC/AT 586 (CGA); Indigo2 (R4400, 150MHz), Indy (R4600, 133MHz), Indy (R5000, 150MHz), IRIS Indigo (R4400, 150MHz); Intellivision (Sears); Interton VC4000

Jupiter Ace

Kaypro 2x; KC 85/3, 85/4; KIM-1

Laser 110, 128, 200, 210, 310, 350, 500, 700, Fellow, TX-8000, VZ-200, VZ-300 (Oceania); Lisa2 (2/10); Lynx (II)

Macintosh Plus, 512ke, SE; MC-10; MegaDuck/Cougar Boy; Micro Professor 1(B); Microbee 32 IC/PC/PC85/56; Microtan 65; MSX CF-1200, CF-2000, CF-2700, CF-3000, CF-3300, Expert (DDPlus, Plus), FS-1300, FS-4000, HB-201(P), HB-501P, HB-75D,P, HB-8000 Hotbit 1.2, IQ-1000 DPC-100,180,200; MSX2, FS-4500, FS-4600, FS-4700, FS-5000F2, FS-5500F1/F2, FS-A1(F,FM,MK2), HB-F1, HB-F1II, HB-F1XD, HB-F1XDMK2, HB-F500(P), HB-F700D,P,S, HB-F900, HB-F9P, HB-G900P, IQ-2000 CPC-300, NMS-8220, NMS-8245, NMS-8250, NMS-8255, NMS-8280, VG-8235, WAVY PHC-23, X-II CPC-400, X-II CPC-400S; MSX2+ FS-A1FX, FS-A1WSX, FS-A1WX, HB-F1XDJ, HB-F1XV, WAVY PHC-35J, WAVY

<sup>6</sup>Siehe [MESS 2008], einen Screenshot bietet Abb. A.5 im Anhang.

PHC-70FD, WAVY PHC-70FD2; Memotech MTX512; MZ-700  
 Nascom 1 NasBug T1,2,4, Nascom 2 (NasSys 1,3); NC100; Nintendo 64; Nintendo Entertainment System  
 Odyssey 2; Oric 1, Atmos, Telesrat, Pravetz 8D  
 PC Engine (TurboGrafx 16), PC-8801 MKIISR; PC/AT (VGA); PDP-1; Philips P2000T,M; PK-01 Lviv; PMD-85.1 Alfa, Mato, PMD-85.2(A,B),3; Pocket Computer 1251,1350,1401, 1402,1403(H); Primo A-32,48,64,B-64  
 QL (diverse Revisionen)  
 Sam Coupe; Sega GameGear; Sega Mark III; Sega Master System/Samsung Gamboy, Plus, II (Tec Toy), III Compact; Sony PlayStation (diverse Modelle), PS one; Sorcerer (diskless); Sord M5; Super GameBoy Super Nintendo Entertainment System; Supervision; Superboard II Model 600; SVI-318,328  
 Tandy TRS-80 Color Computer (2, 2B, 3), CP400, Dragon 32, Dragon 64; Tatung Einstein TC-01; Telmac TMC-600 (Series I,II); TI Avigo 100 PDA; TI Model 990/10 Minicomputer System, 990/4 Minicomputer System; TI-81,85,86,99/8; TI99/4(A) Home Computer,TM 990/189 University Board Microcomputer mit University Basic (Video Board Interface); Tommy Tutor; TRS-80 Model I (Level I Basic), LNW-80, System-80, TRS-80 Model I; TX-0 original/upgraded  
 Vectrex (Spectrum I+); VIC20, VIC1001, VIC20/VC20  
 WonderSwan (Color)  
 Z88, ZX Spectrum, Inves Spectrum 48K+, TC-2048, TK90x Color Computer, TK95 Color Computer, TS-2068, UK-2086, +4, ZX Spectrum 128, +2, +2a, +3, +3e, +128; ZX-80 (Aszmic, 2nd, 3rd), ZX-81, Lambda 8300, PC8300, Power 3000

**Allgemeine Metadaten** MESS ist Open Source und unterliegt der GPL-Lizenz. Es wird von einer größeren Entwicklergemeinschaft getragen und aktiv gepflegt.<sup>7</sup> 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.<sup>8</sup>

### 5.1.2 ARAnyM

ARAnyM - Atari Running on Any Machine - ist als Multiplattform-Virtuelle-Maschine für die Ausführung von Atari ST/TT/Falcon TOS/GEM Applikationen auf beliebiger Hardware angelegt. Das Projekt<sup>9</sup> existiert seit mehr als sieben Jahren und ist Open Source. Der Emulator kann auf einer Reihe von verschiedenen Betriebssystemen, wie Linux, Windows oder Mac-OS X ausgeführt werden.

Er umfasst eine Motorola 68040 CPU mit MMU040, FPU68040 und 68881/2. Als Speicher können angefangen von 14 MByte ST-RAM bis zu 3824 MByte FastRAM konfiguriert werden.

<sup>7</sup>Gute Einstiegspunkte sind [EMUU 2008], [AEP 2008] oder die Homepage - <http://www.mess.org>.

<sup>8</sup>Hinweise zu den ROMs, dem Auslesen und verschiedenen Formaten finden sich unter anderem auf den vorgenannten Quellen.

<sup>9</sup>Siehe für weitergehende Informationen - <http://www.arany.org>.

Die Emulation der Videohardware unterstützt: VIDEL, Blitter, MFP, ACIA und IKBD für höchst mögliche Kompatibilität. Die Soundausgabe ist kompatibel zum Atari XBIOS Sound Subsystem mit TimerA DMA Interrupt.

Es können Atari-Floppies in Double- und High-Density für den Anschluss als Disketten-Images oder von realen Diskettenlaufwerken definiert werden. Es gibt zwei virtuelle IDE-Kanäle für die Verbindung von Festplatten-Images, realen Festplatten oder CD-ROMs. Für Linux-Host-Systeme kann die Software direkt auf das CD-ROM-Laufwerk zugreifen, ohne dafür eine IDE- oder SCSI-Emulation zu nutzen. Es besteht eine umfangreiche Tastatur- und Mausunterstützung. Zur Erleichterung von Softwareinstallationen erlaubt der Emulator einen direkten Zugriff auf das Host-Filesystem via BetaDOS und MiNT XFS-Treiber. Für die Netzwerkeinbindung wird eine Ethernet-Emulation mit einem MiNT-Net Treiber eingesetzt.

Der Emulator kann TOS 4.04, EmuTOS oder Linux direkt booten und kann mit Free-MiNT, MagiC, und anderen Betriebssystemen arbeiten, die auf realen Ataris ablauffähig sind.

### 5.1.3 Home-Computer-Emulatoren auf Java-Basis

Eine Klasse für sich sind Java-basierte Emulatoren. Sie sind nicht an eine ganz bestimmte Host-Plattform gebunden, sondern plattformübergreifend auf allen Maschinen ausführbar, die über eine passende Java Virtual Machine (JVM) verfügen. Beispiele sind:

- JaC64 - Ein schon länger existierender Commodore C64 Emulator,<sup>10</sup> der im Web-Browser als Applet oder in einer Standalone-Version ausführbar ist.
- Java Atari-ST Emulator - auch schon älter, da das Projekt seit 2000 inaktiv zu sein scheint.<sup>11</sup>
- Jasper - ZX Spectrum Emulator - letzte Änderung im September 2002, der Quellcode ist erhältlich.<sup>12</sup>

Auch wenn einige der genannten Projekte nicht mehr aktiv sind, so konnten sie jeweils noch ausgeführt werden. Sie zeigen, wie durch die plattformübergreifende Zwischenschicht auch nach längerer Zeit ohne Aufwand seitens des Anwenders ältere Systeme emuliert werden konnten. Fragen des geeigneten Objekttransports sind jedoch mit diesen Beispielen nicht geklärt. Die gezeigten Programme waren jeweils schon "vorinstalliert". Ebenso konnte man sich mit der Problematik der Übersetzungen der Tastatureingaben vertraut machen. Diese wird im späteren Abschnitt 6.5.4 wieder aufgenommen.

Einen ähnlichen, plattformübergreifenden Ansatz realisiert das noch sehr junge Projekt "IronBabel". Die Basisinfrastruktur fundiert auf .NET. Dabei wird ein modulares Konzept verfolgt: *"IronBabel is divided into the core, graphics modules, sound modules, input modules,*

<sup>10</sup>Siehe hierzu die Homepage - <http://www.jac64.com>, der Quellcode des Emulators ist von <http://sourceforge.net/projects/jac64> erhältlich.

<sup>11</sup>Die vielfach noch referenzierte Homepage - <http://www.atari-st.lovely.net> ist seit Mitte 2007 nicht mehr erreichbar.

<sup>12</sup>Die ursprünglich referenzierte Homepage - <http://www.spectrum.lovely.net> ist ebenfalls nicht mehr erreichbar. Eine Alternative zur Betrachtung bietet - <http://www.wearmouth.demon.co.uk/jav/gp.htm>. Hier sind zudem Übersetzungen der Tastaturbelegung für die Bedienung im Browser angegeben.

*CPUs (PSX, PS2, and N64 are all MIPS-based; Wii, Gamecube, and XBox 360 are all PPC-based), and platforms. A platform defines how the CPU, RAM, sound devices, etc need to be structured to execute a given binary. In this way, new platforms can be targetted with ease.*<sup>13</sup>

## 5.2 Virtuelle X86-Maschinen

Für längere Zeiträume bis in die Mitte der 1990er Jahre galt die Hardwarevirtualisierung professionellen Maschinen, wie Mainframes vorbehalten. X86-Prozessoren waren anfangs zu leistungssarm und schienen nicht über die notwendigen Eigenschaften zu verfügen, um mehrere Betriebssysteme parallel ablaufen zu lassen. Das änderte sich mit den immer leistungsfähiger werdenden Prozessoren zum Ende des letzten Jahrzehnts und der Gründung der Firma VMware.

Die meisten X86-Virtualisierer arbeiten nach dem gleichen Prinzip: Sie emulieren eine Reihe populärer Peripheriegeräte oder Schnittstellen und virtualisieren lediglich die CPU. So besitzen die emulierten Grafikkarten, SCSI-Busse oder IDE-Schnittstellen, BIOSes, Netzwerkkarten und Audioadapter reale Pendanten. Virtualisierer laufen als normale Applikationen auf dem Host-System, können aber durch spezielle privilegierte Kernel-Treiber bestimmte CPU-Operationen am darunterliegenden Betriebssystem vorbei ausführen oder modifizieren. Hierzu wird typischerweise Binary Translation, die Übersetzung von kritischen Befehlen des *Gastsystems* in durch den Virtual Machine Monitor (VMM) behandelbare, angewandt.<sup>14</sup>

Das ändert sich derzeit durch die Erweiterung der Prozessoren von Intel und AMD durch Virtualisierungstechniken, welche unter den Bezeichnungen VT<sub>x</sub> und VT<sub>i</sub><sup>15</sup> sowie SVM und AMD-V<sup>16</sup> zu finden sind. Während traditionell vier Privilegienstufen<sup>17</sup> der Prozessausführung seit dem 386er implementiert waren, teilen die neuen Techniken den höchsten Ring 0 in zwei Unterstufen. Damit ist es möglich, den Betriebssystemkern in die sogenannte VMX-Non-Root zu verschieben. Der Virtualisierungs-Controller, Hypervisor genannt, läuft im privilegierten VMX-Root.

Zur Erreichung von in sich abgeschlossenen virtuellen Maschinen führen die Applikationen eine logische Schicht zwischen Nutzer und Ressource ein, damit die physischen Gegebenheiten der Hardware verborgen bleiben. So kann der Virtualisierer jedem Gastsystem, dieses sind die interessierenden Nutzungsumgebungen, vorgaukeln, dass sie die alleinigen Nutzer einer bestimmten Ressource sind. Die Verwaltung der jeweiligen Ressource obliegt dabei weiterhin dem Host-Betriebssystem. Aus diesen Gründen gelingt es nicht, Virtualisierer ineinandergeschachtelt auszuführen. Dieses bleibt Emulatoren vorbehalten.

<sup>13</sup>Siehe Projekt-Homepage - <http://code.google.com/p/ironbabel>.

<sup>14</sup>Siehe hierzu auch [Thorns 2007], Kapitel 2 oder [Adams und Agesen 2006]. Binary Translation kann auch dazu verwendet werden, Code einer anderen Architektur auszuführen. Dieses nutzt beispielsweise das Rosetta-Paket, um alte PowerPC Programme auf den Intel Versionen von Apple Rechnern ausführen zu können.

<sup>15</sup>Man findet auch die Bezeichnung "Vanderpool Technology".

<sup>16</sup>AMD stellt CPUs in Non-Uniform-Memory-Architecture (NUMA) bereit. Die anfangs Pacifica genannte CPU-Virtualisierung implementiert gleichzeitig die Speichervirtualisierung als CPU-Feature mit. Siehe hierzu auch [Adams und Agesen 2006].

<sup>17</sup>Auch "Ringe" genannt, vgl. hierzu [Stallings 1998].

## 5.3 VMware - Pionier der X86-Virtualisierung

Virtualisierung innerhalb der X86-Architektur wurde mangels geeigneter Prozesseigenschaften und Rechenleistung anfangs nicht in Betracht gezogen. Mit dem Siegeszug dieser Architektur seit den 1990er Jahren und durch den schrittweisen Ersatz von Unix-Workstations durch PCs stiegen Bedürfnis und Leistungsfähigkeit gleichermaßen. Bereits vor zehn Jahren gab es eine gewisse Anzahl abgelöster X86-Betriebssysteme, für die immer noch wichtige, aber nicht portierte, Applikationen im Umlauf waren. Zudem interessierten sich Entwickler und Tester zunehmend für Lösungen, mehr als eine Betriebssysteminstanz auf einem einzigen Rechner ausführen zu können.

Die zu diesem Zweck gegründete Firma VMware [VMware Inc. 2008] programmierte den ersten kommerziell erfolgreichen PC-Virtualisierer und brachte ihn Ende der 1990er Jahre in einer Variante zur Nutzung auf dem Desktop eines Betriebssystems mit grafischer Benutzerschnittstelle auf den Markt. Der große Erfolg des Produkts führte einerseits zur stetigen Produktpflege und Ausweitung der Angebotspalette, andererseits fanden sich eine Reihe von Nachahmern, die konkurrenzfähige Software nachprogrammierten. Einen kompakten Überblick zu verschiedenen X86-Virtualisierern unter dem besonderen Aspekt der Langzeitarchivierung liefert [von Suchodoletz und Welte 2006].

### 5.3.1 Virtualisierer für den Desktop - Workstation

Das als "VMware Workstation" getaufte Produkt war der Einstieg von VMware in die X86-Virtualisierung. Es folgten regelmäßige Updates und Verbesserungen der Ursoftware und im Laufe der Zeit eine Ausweitung der Produktpalette bis hin zu hochskalierenden Lösungen für große Rechenzentren.<sup>18</sup> Für die Langzeitarchivierung ist die VMware Workstation insofern interessant, als dass sie bereits seit knapp zehn Jahren existiert und mit der fortlaufenden Entwicklung der PC-Architektur und ihrer Betriebssysteme mitgeführt wurde.

**Generelle Metadaten** Die VMware Workstation 2.0 erschien Anfang des Jahres 2000, die Version 3.0 wurde im Januar 2002 veröffentlicht. In den folgenden Jahren erschienen sukzessive neuere Versionen und jeweils eine Reihe von Updates älterer Versionen. Insgesamt kann die Workstation auf die längste Produktgeschichte aller untersuchten Virtualisierer zurückblicken.

**Lizenzen und Kosten** Alle VMware Produkte sind kommerzielle Software, zu der keinerlei Quellcode verfügbar ist. Lediglich die notwendigen Treiber der unterstützten Open-Source-Betriebssysteme stehen unter einer freien Lizenz. Die Aktivierung der Software arbeitet bisher offline: Es genügt ein Lizenzschlüssel. Eine Online-Verbindung zum Server des Anbieters ist für die hier betrachteten Produkte nicht notwendig, auch wenn eine Online-Kontaktaufnahme zur Registrierung nahegelegt wird.

Während anfänglich die Workstation ein klassisches Softwareprodukt mit den üblichen Lizenzkosten war, gibt es – vermutlich befördert durch die Konkurrenz – kostenlose Produktvarianten, die nur wenigen Einschränkungen in der Nutzung unterliegen. Da alle Produkte eine gemeinsame Plattform besitzen, lassen sich die Aufgaben des Archivmanagements, die

---

<sup>18</sup>Player, Server, Fusion und ESX-Server als Flaggschiffprodukt.

nur wenige Lizenzen für die jeweiligen Arbeitsplätze benötigen, von denen der Präsentation gegenüber dem Endnutzer trennen. Für Endanwender, die in vorbereiteten Nutzungsumgebungen ihre Primärobjekte sichtbar machen wollen, bieten sich der VMware Player für den lokalen Desktop und der VMware Server für die entfernte Ausführung auf einem spezialisierten gemeinsamen Server an.

**Technische Informationen** Die VMware Workstations arbeiten nach dem im Abschnitt 5.2 erläuterten X86-Prinzip der emulierten Peripherie bei direkter Weitergabe der unkritischen CPU-Kommandos. Hierzu benötigt die Workstation im Host-System jedoch spezielle Kernel-Treiber, die vom jeweiligen Systemadministrator mit entsprechenden Privilegien zu installieren sind.

**Nutzbarkeit, Zuverlässigkeit, Portabilität** Das Produkt orientiert sich klar am Desktop-Benutzer, der direkt mit dem Programm auf einer grafischen Benutzeroberfläche interagiert. Vor dem Erscheinen der Version 6 war es nicht vorgesehen, das Programm ohne grafische Oberfläche zu starten.

Die Zahl der Kommandozeilenoptionen hält sich stark in Grenzen und erlaubt nur wenige der Steuermöglichkeiten, die das grafische Interface bietet. Zudem wird erst seit der jüngsten Version eine Steuerungsschnittstelle offeriert, die es erlaubt, bestimmte Aufgaben maschinen- statt menschengesteuert vorzunehmen. Dieses Paradigma macht es einerseits dem Anwender leicht sich zurecht zu finden und die Steuerung zu verstehen. Andererseits erschwert es die Automatisierung bestimmter Vorgänge, wie das Ein- und Aushängen virtueller Wechseldatenträger.

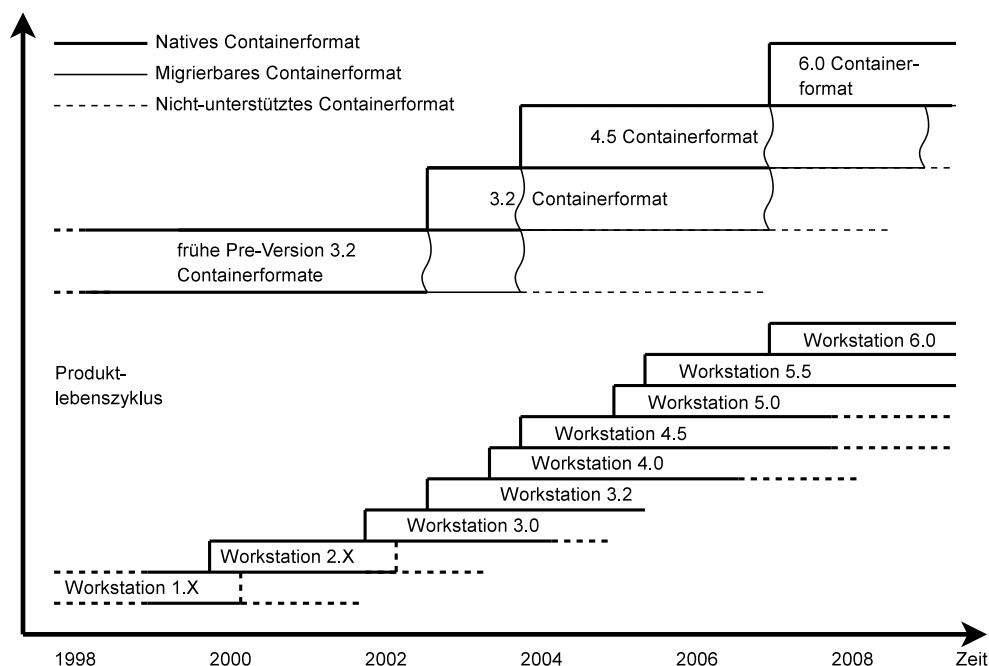
Zentrale Punkte der Bewertung einer Emulations- oder Virtualisierungssoftware für die Langzeitarchivierung sind deren eigene Dateiformate: Einerseits die Formate der Konfigurationsdateien, die üblicherweise in genereller Form bestimmen, wie eine Maschine aussieht und sich verhält, und noch wichtiger andererseits die Formate der virtuellen Datenträger. Die Nutzung real vorhandener Partitionen, die VMware in einigen Produktversionen ebenfalls anbietet, ist für die Langzeitarchivierung nicht von Bedeutung.

Diese Containerdateien bilden die logische Struktur von Festplatten ab. Sie sind dabei an Zugriffsgeschwindigkeit und Platzeffizienz orientiert. So wird, unter der Voraussetzung der Anwahl der entsprechenden Konfigurationsoption, bei der Definition einer beispielsweise 10 GByte großen virtuellen Festplatte nicht sofort der gesamte Speicherplatz im Host-Dateisystem alloziiert. Die Datei wächst lediglich mit dem realen Bedarf der Nutzungsumgebung in der Workstation. Das Dateiformat wurde erst vor einiger Zeit offengelegt.<sup>19</sup>

Das Problem der VMware Containerdateien liegt in ihrer Veränderung des logischen Aufbaus. Zwar sind die Dateiformate über Produktgrenzen einer Generation hinweg identisch, so dass sie in der Workstation, im Player, dem Server oder VMware Fusion ohne Anpassung genutzt werden können. Jedoch gibt es im Rahmen der Produktpflege immer wieder Updates, die ältere Versionen im Laufe der Zeit von neueren Produkten unlesbar werden

<sup>19</sup>Mit dem Produkt VMware Converter, dem Kommando "vmware-mount.pl" stehen zwei hausinterne Programme für den Zugriff auf die Container zur Verfügung. Sehr interessant ist das Tool "qemu-img", welches von und in das VMware-eigene Containerformat wandeln kann. Siehe dazu auch Abschnitt 5.6.1 weiter hinten.





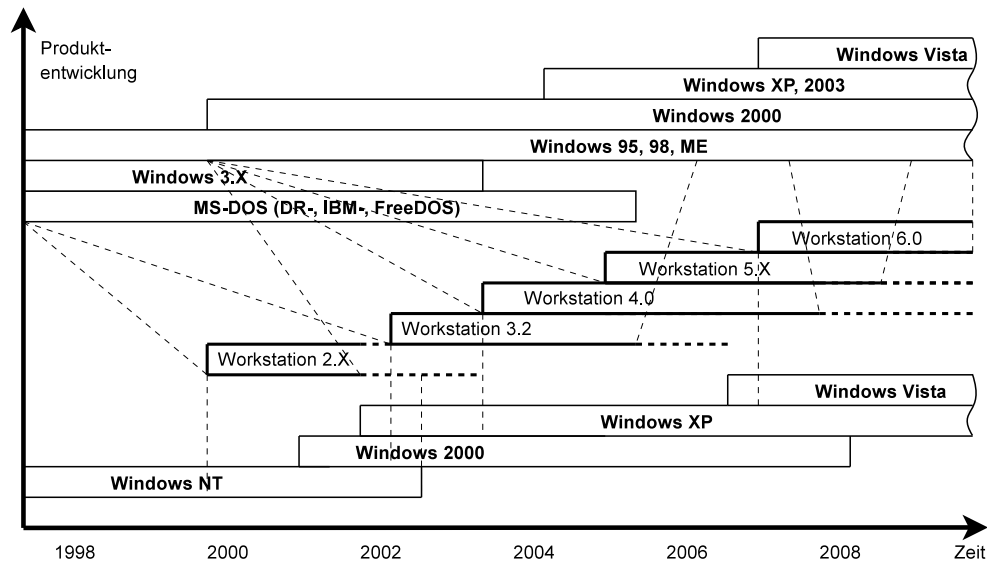
**Abbildung 5.1:** Ein nicht zu unterschätzendes Problem besteht in der Änderung des Containerformats, welches VMware in regelmäßigen Abständen im Zuge der Produktpflege vornahm.

lassen (Abb. 5.1). Zudem unterscheiden sich die Containerformate von virtuellen IDE- und SCSI-Festplatten.

Die Konfigurationsdateien<sup>20</sup> arbeiten nach dem Parameter="Value" Prinzip. Sie sind leicht zu lesen und als Textdateien gut zu editieren. Das erlaubt, sie bequem an bestimmte Nutzungsumgebungen angepasst automatisch erstellen zu lassen. Das Format und die erlaubten Parameter haben sich im Laufe der Zeit leicht verändert. Dieses drückt sich durch eine Versionsnummer im Parameter `config.version = "N"` aus.

**Unterstützte Nutzungsumgebungen und Host-Plattformen** Die regelmäßig erneuerten VMware-Versionen bewegen sich als eine Art "sliding window" (Abb. 5.2) durch den Zeitablauf: Sie werden an die jeweils aktuellen Host-Systeme angepasst und sind typischerweise selbst in der Lage, bis zu diesem Zeitpunkt aktuelle Nutzungsumgebungen ablaufen zu lassen. Wegen ihrer direkten Interaktion mit dem Host-System arbeiten sie mit speziellen Kernel-Treibern zusammen, die von diesem abhängen. Da sich mit neuen Versionen der Host-Systeme die Schnittstellen ändern, funktionieren ab einem bestimmten Zeitpunkt die Treiber nicht mehr. Das gleicht üblicherweise die jeweils folgende Produktversion aus. Mit der Abkündigung von Microsoft Betriebssystemen oder dem Sinken des Marktanteils älterer Kernel-Versionen im Linux Bereich unterstützen neu erscheinende VMware-Produkte sukzessive ältere Betriebssysteme als Host-System nicht mehr.

<sup>20</sup>Die Datei "preferences" regelt gemeinsam über die angelegten Nutzungsumgebungen hinweg für die Einstellungen der Oberfläche. Für jede definierte Nutzungsumgebung wird eine Datei mit der Endung \*.vmx angelegt, die die Einstellungen der virtuellen Hardware aufnimmt. Siehe hierzu eine Liste der verschiedenen Parameter auf <http://sanbarrow.com/vmx.html>.



**Abbildung 5.2:** Die aufsteigenden Versionen der VMware Workstation vollzogen sowohl die Entwicklung der Host-Systeme als auch der Nutzungsumgebungen nach, illustriert am Beispiel der Microsoft Betriebssysteme.

Das Softwarepaket liegt in der Windows Variante im ausführbaren Installer-Format vor, für Linux Systeme werden eine TGZ-gepackte und eine RPM<sup>21</sup> Variante angeboten. Im Paket sind neben den ausführbaren Programmdateien und den notwendigen Bibliotheken auch Konfigurationsdateien sowie eine Sammlung von SVGA- und Maustreibern für die unterstützten Nutzungsumgebungen enthalten.<sup>22</sup> Während für Windows Host-Systeme die Kernel-Treiber in vorkompilierter Form angeboten werden, gilt dieses nur bestimmte Linux Kernel. Liegt dem Installationspaket kein passendes Modul bei, so kann dieses bei der Einrichtung der Software<sup>23</sup> und Vorhandensein einer vollständigen Tool-Sammlung zum Kompilieren von Kernel-Modulen nachgeholt werden.

**Anforderungen an das Host-System** Generell wird von allen Produkten wegen der unvollständigen Emulation als Host-Plattform nur die X86-Architektur unterstützt. VMware Workstation, Player oder Server lassen sich nicht in verschiedenen Versionen parallel auf einem System installieren, da sich die notwendigen Kernel-Module und Treiber unterscheiden. Ausnahmen sind die aktuellen Workstation-Pakete der Version 5.5 und 6.0, die bereits den jeweiligen Player der Version 1 oder 2 enthalten.

Für Linux 2.4er Kernels werden zwei spezielle Kernel-Module "vmnet.o" und "vmmon.o" benötigt. Diese sind für die CD-Versionen der populären Distributionen Bestandteil des Installationspakets, müssen aber für neuere Kernels und unbekannte Versionen mit Hilfe von

<sup>21</sup>Komprimierte TAR-Archive und der Redhat Package Manager sind sehr verbreitete Verfahren der Softwareverteilung.

<sup>22</sup>Das beinhaltet nicht alle Hardwaretreiber – beispielsweise müssen Netzwerkkarten- und Audioadaptertreiber für Microsoft Betriebssysteme selbst beschafft werden. Hiermit beschäftigt sich u.a. Kapitel 7.

<sup>23</sup>Das mitinstallierte Perl-Skript "vmware-config.pl" sorgt beispielsweise für die Einrichtung der virtuellen Netzwerkverbindungen zum Host-System.

"vmware-config.pl" kompiliert werden. Hier kommt es vor, dass diese Übersetzung für aktuelle Kernels wegen geänderter Header-Dateien und Strukturen nicht mehr funktioniert. Ähnliches gilt für Linux 2.6er Kernels, deren Module "vmnet.ko" und "vmmon.ko" heißen. Mit der Workstation der Version 6 und dem Player 2.0 kommt ein weiteres Modul, "vmblock.(k)o" hinzu. Die notwendigen Windows Treiber sind Bestandteil des Installationspakets und müssen nicht übersetzt werden.

**Interaktion, Datenim- und -export** VMware Workstation und Player sind als grafische Desktop-Applikationen angelegt, die nicht ohne diese Schnittstelle betrieben werden können. Lediglich die aktuellste Workstation kann bedingt im Headless-Mode betrieben werden. Der VMware Server verfügt wie die Workstation über eine umfangreiche grafische Schnittstelle, die auch komplett losgelöst von der Host-Maschine auf weiteren Rechnern über das Netzwerk genutzt werden kann. Hinzu kommt ein Web-Interface zur Konfiguration und Steuerung einzelner Nutzungsumgebungen.

Je nach Fähigkeiten und Treiberunterstützung der Nutzungsumgebungen können Daten via Wechsellaufwerk oder virtuelles Ethernet ausgetauscht werden. Das Kommando "vmware-mount.pl" ist in der Lage formatierte Partitionen von VMware Containerdateien in das Host-System einzuhängen. Mit dem Werkzeug "vmware-vdiskmanager" steht ein Offline-VMDK-Manager zur Verfügung, der bestimmte Aktionen, wie das Ändern einiger Parameter, das Verkleinern, Vergrößern oder Defragmentieren beherrscht. Offline Formatierung oder Partitionierung, die für eine automatisierte Vorbereitung von Containern wichtig wären, lassen sich so nicht realisieren.

**Unterstützte Nutzungsumgebungen** Der Fokus der VMware-Produkte liegt eher auf aktuellen Betriebssystemen, die in einem gewissen Umfang immer noch am Markt sind. So setzt beispielsweise die Treiberunterstützung für den speziellen VMware VGA-Controller erst mit Windows 95 ein.

Aufgrund des eingesetzten Phoenix BIOS, welches die Standard-API der X86 bereitstellt, sind auch alle älteren Betriebssysteme lauffähig, die auf dieser API aufsetzen. Eben dieses gilt auch für das Grafik-BIOS, welches jedoch jede Ausgabe auf 640x480 Bildpunkte bei 256 Farben beschränkt.

### 5.3.2 Workstation 3.2

Die Version 3.2 erschien 2001 und war ein sehr lange verfügbares Produkt, das als Update die 3.0 recht zügig ablöste (vgl. Abbildung 5.1). Eine letzte Aktualisierung erschien 2005 und war bis Ende 2006 über den Downloadbereich von [VMware Inc. 2008] erhältlich. Diese Workstation war die letzte Ausgabe für Linux-Hosts, die für das grafische Benutzer-Interface statisch gegen die Motif-Bibliothek<sup>24</sup> gelinkt war.

Eine wichtige Neuerung dieser Version lag in der Veränderung des eingesetzten Container-formats der virtuellen Festplatten, welches nicht mehr rückwärtskompatibel zur Version 3.0

<sup>24</sup>Motif war eine kommerzielle, auch für Linux erhältliche Bibliothek für grafische Elemente, wie Fensterrahmen, Menüs oder Dialogboxen. Sie wurde beispielsweise auch von frühen Netscape-Varianten benutzt. Mit der Durchsetzung der GTK- und Open-Source-Freigabe der Qt-Bibliotheken verlor Motif rasant an Bedeutung.

und davor war. Das stellte insofern kein Problem dar, als dass beim Aufruf eines Gastsystems in der neuen Version eine automatische Konvertierung des Formats vorgenommen wurde. Mit dem Erscheinen späterer Versionen funktionierte der Import dann nicht mehr.

Mit dem Produkt wurde im typischen Softwarepaket ein ausführliches gedrucktes Handbuch verbreitet, die Dokumentation in englischer Sprache ist immer noch online verfügbar.<sup>25</sup> Zum Betrieb der Software wird eine Lizenz der Version 3 benötigt.

Lauffähig ist die Version auf Linux Systemen, solange die Übersetzung der benötigten Kernel-Module für die Monitor- und Netzwerk-Interfaces gelingt. Auf Microsoft Betriebssystemen kann die Workstation 3.2 sowohl in Windows NT, 2000 und XP installiert werden. Ein Betrieb unter Vista ist nicht mehr möglich.

Die Workstation der Version 3.2 bildet ein Pentium I Basissystem bestehend aus einem speziellen PHOENIX BIOS und dem Intel 430TX Chipset<sup>26</sup> in Software nach. Installierten Nutzungsumgebungen stehen maximal 128 MByte Arbeitsspeicher zur Verfügung.

Die Version ist wegen Veralterung und zunehmenden Problemen, sie auf aktuellen Referenzumgebungen auszuführen, nicht für eine Langzeitarchivierung von Interesse. Die Aufgaben dieser Version lassen sich besser durch ihre Nachfolgeprodukte oder alternative Werkzeuge, wie Open-Source-X86-Emulatoren, beispielsweise QEMU, erledigen.

### 5.3.3 Workstation 4.0

Das erste Mal veröffentlicht wurde die Version 4.0 im April 2003. Insgesamt fünf Updates erschienen bis zur Version 4.0.5 im Jahre 2004. Das Produkt steht wie schon die Vorgängerversion nicht mehr auf der VMware-Homepage zur Verfügung. Mit der neuen Major-Release wurde eine neue Lizenz erforderlich, da Lizenzen der Version 3.0 und 3.2 nicht mehr eingesetzt werden konnten.

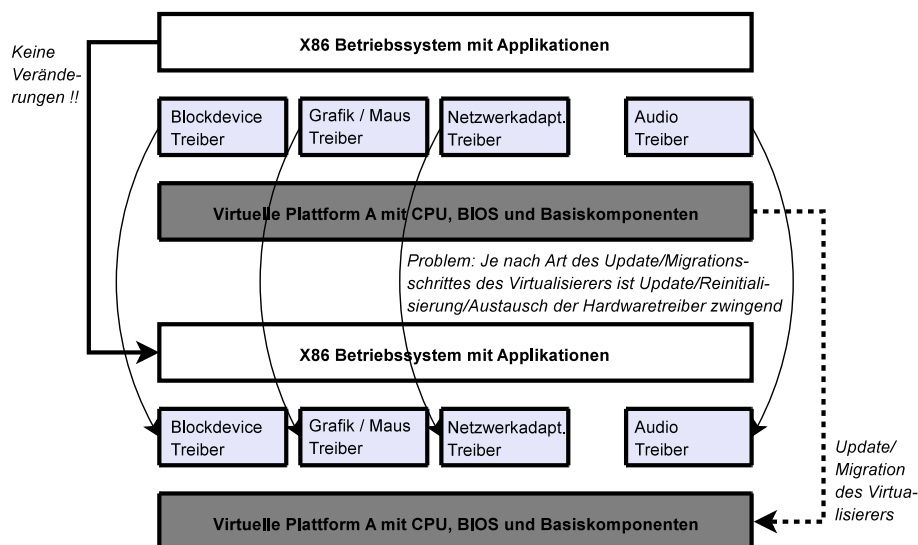
Die Hardware der virtuellen Maschine<sup>27</sup> erhielt eine wesentliche Auffrischung, womit mindestens ein Treiber-Update der Nutzungsumgebungen erzwungen wurde (Abb. 5.3): Als Basis kam nun der Intel 440BX/ZX Chipsatz zum Einsatz. Die Audiokarte basierte auf der Ensoniq mit dem verbreiteten ES1371<sup>28</sup> Soundchip. Ebenfalls aktualisiert wurde der proprietäre Grafikadapter auf SVGA II, der lediglich 2D-Fähigkeiten besitzt. In diesem Zuge wurde die Treiberunterstützung für ältere Windows 3.X Varianten aufgegeben. Eine wesentliche Neuerung der Version lag in der Einführung einer virtuellen USB-Schnittstelle, die an den USB-Controller des Host-Systems angeschlossene Geräte transparent in die virtuelle Maschine durchreichen konnte. Dieses Feature ist jedoch für die Nutzung in der Langzeitarchivierung von nachrangiger Bedeutung. Neben den großen Umstellungen innerhalb der Maschine kamen Änderungen des Aussehens hinzu: Ab dieser Version wurde für die Linux-Variante die GTK-Bibliothek für alle grafischen Ausgaben eingesetzt, welche dynamisch gelinkt ist. Um jedoch zu verhindern, dass das Programm auf Systemen nicht startet, auf denen diese Bibliotheken nicht installiert

<sup>25</sup>Vgl. [http://www.vmware.com/pdf/ws32\\_manual.pdf](http://www.vmware.com/pdf/ws32_manual.pdf) auf [VMware Inc. 2008].

<sup>26</sup>Virtuelle Hardware der Version 2 - dieses drückt sich in der Konfigurationsdatei zu den erstellen virtuellen Maschinen in der Variable "virtualHW.version" aus.

<sup>27</sup>Virtuelle Hardware der Version 3.

<sup>28</sup>Er war vielfach auf den beliebten "Creative Soundblaster PCI" Karten verbaut. Die Soundübergabe an die Host-Plattform ließ zu wünschen übrig, so dass es immer wieder zu Unterbrechungen und Verzerrungen kam.



**Abbildung 5.3:** VMware stellt dem Benutzer zwar einen Assistenten zur Seite, der ihn bei der Migration seiner virtuellen Festplatten bei Format-Updates unterstützt, eine Aktualisierung der Betriebssystemtreiber muss jedoch zusätzlich erfolgen.

sind, wird ein vollständiges Set im Installationspaket mitgeliefert.

In der Workstation konnten Linux Varianten bis zum 2.4er Kernel ausgeführt werden. Es gab bis zum Ende Probleme mit den zunehmend verwendeten 2.6er Kernels. Die VMware Grafikkarte wurde Bestandteil der XFree86-Treibersammlung für das X11-Basisystem für grafische Oberflächen. Sound- und Netzwerkkarten, IDE, USB und SCSI der virtuellen Hardware gehören alle zur Grundausstattung der Linux-Kernels der Versionen 2.4 und 2.6.

### 5.3.4 Workstation 4.5

Im Gegensatz zu ihren Vorgängerinnen steht diese Version noch auf der Homepage des Herstellers bereit. Die letzte Ausgabe trägt die Nummer 4.5.3, Build 19414 und stammt vom 27. Dezember 2006. Mit dieser Version änderte sich das Containerformat erneut (Abb. 5.1). Die Lizenz wurde jedoch beibehalten, so dass die Software mit einem Schlüssel der 4.0 betrieben werden kann.

Die unterstützten Nutzungsumgebungen haben sich nicht wesentlich geändert, jedoch funktionierten nun Linux-Distributionen mit 2.6er Kernel korrekt. Eine weitere für die Langzeitarchivierung vielleicht nur am Rande interessante Neuerung bestand in der Einführung der Netzwerk-Boot-Fähigkeit. Dazu war das enthaltene Phoenix BIOS um ein zur virtuellen Ethernet-Karte passendes PXE erweitert worden.

### 5.3.5 Workstation 5.0

Die wesentliche Neuerung dieser Version bestand in der Einführung der sogenannten "Shared Folder". Diese erlauben für bestimmte Nutzungsumgebungen einen direkten Datenaustausch zwischen Host- und Gastsystem ohne den Umweg über Wechseldatenträger oder virtuelle

Netzwerkverbindungen.<sup>29</sup>

Die letzte veröffentlichte Version ist die 5.0 in der Revision 13124 von Anfang April 2005. Mit ihrer Einführung wurde eine neue Lizenz zum Betrieb erforderlich.

### 5.3.6 Workstation 5.5

Mit dieser Version wurde das nach der Workstation 5.0 eingeführte Produkt VMware Player gemeinsam paketierte. Das bedeutet, dass sich Workstation und Player der passenden Version gemeinsam installieren lassen und diese ein gemeinsames Set an Kernel-Treibern nutzen. Die aktuelle Version 5.5 stammt von April 2005, das letzte Update erfolgte im September 2007. An der virtuellen Hardware änderte sich nichts.<sup>30</sup> Es wurde gegenüber der Vorgängerversion keine neue Lizenz erforderlich.

### 5.3.7 Workstation 6.0

VMware Workstation 6.0 ist die vorerst jüngste Ausgabe der Workstation, die nun auch unter Windows Vista ablauffähig ist. Hinzugekommen ist eine spezielle Programmier- und Debuggingsschnittstelle VIX API 2.0, die in erster Linie für Softwareentwickler von Interesse ist, da sie beispielsweise Module für das Werkzeug Eclipse<sup>31</sup> enthält. Sie erlaubt es, Applikationen von "außen" zu starten, was für eine Integration in automatisierte Umgebungen interessant werden könnte. Mit der aktuellen Version muss eine erneute Lizenz für den Betrieb erworben werden.

Es gab einige Aktualisierungen an der virtuellen Hardware, die nun je nach Konfiguration einer Nutzungsumgebung die Version 6 trägt. In dieser Version wurde die Einbindung der virtuellen Block-Devices aktualisiert und es kamen eine Multimonitorunterstützung, der USB 2.0 Support<sup>32</sup> sowie eine Mehrprozessorfähigkeit hinzu. Die Nutzungsumgebungen können nun maximal 8 GByte virtuellen Speicher belegen und mit bis zu zehn Netzwerkadaptern arbeiten. Die Workstation kann nun eine Reihe von 64 bit-Betriebssystemen, darunter auch Windows Vista, ausführen.<sup>33</sup>

Es wurde eine Art Headless-Mode eingeführt, der es gestattet, nach dem Start der grafischen Benutzeroberfläche, diese komplett in den Hintergrund zu schieben. Zudem ist ein Remote-Zugriff per VNC eingebaut, der jedoch für eine korrekte Tastaturlayoutübersetzung konfiguriert werden muss.<sup>34</sup>

<sup>29</sup>Dazu müssen mit den VMware Tools spezielle Treiber innerhalb der Nutzungsumgebungen installiert sein. Es werden bestimmte Verzeichnisse im Host-Dateisystem freigegeben, welche sich innerhalb des Gasts mit festlegbaren Zugriffsregeln einbinden lassen. Ältere Nutzungsumgebungen wie DOS, Windows 3.X oder Windows 98 profitieren nicht von dieser Option. Siehe zudem Abschnitt 6.8.

<sup>30</sup>Einige neuere Nutzungsumgebungen wurden nun besser unterstützt; trotzdem gab es mit aktuelleren Linux Kernen immer mal wieder Performance-Probleme, so mit der OpenSuSE10.2.

<sup>31</sup>Eclipse ist eine in Java geschriebene Entwicklungsumgebung, die ursprünglich von IBM initiiert wurde und nun als Open Source Software weiterentwickelt wird.

<sup>32</sup>Die Schnittstelle ist als EHCI angelegt, so dass generische Treiber eingesetzt werden können, die in den neueren Nutzungsumgebungen typischerweise bereits Bestandteil der Betriebssysteminstallation sind.

<sup>33</sup>Für dieses bringt sie sogar in den VMware Tools signierte Treiber mit.

<sup>34</sup>Dieses Problem tritt nicht nur an dieser Stelle auf, sondern ist eine generell zu berücksichtigende Schwierigkeit von Referenzumgebungen, wie Abschnitt 6.5 zeigt.

### 5.3.8 VMware Player

Die erste Version des VMware Players erschien zeitlich kurz nach der Veröffentlichung der Workstation 5.0. Es wird als kostenloses Einzelprodukt vermarktet und ist seit Version 5.5 zusätzlich Bestandteil des Workstation-Pakets. Der Player eignet sich dazu, bereits vorbereitete Nutzungsumgebungen auf dafür geeigneten Referenz-Workstations<sup>35</sup> auszuführen.

### 5.3.9 VMware Server

Der VMware Server ist aus dem ursprünglich als GSX-Server vermarktetem Produkt hervorgegangen. Anders als sein Vorgänger erfordert er zu seiner Benutzung keine kostenpflichtige Lizenz mehr. Wie die anderen VMware Produkte auch handelt es sich um eine Closed-Source-Software. Der VMware Server lässt sich zwischen den Desktop-Produkten, wie dem Player, der Workstation oder Fusion auf der einen Seite und dem auf große Rechenzentren ausgelegten ESX-Server auf der anderen einordnen. Er erlaubt die Trennung der virtuellen Maschine von ihrer grafischen Benutzeroberfläche. Typischerweise laufen auf einer geeigneten X86-Server-Maschine mehrere Nutzungsumgebungen nebeneinander, die sich die verfügbaren Hardwareressourcen teilen. Die Administration des Servers kann weitgehend mittels eines Web-Interfaces erfolgen. Diese Schnittstelle stellt zudem die benötigte Softwarekomponente zum Herunterladen bereit, die für den Fernzugriff auf die einzelnen Konsolen eingesetzt wird. Neben der Remote-Steuerung steht mit "vmware-cmd" ein Kommandozeilenprogramm zur Bedienung einzelner virtueller Maschinen zur Verfügung.

Die vorgenommene Trennung sorgt dafür, dass der Installations- und Einrichtungsaufwand für den zugreifenden Endanwender gering bleibt. Alle komplexeren Vorgänge können von Fachpersonal zentral vorgenommen werden. Jedoch beschränken sich die Interaktionsmöglichkeiten des entfernten Benutzers auf Tastatur- und Mauseingaben und die Betrachtung der Bildschirmausgaben. Ein Transport von Objekten, Audioein- und -ausgaben oder die Verbindung lokaler Wechseldatenträger sind auf diesem Wege nicht möglich. Weitergehende Alternativen hierzu diskutiert Abschnitt 6.5.3.

Diese Vorgehensweise könnte es Gedächtnisorganisationen erlauben, bestimmte Dienste bequem über ein Computernetz anzubieten, ohne dafür lizenzpflichtige Software oder rechtlich geschützte Primärdaten aus ihren Beständen an Endbenutzer übertragen zu müssen.

VMware Server, Player, Workstation und Fusion teilen sich ein gemeinsames, untereinander kompatibles Containerformat. Ebenso sind die virtuelle Hardware und die Konfigurationsdateien zwischen den Produkten weitgehend identisch und damit in gewissem Umfang austauschbar. Der VMware Server der Version 1, die erste offizielle Ausgabe erschien im Oktober 2006 und entspricht in seiner virtuellen X86-Hardware der Ausstattung der Workstation 5.5. Die neue Version 2 orientiert sich an der Workstation 6.0.

### 5.3.10 VMware Fusion

"Fusion" erweitert die VMware Produktpalette auf eine weitere Referenzplattform, die Intel-basierten Apple. Das lizenzpflichtige Produkt erschien Mitte 2007 auf dem Markt und ist von

---

<sup>35</sup>Da es sich um ein technologisch zur Workstation identisches Produkt handelt, kann es auf identischen Host-Systemen ablaufen.

seiner Zielgruppe her mit der der traditionellen Workstation vergleichbar. Die technologische Basis unterscheidet sich nicht, so dass sich Containerdateien zwischen den verschiedenen Produkten austauschen lassen.<sup>36</sup> Ein interessanter Aspekt taucht mit einer ersten Unterstützung von DirectX 8.1 auf. Damit ist Fusion das erste VMware-Produkt mit 3D-Unterstützung einiger Gastsysteme.<sup>37</sup> Fusion läuft auf Intel-CPU basierendem Apple Mac-OS X ab der Version 10.4.8.

## 5.4 Weitere kommerzielle Virtualisierer

Wie in vielen anderen Bereichen des Softwaremarkts zu beobachten, rief der Erfolg von VMware weitere Mitbewerber auf den Plan. Diese mussten, um sich gegen den Vorreiter durchsetzen zu können, den Anwendern Argumente, wie einen geringeren Preis oder neue Funktionalität liefern. Die Konkurrenz im Bereich Virtualisierung hat das Feld beflügelt. Durch eine höhere Auswahl an Produkten steigt die Wahrscheinlichkeit, Nutzungsumgebungen geeignet wiederherstellen zu können. Die nachfolgend vorgestellten Produkte spielen eher in spezielleren Szenarien, in denen es auf eine bestimmte Referenzumgebung oder alternative Formen des Datenaustauschs ankommt, eine Rolle. Deshalb werden sie etwas kürzer behandelt.

### 5.4.1 Parallels Workstation und Desktop

SWsoft stieg recht spät in den Markt ein, weshalb das Produkt bisher noch nicht auf eine längere Historie zurückblicken kann. Interessant ist daher am ehesten die Ausrichtung auf drei X86-Referenzumgebungen. Das Produkt kann damit im üblichen Feld mithalten, bietet aber wegen seiner proprietären Lizenz und den damit verbundenen Kosten keine besonderen Vorteile für eine Langzeitarchivierung.

**Interaktion, Datenim- und -export** "Workstation" und "Desktop" nutzen ein identisches Containerformat für ihre virtuellen Festplatten, welches jedoch nicht zu anderen Virtualisierern kompatibel ist. Die üblichen virtuellen Disketten und optischen Datenträger im ISO-Format lassen sich einsetzen.

**Unterstützte Nutzungsumgebungen** Parallels emuliert als Basis ein Intel 430TX Mainboard, eine Plattform, die mit dem Pentium-Prozessor recht weit verbreitet war. Tastatur und Maus werden direkt aus der Host-Plattform durch die üblichen APIs für Applikationen übernommen. Die Maus wird dabei über eine PS/2 Schnittstelle abgebildet. Die Software unterstützt das Einbinden von 1.44 MByte Diskettenlaufwerken, entweder als physikalisches Gerät direkt von der Host-Plattform oder als Image-Datei der entsprechenden Größe.

Ebenso werden Festplatten, CD- oder DVD-Laufwerke behandelt. Bis zu vier Geräte können über das virtuelle IDE-Interface angebunden werden. Die optischen Laufwerke unterstützen ISO-Images oder physikalische Laufwerke, haben wegen der proprietären, an das

<sup>36</sup>Hierbei spielt die Revision der virtuellen Hardware eine gewisse Rolle, da Treiber und VMware Tools passen sollten, um eine Reininitialisierung zu vermeiden.

<sup>37</sup>Es erfolgt eine Übersetzung in 3D-Aufrufe des Host-Systems. Die Funktionalität ist derzeit auf Windows bis XP beschränkt, OpenGL und auch das neuere DirectX 9 von Windows Vista sind nicht ausführbar.



DVD-Laufwerk gebundenen Sicherheitsarchitektur, jedoch keinen Zugriff auf Video-DVDs. Die virtuellen Festplatten können eine Kapazität von 20 MByte bis 128 GByte als Image-Dateien aufweisen. Die Netzwerkkarte ist mit der RTL8029 zur ehemals weit verbreiteten generischen NE2000/PCI kompatibel. Die Grafikausgabe erlaubt den Nutzungsumgebungen VGA und SVGA mit VESA 3.0 Unterstützung bis zu Auflösungen der Host-Plattform.<sup>38</sup> Die virtuelle Soundkarte ist AC97-kompatibel.

Diese "Hardwareausstattung" gestattet mit der zugrundeliegenden Software den Betrieb von etlichen der üblichen X86-Betriebssystemen.

### 5.4.2 Virtual PC und Server für X86

Microsoft sprang verhältnismäßig spät auf den Virtualisierungszug auf. Anfangs wurde das Projekt recht ehrgeizig verfolgt und VMware mindestens über die Lizenzkosten durchaus Konkurrenz gemacht. Virtual PC wurde mithilfe von Innotek [innotek AG 2008] von der MacOS Version von Connectix portiert. Nach der Desktop-Version wurde 2005 mit dem Microsoft Virtual Server 2005 ein dem VMware GSX und aktuell VMware Server vergleichbares Produkt auf den Markt gebracht [Thorns 2007], Kapitel 10. Seit 2006 sind Virtual PC und Server kostenlos verfügbar.

**Generelle Metadaten** Virtual PC und Server nutzen die identische technologische Basis, so dass die Container der virtuellen Festplatten im gewissen Umfang untereinander austauschbar sind. Das "Server" Produkt unterstützt jedoch zusätzlich einen virtuellen SCSI-Controller für Festplatten und optische Wechsellaufwerke. Virtual Server verfügt über kein klassisches Benutzer-Interface sondern ist über ein Web-Interface zu steuern. Im Internet Explorer, andere Web-Browser sind quasi nicht nutzbar, lässt sich auf ein Client-System per ActiveX-Control zugreifen.

Die Produktpflege fiel mit der aktuellen Ausgabe Virtual PC 2007 eher halbherzig aus. Nach einer Pause von fast drei Jahren nach Version 2004 hätte man mehr als das Update auf neue Host-Umgebungen, wie Vista und 64 bit Maschinen erwartet. Dieser Virtualisierer scheint eher für das Ausführen älterer Nutzungsumgebungen und für in neueren Betriebssystemen nicht mehr lauffähige Applikationen geeignet zu sein. Angekündigte Produkte, wie Viridian, streben für die Zukunft einen stärkeren Fokus auf Serverkonsolidierung an.

**Nutzbarkeit, Zuverlässigkeit, Portabilität** Die eingeschränkte Auswahl an Referenz- und möglichen Nutzungsumgebungen macht das Produkt nur für Institutionen interessant, die lediglich Microsoft-Betriebssysteme eingesetzt haben und einsetzen wollen. In diversen Tests zeigte sich, dass auch Nicht-Microsoft-Betriebssysteme installiert werden konnten und funktionierten. Da sowohl diese als auch Installationen von offiziell unterstützten Systemen sehr lange dauerten und die Performance scheinbar gegen Virtualisierer wie VMware oder VirtualBox abfiel, wurde dieser Weg nicht weiter verfolgt. Das in der aktuellen Version mitgelieferte Portierungswerkzeug erlaubt es virtuelle VMware-Festplatten, in das hauseigene Format umzuwandeln.

<sup>38</sup>Hier bestehen Unterschiede zwischen "Workstation" und "Desktop". Für das Produkt der Apple X86-Plattform wird eine 3D-Unterstützung angeboten.

**Anforderungen an das Host-System** Virtual PC ist nicht nur auf Microsoft Windows (2000, XP, Vista) festgelegt, was das Host-System anbetrifft, sondern auch ziemlich beschränkt bezüglich der ausführbaren Nutzungsumgebungen. Die Version 2004 des Produkts gibt es lediglich in einer 32 bit-Ausführung, die aktuelle Version 2007 in 32 und 64 bit.

Während der Virtual PC auf typische Endbenutzer- und Desktop-Systeme fokussiert, ist der Virtual Server lediglich für die diversen Windows 2003 Server und als Ausnahme Windows XP freigegeben.

**Interaktion, Datenim- und -export** In reinen Microsoft-Umgebungen bietet Virtual PC, neben dem klassischen Austausch mittels virtueller Datenträger und Netzwerkverbindungen, eine interessante Funktionalität: Objekte können per Drag&Drop zwischen Host-System und Referenzumgebung verschoben werden. Das entspricht dem Credo des Herstellers, seine älteren Systeme weiter zu unterstützen und möglichst nahtlos zu integrieren. Der Virtual Server beherrscht diese für den Nutzer einfache Form des Datenaustauschs jedoch nicht.

**Unterstützte Nutzungsumgebungen** Die virtuelle Hardware bietet das gewohnte Bild einer Standard-Intel-Plattform basierend auf dem Ende der 1990er Jahre sehr populären und lange eingesetzten 440BX Chipset. Damit stehen insgesamt vier virtuelle IDE-Anschlüsse für Festplatten als Disk-Images oder optische Laufwerke bereit. Als BIOS kommt wie in der Mac-OS-Variante ein MRBIOS von Microid Research zum Einsatz. Die Grafikkarte ist eine S3 Trio32/64 VGA, die ebenfalls weit verbreitet war und für viele Nutzungsumgebungen eine gute Treiberversorgung bietet.<sup>39</sup> Als Audioschnittstelle wird eine Soundblaster 16 PCI verwendet. Der Virtual Server bietet keine Soundkarte an.

Wie bei den anderen X86-Virtualisierern können virtuelle<sup>40</sup> oder reale Diskettenlaufwerke verbunden werden. Das gilt ebenfalls für physikalisch vorhandene CD/DVD-Laufwerke oder deren virtuelles Pendant als ISO-Image. Bei der virtuellen Netzwerkkarte handelt es sich um eine DEC21140 (Tulip) 100 MBit/s Variante, die längere Zeit von mehreren Herstellern angeboten wurde. Eine SCSI-Unterstützung ist nur im Serverprodukt vorgesehen.

Anders als bei den meisten Virtualisierern ist keine USB-Anbindung vorhanden; was jedoch nur für eingeschränkte Benutzergruppen überhaupt interessant sein dürfte. Bezüglich der virtuellen Hardware erfolgten für Virtual PC 2007 keine Änderungen, wodurch ein Treiber-Update der Nutzungsumgebungen vermieden wird.

## 5.5 Open-Source-Virtualisierer

Im Open-Source-Bereich gab es bereits recht früh parallel zu den ersten VMware Workstations erste Versuche, eine freie Alternative zu programmieren: Das Plex86 Projekt erlebte eine Phase der Aufmerksamkeit, bis es nach einiger Zeit wieder aufgegeben wurde. Ein stabiles Stadium wurde nicht erreicht. Da Virtualisierung insbesondere für Technologiefirmen und

<sup>39</sup>Damit setzt Virtual PC anders als VMware oder VirtualBox auf die Nachbildung realer Hardware. Das verbessert die Situation der VGA-Treiber höherer Auflösungen für alte Nutzungsumgebungen, wie Windows 3.X.

<sup>40</sup>Via Disketten-Image im standardisierten Format, wie es beispielsweise "dd" erzeugt.

Softwareentwickler zunehmend wichtig wurde, blieben neue Ansätze nicht aus. Es bildeten sich mehrere Zweige für Neuentwicklungen heraus:

- Forschungsinstitute, wie die Universität Cambridge traten mit neueren Ansätzen, wie XEN<sup>41</sup> an, die versuchen den zwangsläufigen Leistungsverlust durch die eingeschobene Abstraktionsschicht des Hypervisors deutlich zu verringern. Das stellt jedoch besondere Anforderungen, wie die Anpassung bestimmter Codeblöcke, an die Betriebssysteme der Nutzungsumgebungen, die insbesondere für die älteren nicht gegeben sind. Zudem spielen gerade bei diesen Nutzungsumgebungen Performance-Aspekte wegen des geringen Leistungsbedarfs eine untergeordnete Rolle.
- Kommerzielle Firmen, die eine Open Source Software als Auftragsprodukt herausbringen und damit den Markt der auf den PC-Desktop zielenden Virtualisierer erneut in Bewegung setzen.
- Reine Open-Source-Projekte und Einzelentwickler, die beispielsweise mit QEMU eine zunehmend relevante weitere Software in diesem Bereich platzieren.

QEMU ist in erster Linie ein Emulator, kann aber durch ein Kernel-Modul<sup>42</sup> zu einem klassischen Virtualisierer erweitert werden. Neuere Linux Kernel erlauben die Nutzung der Virtualisierungstechniken der aktuellen AMD und Intel Prozessoren durch ein spezielles Linux Kernel-Modul.<sup>43</sup> Der Emulator ist modular angelegt und kann neben der X86-Architektur weitere nachbilden.

### 5.5.1 VirtualBox

Der Erfolg von VMware hat einen interessanten Mitspieler und Nachahmer auf das Feld gerufen: Die Firma Innotek<sup>44</sup> sammelte bereits Erfahrungen auf dem Gebiet der Virtualisierung bei der Portierung und Programmierung von Virtual PC. Sie kam Anfang 2007 mit einer Open-Source-Version eines neuen Virtualisierers auf den Markt. Als Späteinsteiger in die Gruppe der direkten Akteure in diesem Geschäftsfeld brachte die Open-Source-Lizenz sicherlich die notwendige Aufmerksamkeit, um in der Nutzergemeinde positiv wahrgenommen zu werden. Das grafische Benutzer-Interface orientiert sich klar an dem Gewohnten für den Desktop und den Bereich der Softwareentwicklung.

**Generelle Metadaten** Das erste öffentliche Release erfolgte Anfang 2007 in der Version 1.2. In der Zwischenzeit erschienen weitere Versionen mit stetigen Verbesserungen und neuen Funktionen.

Als Virtualisierer versucht die Software, so viel Programmcode wie möglich nativ auf der CPU auszuführen. Der Ring-3-Code der Nutzungsumgebung wird überwiegend direkt an das Host-System durchgereicht. Stehen in der Nutzungsumgebung Ring-0-Operationen an, werden diese im Host-System im Ring ausgeführt, der üblicherweise nicht in Benutzung

<sup>41</sup>(Para)Virtualisierung, siehe - <http://www.cl.cam.ac.uk/research/srg/netos/xen>.

<sup>42</sup>"kqemu.ko, siehe hierzu [Bellard 2008] und [Thorns 2007], S. 271.

<sup>43</sup>KVM - Kernel-Based Virtual Machine, siehe [QUMRANET 2007].

<sup>44</sup>Vgl. [innotek AG 2008]. Innotek wurde im Februar 2008 von SUN Microsystems übernommen.

ist. Lässt sich dieses nicht realisieren, wird ein Emulator eingeschaltet, der den Code nativ ausführt. Dieser nutzt wiederum die Quellcodebasis von QEMU (Abschnitt 5.6.1).

**Lizenzen und Kosten** Der Quellcode des Projekts steht unter GPL. Es steht eine spezielle kompilierte Version unter einer speziellen Lizenz.<sup>45</sup> Laut Hersteller sollen neue, anfangs proprietäre Funktionen von der kommerziellen Ausgabe schrittweise in die Open-Source-Version übernommen werden.

**Nutzbarkeit, Zuverlässigkeit, Portabilität** VirtualBox verwendet ein eigenes, inkompatibles Format für seine Festplattendateien. In aktuellen Versionen ist es darüber hinaus in der Lage, mit VMware VMDK-Containern umzugehen. Die kommerzielle Version erlaubt es, zusätzlich iSCSI-Targets als virtuelle Festplatten einzubinden.

Für die virtuelle Maschine existieren in den neueren Versionen mehrere Frontends:

- Ein Konsolenprogramm mit dem Namen VBoxManage, welches über eine Reihe von Optionen und Parametern gesteuert werden kann. Dieses eignet sich daher für die Einbindung in automatisierte Umgebungen.
- Eine grafische Benutzerschnittstelle - VirtualBox, die sich am Gewohnten dieser Softwareklasse orientiert und auf der Grafikbibliothek Qt basiert. Diese Schnittstelle realisiert noch nicht alle Optionen, die mit dem Konsolenprogramm umgesetzt sind.
- Ein Programm (VBoxSDL), welches die SDL-Bibliothek<sup>46</sup> nutzt.
- Und ein Remote-Desktop-Protocol-Server (VBoxVRDP), der in der Konsole abläuft. Auf diesen kann mittels sogenanntem Session-Shadowing von mehreren Clients aus gleichzeitig zugegriffen werden.

Die Applikation kann in einem Vollbild- und Fenstermodus ausgeführt werden.

**Anforderungen an das Host-System** Die virtuelle Maschine kann auf 32 bit- und 64 bit X86-Systemen mit den Host-Betriebssystemen Linux der Kernelversionen 2.4 und 2.6, Windows 2000, XP, Vista oder Mac-OS X genutzt werden. Eine Parallelinstallation mit anderen Virtualisierern, wie VMware Workstation, Player oder QEMU ist möglich.

Für unveränderliche Container setzt VirtualBox auf eine andere Strategie als VMware mit dem "nonpersistent mode": Von der "immutable" Disk wird jeder Block in eine spezifizierbare separate Datei kopiert. Auf diese Weise wächst eine Cache-Datei mit benutzten Blöcken bis auf die Größe des Ausgangscontainers an.

<sup>45</sup>PUEL - VirtualBox Personal Use and Evaluation License, siehe [innotek AG 2008], speziell [http://www.virtualbox.org/wiki/VirtualBox\\_PUEL](http://www.virtualbox.org/wiki/VirtualBox_PUEL). In der vorkompilierten Version sind Komponenten enthalten, die nicht mit dem Sourcecode verbreitet werden.

<sup>46</sup>Vgl. hierzu Abschnitt 7.7.1.

**Interaktion, Datenim- und -export** Wie die neueren VMware Versionen kennt VirtualBox "Shared Folders": Damit können zwischen Host-System und Nutzungsumgebung Ordner gegenseitig freigeben werden, was einen einfacheren Datenaustausch als über Netzwerk oder virtuelle Laufwerke erlaubt. Diese Fähigkeit steht nur für ausgewählte Nutzungsumgebungen zur Verfügung.<sup>47</sup>

**Unterstützte Nutzungsumgebungen** VirtualBox basiert auf einer Intel 440FX<sup>48</sup> Plattform mit zwei virtuellen IDE-Controllern und einem integrierten nicht-interaktivem BIOS.<sup>49</sup> Ebenso wie VMware kann mittels PXE über eine virtuelle Netzwerkkarte gebootet werden. Insgesamt lassen sich bis zu vier Netzwerkkarten vom Typ AMD PCnet32<sup>50</sup> konfigurieren. An den IDE-Controller können bis zu vier virtuelle Festplatten oder alternativ CD- bzw.



**Abbildung 5.4:** VirtualBox im Fenstermodus mit einem IBM-DOS 4.00. Der Virtualisierer läuft auf einem X86-Linux-Host und bootet eine virtuelle Diskette.

DVD-Laufwerke angeschlossen werden. Letztere lassen sich direkt aus dem Host-System importieren oder per ISO-Image abbilden. Ähnliches gilt für Diskettenlaufwerke - Zugriff auf ein physisches Gerät oder eine passende Image-Datei. Der Audio-Controller ist ein emulierter Intel AC97 (82801AA). Es kann auf das USB-Subsystem des Host-Systems zugegriffen

<sup>47</sup>Es müssen passende Treiber in der Nutzungsumgebung installiert werden. Diese werden für alte Umgebungen nicht mehr erstellt.

<sup>48</sup>Exakt 82441FX PMC Natoma - Es wird ACPI- und I/O-APIC-Unterstützung angeboten.

<sup>49</sup>Es sind keine klassischen BIOS-Einstellungen, wie von traditionellen BIOSes gewohnt vorzunehmen. Die Boot-Geschwindigkeit vergleichsweise zu anderen Virtualisierern sehr hoch. Die Bestimmung der Boot-Reihenfolge kann im Setup der Software oder durch Druck auf die F12-Taste dauerhaft oder einmalig festgelegt werden.

<sup>50</sup>79c97X Chip. Hierbei handelt es sich um eine sehr populäre Erweiterungskarte, die beispielsweise auch in VMware Workstation oder QEMU nachgebildet wird.

werden. USB-Geräte können dynamisch einer Nutzungsumgebung zugewiesen und an das Host-System zurückgegeben werden.

Bei der Grafikkarte handelt es sich um einen speziellen Videoadapter, dessen Speicher von 1 bis 128 MByte festgelegt werden kann. Die Bildschirmauflösung ist je nach Nutzungsumgebung auf 800x600 oder 1024x768 Punkte beschränkt, kann aber durch entsprechende Treiber aus den Gasterweiterungen nachträglich vergrößert werden. Dabei sind Grafikauflösungen von maximal 64000x64000 Pixeln in 32 bit Farbtiefe möglich. Auf diese Weise lässt sich ein Mehrschirmbetrieb realisieren. Die Grafikauflösung der Nutzungsumgebung passt sich an die Fenstergröße im Host an. Die Tastatur wird 1:1 übersetzt, wobei es eine spezielle Escape-Taste zum Wechsel aus der Gastumgebung gibt. Es stehen keine parallelen, seriellen Schnittstellen oder SCSI-Host-Controller zur Verfügung.

Die Palette der Gastsysteme ist breit: Linux der Kernel 2.2 bis 2.6, diverse BSD-Varianten, Netware, Solaris, DOS und Windows 3.X, 95, 98, ME sowie Windows 2000, 2003, XP (Abb. 5.4). Zusätzlich wird OS/2 unterstützt.<sup>51</sup>

## 5.5.2 DOSEMU

DOSEMU<sup>52</sup> zählt zu den älteren Virtualisierern.<sup>53</sup> Die erste Version wurde 1992 veröffentlicht. In der Zwischenzeit erschienen vielfältige Aktualisierungen, das Projekt ist immer noch aktiv. Die Software steht unter der GPL.

DOSEMU nutzt den vorhandenen Prozessor, so dass die DOS-Programme direkt ausgeführt werden. Der dadurch erzielte Geschwindigkeitsgewinn spielt für aktuelle Host-Systeme eigentlich keine Rolle mehr. Durch dieses Verfahren können zudem Probleme, wie zu schnell ablaufende Programme auftreten.

DOSEMU funktioniert aufgrund der direkten CPU-Nutzung nur auf Systemen mit Intel X86-kompatibler CPU. Die virtuelle Hardware umfasst eine Art BIOS für grundlegende Funktionen. Die Tastatur- und Mauseingaben werden direkt übersetzt. Dabei lassen sich verschiedene Tastaturlayouts einstellen. DOSEMU kann mit virtuellen Festplatten und Diskettenlaufwerken umgehen. Einige typische ältere Hardware wird emuliert oder kann direkt mit den Geräten des Host-Systems hindurchgereicht werden.

Nutzungsumgebungen beschränken sich auf DOS-Applikationen bis hin zu Windows 3.X. Aufgrund der genannten Einschränkungen, einer nicht besonders guten Dokumentation und einer recht kleinen Benutzer-Community erscheint diese Software nur für wenige Einsatzszenarien sinnvoll. Eine interessante Alternative besteht in der DOSBOX (Abschnitt 5.6.5).

<sup>51</sup>Dieses verwendet mehr als die typischen zwei Ringe der CPU und stellt deshalb besondere Anforderungen an den Virtualisierer.

<sup>52</sup>Siehe Projekt-Homepage - <http://www.dosemu.org>, die Version 1.4, die zusätzlich auf 64 bit X86-Systemen ablauffähig ist, erschien Mitte 2007.

<sup>53</sup>Es handelt sich um eine Mischung aus Betriebssystem- und Hardwarevirtualisierer. Die typische Peripherie wird virtualisiert, CPU-Kommandos weitgehend durchgereicht. Es sind lediglich bestimmte Nutzungsumgebungen, wie DOS oder Windows 3.X ausführbar.

## 5.6 X86-Hardwareemulatoren

Die Entwicklung von Emulatoren für die X86-Architektur ist noch relativ neu. Der PC wurde zur Standardmaschine im Laufe der 1990er Jahre. Deshalb gab es Anfangs nur Randbereiche, wie die verschiedenen Macintosh-Architekturen, wo ein Bedarf bestand, X86-Software ablaufen zu lassen. Deshalb gehörte der Virtual PC für PowerPC Apples zu den ersten verbreiteten Emulatoren der X86-Architektur überhaupt. Mit der steigenden Leistungsfähigkeit der PCs einerseits und der zunehmenden Anzahl veralteter Betriebssysteme andererseits entstand der Bedarf nach Virtualisierung und Emulation dieser Architektur. In diesem Bereich finden sich inzwischen mehrere Ansätze. Im Gegensatz zur Virtualisierung steht nicht in erster Linie Performance der Ausführung von Nutzungsumgebungen im Vordergrund, sondern die Architekturunabhängigkeit.

### 5.6.1 QEMU

Zu den zweifellos interessantesten Emulatoren muss QEMU gerechnet werden. Betreut und weiterentwickelt wird das Projekt von Fabrice Bellard.<sup>54</sup> Inzwischen hat sich eine große Benutzer- und Entwicklergemeinschaft gebildet.

QEMU wurde primär als Emulator für verschiedene Rechnerarchitekturen angelegt. Für die X86-Plattform sind jedoch inzwischen Varianten zur Virtualisierung verfügbar. Die traditionellere arbeitet mit einem speziellen Kernel-Modul, welches ähnliche Aufgaben wie bei den VMware Produkten hat, nämlich privilegierte Zugriffe im Host-System zu ermöglichen. Für den Einsatz mit den Intel oder AMD CPU-Virtualisierungstechniken wurde eine Variante von QEMU abgeleitet, die auf die Linux KVM<sup>55</sup> zugreifen kann. Diese Entwicklungen sind eher unter Performance- denn Langzeitarchivierungsgesichtspunkten interessant.

**Generelle Metadaten** Das Projekt ist Open Source und steht unter der GPL. Die Quelldaten sind damit frei verfügbar. Die Version 0.9.1 ist seit Januar 2008 verfügbar. Zudem befinden sich die Versionen 0.8.2 und 0.9 im Umlauf. Beim Versionswechsel wurde die Menge der emulierten Peripherie erhöht und es kam ein neues Copy on Write Disk Image Format der Version 2 hinzu.

Das QEMU-Paket besteht neben dem Basisprogramm "qemu" aus mehreren ausführbaren Dateien, die der jeweils emulierten Zielplattform entsprechen *qemu-mipsel*, *qemu-arm*, *qemu-ppc*, *qemu-archeb*, *qemu-sparc*, *qemu-mips*. Neben diesen Programmen ist ein Werkzeug zum Transformieren von Containerformaten enthalten: "qemu-img".

QEMU bietet einen Fenster- und Vollbildmodus an. Zudem lässt sich die Steuerung durch grafische Benutzerschnittstellen, die für Windows und Linux von Dritten angeboten werden,<sup>56</sup> vornehmen. Der Emulator kann darüber hinaus ein VNC-Server-Interface anbieten, welches ähnlich wie der VMware oder Virtual Server einen grafischen Remote-Zugang über Netzwerk erlaubt.

<sup>54</sup>Siehe News-Bereich der Projekt-Homepage [Bellard 2008].

<sup>55</sup>Kernel Virtual Machine, siehe - <http://kvm.qumranet.com/kvmwiki>.

<sup>56</sup>Eine solche grafisch orientierte Applikation ist der "qemu-launcher".

**Nutzbarkeit, Zuverlässigkeit, Portabilität** QEMU ist eine Applikation ohne grafisches Benutzer-Interface. Es stellt eine Reihe von Kommandozeilenoptionen<sup>57</sup> 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 zählt 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 Mausektionen mittels Monitor-Interface triggern. So können bestimmte Tastenkombinationen übermittelt werden, die typischerweise von der Referenzumgebung abgefangen werden.<sup>58</sup> Auf diese Weise wären ganze Programmabläufe automatisierbar, die sonst eine direkte Benutzerkommunikation erfordern:

- Skriptgesteuerte Installation eines Betriebssystems wie DOS mit einem Set mehrerer virtueller Datenträger. Aufgaben, wie das Anlegen von Partitionen oder eine Formatierung des Datenträgers, können durch eine Sequenz von Tastatureingaben abgebildet werden.
- Ebenso ließen sich Primärobjekte aus virtuellen optischen oder Diskettenlaufwerken aufnehmen und in einer bestimmten Applikation öffnen.
- Selbst Abläufe, die Mausektionen erfordern, können nicht-interaktiv ausgeführt werden, wenn Fensterplatzierungen und ähnliche Aktionen der Nutzungsumgebung sicher vorhersagbar sind.

Weitere Anwendungen und Beispiele finden sich in [Welte 2008]. Zuguterletzt kann via Monitor-Interface die virtuelle Maschine in einen Snapshot gesichert oder aus diesem geladen werden, die Emulation gestoppt und wieder angefahren sowie die Maschine beendet oder rebootet werden.

QEMU kann im Emulations- oder Virtualisierungsmodus laufen. Letzteres ist auf zwei Wegen möglich: Entweder durch Einsatz eines Kernel-Moduls oder durch Verwendung des Linux KVM und des hierfür angepassten Softwarepakets.<sup>59</sup> Oder durch den QEMU-Accelerator KQEMU.

Das BIOS und seine Erweiterungen (VGA und ROM-Extensions wie (g)PXE) sind als separate Dateien abgelegt und können daher bei Bedarf leicht ausgetauscht werden.

**Anforderungen an das Host-System** Der Emulator kann für die Host-Betriebssysteme Linux, Windows, FreeBSD, NetBSD, OpenBSD und Mac-OS X übersetzt werden. QEMU

<sup>57</sup>Beispiel für den Start des Emulators mit PCI-Sound-, AMD PCnet32 Netzwerkkarte, virtuellem CD- und Diskettenlaufwerk: `qemu -soundhw es1370 -net nic,model=pcnet -cdrom cd-image.iso container.type`. Siehe zudem (Abb. A.2 232).

<sup>58</sup>Vgl. zu dieser Problematik den späteren Abschnitt 6.5.4: Ein Emulator läuft als Applikation in einem Betriebssystem und wenn er dann noch in einer VNC-Umgebung in einem Browser-Fenster dargestellt wird, sind die Tastatureingaben für die Nutzungsumgebung mehrfach gefiltert.

<sup>59</sup>Siehe [QUMRANET 2007]. Der Betrieb des QEMU/KVM erwies sich zum Testzeitpunkt als deutlich instabiler, verglichen mit der traditionellen Betriebsart. Die Installation der verschiedenen Kernel-Module benötigt erweiterte Zugriffsrechte in der Referenzumgebung. Der reine Emulatormodus kommt ohne diese aus.



benötigt zur Bildschirmausgabe die SDL-Bibliothek,<sup>60</sup> welche Applikationen eine portable Low-Level-Schnittstelle für den Zugriff auf die Videoausgabe, Audiogeräte, Maus und Tastatur erlaubt. Generell stellen mehrere parallel ablaufende Instanzen des Emulators kein Problem dar. Die Beschränkung liegt in der Menge der Gesamtressourcen und der Zuweisungen an einzelne Systeme.<sup>61</sup> Darüber hinaus ist es kein Problem, QEMU gemeinsam mit anderen Virtualisierern oder Emulatoren in einer Referenzumgebung zu betreiben.

**Interaktion, Datenim- und -export** Die Flexibilität von QEMU ist beachtlich: Es werden in der Version 0.9 die Formate: QCOW2, VVFAT, VPC, Bochs, DMG, CLOOP, VMDK, QCOW, COW, Host Device und RAW<sup>62</sup> verstanden. Im Softwarepaket des Emulators ist ein Konvertierungsprogramm "qemu-img" enthalten, welches die Umwandlung verschiedener Containerformate auch anderer Virtualisierer ineinander erlaubt.<sup>63</sup> Mit diesem Programm können zudem Leercontainer der verschiedenen Formate erzeugt und Snapshots<sup>64</sup> verwaltet werden. Neben der Konvertierung durch "qemu-img"<sup>65</sup> kann QEMU beispielsweise auf VMware VMDK-Dateien direkt operieren. Unabhängig vom Containerformat ist QEMU fähig, im nicht-persistenten Modus zu arbeiten, so dass die entsprechende Datei während der Laufzeit nicht geändert wird. Damit zählt QEMU zu den flexibelsten Programmen überhaupt in seiner Klasse. Analog zum "vmware-mount.pl" erlaubt das Kommando "lomount"<sup>66</sup> einzelne Partitionen eines Disk-Images im Host-System zu mounten.

Für den Datenaustausch per IP-Netzwerk beherrscht QEMU zwei verschiedene Modi, die jeweils mehrere virtuelle Netzwerkkarten erlauben. Der User Mode Network Stack wird mit der Option `-net nic,model=...` `-net user` gestartet, der ohne Administratorrechte auskommt. Für eine vollwertige, route-bare Netzwerkverbindung kann durch Aufruf von `qemu ... -net nic -net tap` mit dem TUN/TAP-Interface des Host-Systems kommuniziert werden.

**Unterstützte Nutzungsumgebungen** QEMU erlaubt prinzipiell die Ausführung aller X86-kompatiblen Betriebssysteme. Der QEMU der X86-Variante bildet, analog zu seinen kommerziellen Pendants, mit dem Intel 440FX Mainboard eine populäre Pentium-Plattform nach, die insgesamt zwei IDE-Kanäle mit jeweils bis zu zwei IDE-Geräten ansteuern kann. Das BIOS wird von Bochs übernommen. Die Plattform unterstützt seit Version 0.8.2 ACPI.

Als Grafikadapter steht primär eine virtuelle Cirrus Logic CLGD 5446 PCI VGA zur Verfügung. Sie kann maximal 1280x1024 Pixel mit 16 bit Farbtiefe abbilden. Zudem existiert ein VESA-kompatibler Standard-VGA-Adapter, der bis zu 1600x1200 Bildpunkte mit 24/32 bit

<sup>60</sup>Simple DirectMedia Layer, siehe für weitere Ausführungen Abschnitt 7.7.1.

<sup>61</sup>So konnte schon ein mittlerer AMD64 X2 im 32 bit-Mode mit 2 GByte Hauptspeicher mehr als sieben parallele Sitzungen aus Windows 3.11, Windows 95, 98 Nutzungsumgebungen ablaufen lassen.

<sup>62</sup>Damit kann QEMU die Formate von Virtual PC, Bochs, VMware, "dd"-Output (RAW Disk Container, siehe Abschnitt 7.10) zumindest lesen. COW steht für Copy-On-Write, CLOOP für Compressed Loopback.

<sup>63</sup>"qemu-img" als Werkzeug der Archivierung kompletter Umgebungen: Abb. 6.12 in Abschnitt 6.6.1.

<sup>64</sup>Zwischenabbilder eines Zustands der virtuellen Festplatte zu einem bestimmten Zeitpunkt.

<sup>65</sup>Interessanterweise konnten durch die Konvertierung von VMDK zu VMDK VMware Workstation 3.2 Container, die vorher von der Workstation 6.0 abgewiesen wurden, anschließend erfolgreich eingelesen werden.

<sup>66</sup>Als separates Programm aus dem QEMU-HD-Mounter - <http://www.dad-answers.com/download/qemu/utilities/QEMU-HD-Mounter/lomount> verfügbar.

Farbtiefe anbietet. Beide nutzen ein Plex86/Bochs VGA-BIOS. Die neueste Version 0.9.1 emuliert zusätzlich die VMware SVGA II Grafikkarte. Inzwischen gibt es sogar einen ersten experimentellen Patch<sup>67</sup> für eine OpenGL-Unterstützung. Der Emulator kennt drei verschiedene Arten von Soundkarten: Die einst populären Soundblaster 16, Yamaha YM3812 (OPL2/Adlib) sowie die ENSONIQ AudioPCI ES1370.

Ebenso üppig sieht die Auswahl an Ethernet-Adaptern aus: Zur Verfügung stehen die NE2000 ISA- und PCI-Karten, die auf dem Realtek 8029 Chip basieren, die 100 Mbits Realtek 8139 PCI sowie die schon von anderen Virtualisierern bekannte AMD PCnet32. Seit Version 0.9 ist QEMU in der Lage BIOS-Erweiterungen für den Start über Ethernet-LAN einzubinden und damit beispielsweise PXE-Funktionalität bereitzustellen.

Besonders interessant für sehr alte Betriebssysteme aus der frühen X86-Ära bietet QEMU einen virtuellen Nur-ISA-PC-Modus.<sup>68</sup>

Maus und Tastatur werden vom Host-System eingebunden und auf die gewohnten PS/2-Geräte abgebildet. Optische Laufwerke lassen sich als typisches ISO-Image als virtuelles Gerät oder als reale Hardware einbinden. Bis zu zwei Diskettenlaufwerke<sup>69</sup> sind einstellbar – virtuell als Images oder direkt aus dem Host-System.

QEMU kann bis zu vier separate Festplatten-Images in eigenen Formaten einbinden, ist aber zudem in der Lage, die im vorherigen Abschnitt genannten Containerformate ebenfalls zu nutzen. Serielle und parallele Schnittstellen lassen sich bei Bedarf an die Pendanten der Host-Umgebung umleiten, sind aber für eine Langzeitarchivierung weniger von Interesse.

**Weitere Nutzungsumgebungen** QEMU ist in der Lage neben X86-, X86-64 beziehungsweise AMD64 mit Apple PowerPC- und Sparc 32/64 bit-Hardware<sup>70</sup> umzugehen. Das BIOS ist eine freie, Open-Firmware-kompatible Implementierung.<sup>71</sup>

Weitere Betriebssysteme lassen sich ebenfalls nutzen, wenn ihre Hardwarevoraussetzungen erfüllt sind. Dazu sollen die Plattformen Freescale Coldfire,<sup>72</sup> Alpha, ARM und S/390 zählen, die sich noch im Teststadium befinden. Darüber hinaus ist eine Unterstützung für IA-64 und MIPS geplant oder in der Entwicklung.

QEMU emuliert eine PowerMac Hardware, die sich aus den folgenden Peripheriegeräten zusammensetzt: UniNorth PCI Bridge, PCI-VGA-kompatible Grafikkarte mit VESA Bochs Extensions, zwei PMAC-IDE-Interfaces mit Festplatten- und CD-ROM-Unterstützung, den eine NE2000 PCI-Netzwerkkarte und entsprechende Eingabegeräte.<sup>73</sup> Mit dieser Ausstattung lassen sich eine Reihe älterer Mac-OS installieren.

<sup>67</sup>Siehe Ankündigung des Entwicklers - <http://article.gmane.org/gmane.comp.emulators.qemu/14666>. Die GL-Aufrufe eines Linux Gastes werden durch eine spezielle Ersatzbibliothek an das Host-System durchgeleitet.

<sup>68</sup>Die Aktivierung erfolgt durch `qemu -M isapc`.

<sup>69</sup>Ältere QEMU-Versionen hatten das Problem, dass Diskettenlaufwerke über ihre BIOS-Schnittstellen aber nicht durch Betriebssystemtreiber angesprochen werden können. Sie funktionierten unter DOS und Windows 3.X jedoch nicht mehr für Windows 95 und aufwärts.

<sup>70</sup>SparcStation 5 (SUN4m): `qemu-system-sparc` und `qemu-system-sparc64` für SUN4u.

<sup>71</sup>Open Hack'Ware, Homepage - [http://perso.magic.fr/l\\_indien/OpenHackWare](http://perso.magic.fr/l_indien/OpenHackWare).

<sup>72</sup>Dieser arbeitet mit einer Untermenge des 68k-Befehlssatzes.

<sup>73</sup>ADB-Tastatur und -Maus.

### 5.6.2 Bochs

Neben dem eingestellten Virtualisierer Plex86 zählt Bochs<sup>74</sup> zu den ältesten Open Source Projekten in diesem Feld. Die Entwicklung begann nach der Jahrtausendwende, die zum Zeitpunkt dieser Arbeit aktuellste Version wurde im September 2007 veröffentlicht. Der Emulator bildet die 32 bit X86-Architektur nach. Derzeit ist Bochs geeignet, 386, 486, die diversen Pentium-Varianten<sup>75</sup> oder X86-64 CPUs inklusive der verschiedenen Erweiterungen der Instruktionensets, wie MMX, SSEx und 3DNow! zu emulieren. Mit der neuesten Version steht zudem ACPI-Unterstützung zur Verfügung. Der Emulator verwendet ein eigenes Containerformat, welches von QEMU und "qemu-img" verstanden wird.

**Generelle Metadaten** Der Emulator wurde in C++ programmiert. Er unterliegt der LPGL<sup>76</sup>. Das Benutzer-Interface von Bochs ist eher sparsam gestaltet. Es stellt damit keine hohen Ansprüche an zusätzliche Bibliotheken.<sup>77</sup> Der Emulator kann auf Linux 2.4 und 2.6 sowie neueren Windows Host-Systemen ausgeführt werden. Für Bochs wurde ein eigenes BIOS geschrieben, welches die typische API der X86-Plattform bereitstellt.

Dieses erlaubt, Nutzungsumgebungen auf Basis von Linux 2.4 und 2.6, diverse DOS-Varianten und Windows 95, 98, Windows NT, 2000, XP und Windows Vista auszuführen.

### 5.6.3 Java PC Emulator - Dioscuri

Java bietet als plattformunabhängige Programmiersprache<sup>78</sup> die Chance, Programme zu erstellen, die in unterschiedlichen Referenzumgebungen ablauffähig sind. Solcherart Applikationen vermeiden zu starke Abhängigkeiten von bestimmten Kombinationen aus Rechnerarchitektur und Betriebssystem. Zudem ist die JVM in viele Web-Browser eingebettet und kann auf diese Weise sogenannte Applets auf der Maschine des Endanwenders ausführen. Ein Beispiel hierfür ist der im nächsten Abschnitt präsentierte JPC. Für solche Konstrukte sind Überlegungen zu treffen, wie ein Datenim- und -export neben der grafischen und Soundausgabe aussehen könnten.<sup>79</sup>

Die Niederländische Nationalbibliothek, das Nationalarchiv sowie die Technologieberatung Tessella entwickelten in Kooperation eine zuerst 16 bittige Version eines X86-Emulators. Dieser, Dioscuri<sup>80</sup> getauft, soll im Zuge des PLANETS-Projekts weiterentwickelt werden, so dass zukünftig 32 bit-Betriebssysteme bis hin zu aktuellen Windows und Linux Versionen ablauffähig sein sollen.

<sup>74</sup>Ausgesprochen "Box", siehe Homepage - <http://bochs.sourceforge.net>.

<sup>75</sup>586, 686, Pentium, PentiumII/III, Pentium4.

<sup>76</sup>Lesser GPL - eine abgewandelte GPL beispielsweise für dynamisch gelinkte Bibliotheken, siehe hierzu - <http://www.gnu.de> (deutsche Übersetzung).

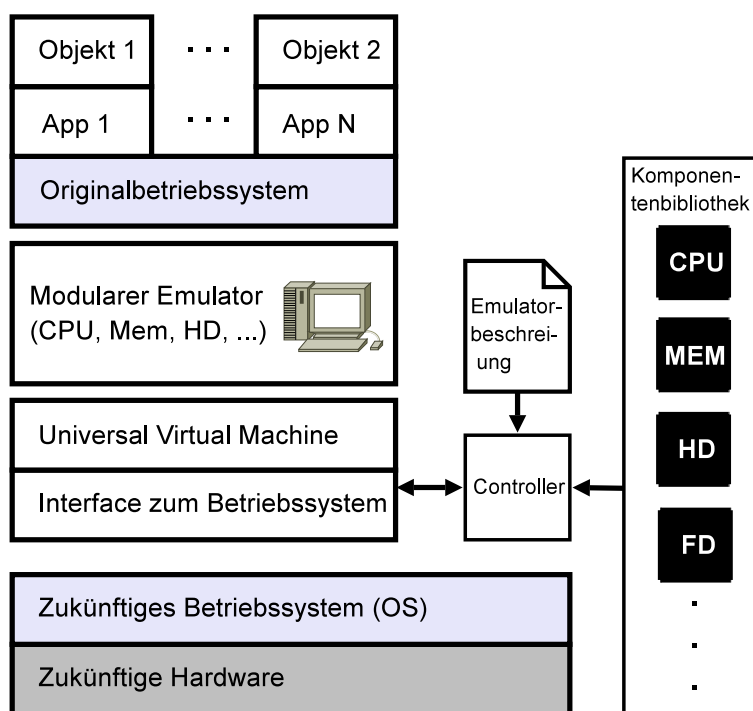
<sup>77</sup>Lediglich die SDL ist neben den typischerweise bereits installierten Bibliotheken erforderlich.

<sup>78</sup>Siehe hierzu Abb. 3.9, S. 69: Für jede unterstützte Rechnerplattform gibt es eine sogenannte Java Virtual Machine (JVM), die eine architekturunabhängige Abstraktionsschicht mit der jeweiligen Plattform verbindet.

<sup>79</sup>Dieser Typus von Problemstellung ist lösbar, wie in [Welte 2008] gezeigt wird.

<sup>80</sup>Eine knappe Beschreibung liefert [van der Hoeven 2007]. Der Name des Projekts bezieht sich auf die griechische Sage der beiden Zwillinge Castor und Pollux, von denen einer sterblich und der andere unsterblich ist, siehe [von Ranke-Graves 1984]. Mit diesem Gleichnis soll die Idee von Emulation ausgedrückt werden: "Endlichen" digitalen Objekten sollen ihre unsterbliche Equivalente zur Seite gestellt werden.

Der Emulator wurde als modulares Werkzeug<sup>81</sup> angelegt: Jede Hardwarekomponente existiert als eigenständiges Modul und aus der Kombination von Komponenten, unter Berücksichtigung eines gewissen Basis-Sets, wird ein vollständiger Hardwareemulator generiert (Abb. 5.5).



**Abbildung 5.5:** Modularer Aufbau des in Java programmierten X86-Emulators Dioscuri (nach [van der Hoeven u. a. 2005]).

**Generelle Metadaten** Aus Sicht des Autors bietet die Open Source Lizenz GPL <sup>82</sup> Version 2 die besten Chancen für die Weiterentwicklung und langfristige Perspektive eines Java-Emulators für X86-Maschinen. Es besteht auf diese Weise die Möglichkeit einen Standard zu schaffen und eine breite Anwendergemeinde für den Einsatz zu gewinnen. Kurzfristige ökonomische Erfolge lassen sich vermutlich nicht erzielen. Sie sind zudem nicht die Kernkompetenz der initiierenden Gedächtnisorganisationen. Der Emulator wurde speziell unter den Gesichtspunkten der digitalen Langzeitarchivierung konzeptionell angelegt.

**Anforderungen an das Host-System** Dadurch, dass der Emulator in Java geschrieben wurde, kann er theoretisch auf der sehr breiten Palette von Rechnerplattformen zum Einsatz kommen, die über eine passende JVM verfügen. Die Voraussetzung besteht in einem Java Runtime Environment (JRE) der Version 1.5.x. Dioscuri wurde erfolgreich auf folgenden Plattformen getestet:

<sup>81</sup>Vgl. hierzu die Ausführungen in Abschnitt 4.8.4 zu modularen Emulatoren und insbesondere die Artikel [van der Hoeven u. a. 2005], [Verdegem und van der Hoeven 2006] und [van der Hoeven u. a. 2007].

<sup>82</sup>Vgl. hierzu <http://www.gnu.de> und Abschnitt 7.11.

- Windows XP mit JRE v1.5.x
- Linux Fedora Core 4, Ubuntu 7.04, OpenSuSE 10.2, 10.3 Linux jeweils mit JRE v1.5.x
- SUN Sparc Solaris mit JRE v1.5.x

Daraus lässt sich eine generelle Einsatzfähigkeit für die meisten derzeitigen Standardplattformen ableiten.

**Interaktion, Datenimport- und -export** Der Emulator läuft in einem Desktop-Fenster des jeweiligen Referenzsystems. Das Benutzer-Interface verfügt über ein Menü, mittels dessen sowohl die Steuerung erfolgt, als sich auch Parameter beeinflussen lassen. Das Format der virtuellen Festplatte ist einfach gehalten und kann daher bei nicht laufendem Emulator mit Daten bestückt werden. Ebenso findet der Anwender virtuelle Diskettenlaufwerke zum Datenaustausch vor.

**Unterstützte Nutzungsumgebungen** Nach dem bisherigen Stand der Entwicklungen bildet der Emulator eine 16 bit Intel 8086 basierte CPU nach, die DMA-Unterstützung bietet und Interrupt-Behandlung beherrscht. Das BIOS stammt wie bei anderen Open-Source-Emulatoren auch von Plex86/Bochs. Derzeit stehen Nutzungsumgebungen 1 MByte RAM zur Verfügung, die mit späteren Versionen erweitert werden sollen. Als Festspeicher sind Diskette und Festplatte implementiert, die über die üblichen BIOS-Funktionen angesprochen werden können. Derzeit sind nur Tastatureingaben möglich, eine Mausunterstützung besteht noch nicht. Ausgaben können auf eine virtuelle VGA-Karte mit Plex86/Bochs VGA BIOS erfolgen.

Diese Ausstattung begrenzt den Einsatz von Betriebssystemen derzeit noch auf ältere DOS-Versionen und ein spezielles 16 bit Linux. Mit dem Ausbau auf höhere CPU Versionen und größeren Speicher werden weitere X86-Betriebssysteme ausführbar.

**Nutzbarkeit, Zuverlässigkeit, Portabilität** Die Programmiersprache Java, sonst nicht üblich für die verbreiteten kommerziellen und Open-Source-Virtualisierer, wurde in Hinblick auf aktuelle Portabilität zwischen verschiedenen Rechnerplattformen und eine notwendige Portierung auf zukünftige Architekturen gewählt. Da sich das Werkzeug noch mitten in der Entwicklung befindet, sind Aussagen zur Nutzbarkeit und Zuverlässigkeit nur bedingt zu treffen.

#### 5.6.4 University of Oxford Java PC Emulator

JPC ist ein X86-Emulator, der für die Trennung von verschiedenen Umgebungen auf einer Maschine entwickelt wurde. Der Fokus lag daher auf Performance und nicht auf einer langfristigen Verfügbarkeit von alten Nutzungsumgebungen. Der Emulator kann wegen seiner Programmierung in Java auf verschiedenen Referenzumgebungen ausgeführt werden. Die Autoren des Programms sehen durchaus eine Anwendung auf mobilen Endgeräten und anderen Rechnerarchitekturen als Einsatzoption. JPC steht zudem als Java Applet zur Verfügung

und kann damit ohne weitere Installation (eine JVM vorausgesetzt) von überall her über ein Datennetz aufgerufen werden.<sup>83</sup>

Die am Projekt beteiligten Entwickler geben die Ablaufgeschwindigkeit des JPC mit bis zu 10,% der realen Maschine an [JPCP 2007]: *"This puts it amongst the fastest emulators of an x86 PC, despite needing only a pure Java environment. At first sight it might seem impossible for a Java based emulator to approach other emulators which use "native" code. However JPC uses a number of optimisation strategies to achieve an acceptable speed. In fact these strategies are very similar to those used by modern X86 hardware; e.g. using a RISC microcode set to streamline the X86 instructions. JPC then uses dynamic binary translation to get the best performance."*

**Generelle Metadaten** Das Projekt startete im August 2005 an der Oxford University im Bereich der Teilchenphysik.<sup>84</sup> Das Programm steht seit Mitte 2007 unter der GPL.

**Anforderungen an das Host-System** Interessanter Aspekt dieses Programms ist, dass es neben einer klassischen Ausgabe im Browser eines Anwenders ausgeführt werden kann. Damit sind bis auf das Vorhandensein einer passenden Java Virtual Machine der Version 1.5 keine wesentlichen Voraussetzungen an das Host-System des Benutzers gestellt. Das bedeutet, dass jede beliebige Rechnerarchitektur und jedes beliebige Betriebssystem, welche über eine entsprechende JVM verfügen, sich als Host-Umgebung für den Emulator anbieten.

**Interaktion, Datenim- und Export** Die Benutzerinteraktion mit dem JPC läuft über das Browser-Fenster. Damit bestehen einige Einschränkungen bezüglich der transportierten Tastaturcodes. In einer früheren Version von Anfang 2007 war es deshalb mit einer deutschen Tastaturbelegung nicht möglich, einen Doppelpunkt einzugeben oder spezielle Steuerungssequenzen auszulösen. Dieses wurde in der aktuellen Version dadurch gelöst, dass im Applet ein zusätzliches Tastaturfeld eingeblendet wird, welches auf Mausklick definierte Sequenzen auslöst.

**Unterstützte Nutzungsumgebungen** Die virtuelle Hardware emuliert eine Intel-kompatible CPU im Real-Mode; der Protected-Mode wird mit "Beta" angegeben. Als BIOS-Ersatz für System und Grafikkarte kommt die gewohnte Plex86/Bochs Kombination zum Einsatz. Die Peripherie ist ähnlich wie beim Dioscuri komplett emuliert. Derzeit sind lediglich Tastatureingaben vorgesehen; eine Mausunterstützung fehlte zum Zeitpunkt der Recherchen.

**Nutzbarkeit, Zuverlässigkeit, Portabilität** Dieser Emulator demonstriert einen interessanten Aspekt des Java-Einsatzes: Die Applikation kann, eine geeignete JVM vorausgesetzt, auf der Maschine des Anwenders als Java-Applet ausgeführt werden, ohne dass dafür eine direkte Installation der Software notwendig wäre.

<sup>83</sup>Die bereits bei den Java-basierten Home-Computer-Emulatoren angesprochenen Probleme des Objekttransports (Abschnitt 5.1.3) bleiben bestehen.

<sup>84</sup>Projekt-Homepage, siehe [JPCP 2007].

Bei der Anwendung dieses Emulators im Web-Browser – man ist derzeit auf die Software festgelegt, die seitens der Anbieter des Applets installiert wurde – waren einige interessante Probleme zu beobachten. Die Schachtelung der Anwendungen ineinander (vgl. Abb. 4.7, S. 106) kann einige Probleme bereiten: Es besteht ein wesentlicher Unterschied, ob Tastatureingaben direkt interpretiert, oder durch eine vorgegebene Umgebung aufbereitet, abgefangen oder verändert werden. Ein Java-Applet läuft in einer JVM in einem Browser als ein Benutzerprozess in einem Fenster der jeweiligen grafischen Benutzeroberfläche. Damit kommt nur der Teil der Tastatureingaben in dieser Umgebung an, die nicht vorher durch das Betriebssystem oder den Window-Manager abgefangen werden. So wird beispielsweise die Kombination aus Control-Alt-Delete, die einen Warmstart eines X86-PC auslösen kann, vom Host-Betriebssystem abgefangen. Ähnliches gilt für diverse Funktionstasten und Tatenkombinationen mit Alt oder AltGr, die im Window-Manager für das Öffnen von Menüs und die Auswahl von Einträgen genutzt werden. Ähnliches gilt für unterschiedliche Lokalisierungen: Sitzt der Anwender vor einer deutschen Tastatur mit deutschsprachig angepasster Benutzeroberfläche, so ist die Frage, wie eine englische Tastaturbelegung in der Nutzungsumgebung übersetzt werden soll.<sup>85</sup>

### 5.6.5 DOSBOX

Ein recht spezieller Emulator ist die DOSBOX,<sup>86</sup> die für das abgeschlossene Gebiet der DOS-Spiele geschrieben wurde. Sie implementiert die wichtigsten DOS-Funktionen direkt in einer Art kombinierten Firmware zusammen mit den notwendigen BIOS-Funktionsaufrufen. Das Programm steht unter der GPL.

DOSBOX für sich ist ein kommandozeilenorientiertes Programm. Mit DBOXFE steht jedoch zudem eine grafische Benutzerschnittstelle zur Verfügung, die bestimmte Einstellungen einfacher vornehmen lässt. Es erlaubt zudem, bestimmte Spiele direkt starten zu können.

**Nutzbarkeit, Zuverlässigkeit, Portabilität** Ein wesentlicher Vorteil der DOSBOX liegt in der Bereitstellung aller wichtigen Schnittstellen zum Host-System ohne dafür notwendige Installation eines DOS und Bereitstellung der eventuell notwendigen Lizenzen. Für die Ein- und Ausgabe nutzt das Programm die SDL-Bibliothek. Zudem werden alle Treiber für die unterstützte Hardware zur Verfügung gestellt. Durch die direkte Nutzung des Dateisystems der Referenzplattform wird die Installation der darzustellenden Objekte und das generelle Management der Präsentationsplattform wesentlich erleichtert.

Ein signifikanter Nachteil liegt in der Beschränkung auf bestimmte digitale Objekte und die Referenzumgebung. Damit ist der Ansatz eher für technische Museen oder auf bestimmte Objekte zentrierte Ausstellungen denn für Archive oder Bibliotheken von Interesse.

**Anforderungen an das Host-System** DOSBOX wurde für Linux 2.4, 2.6 und neuere Windows Betriebssysteme geschrieben, um auf diesen Host-Plattformen ältere DOS-Spiele

---

<sup>85</sup>Vgl. hierzu Abschnitt 6.5.4.

<sup>86</sup>Projekt-Homepage - <http://www.dosbox.com>. Zum Zeitpunkt des Schreibens war die Version 0.72 von Ende August 2007 aktuell. In Experimenten wurde auch die Vorgängerversion 0.65 genutzt (Abb. 4.4, S. 95 und Abb. A.4, S. 235).

direkt ausführen zu können.

Die Eingaben an die DOSBOX via Tastatur oder Maus werden direkt aus dem Host-System übersetzt, die Lokalisierung kann festgelegt werden. Die Ausgabe erfolgt in ein Fenster der grafischen Benutzeroberfläche des Host-Systems.

**Unterstützte Nutzungsumgebungen** Der Emulator arbeitet mit einem eingebauten BIOS. Das Tastaturlayout kann festgelegt werden. Anders als die typischen X86-Virtualisierer und Emulatoren greift DOSBOX direkt auf Verzeichnis seines Host-System zurück und bindet dieses an DOS-Laufwerksbuchstaben. So lassen sich Installationen von Primärobjekten ohne laufenden Emulator vornehmen. Netzwerkfunktionalität wie IPX oder Modem können emuliert werden. DOSBOX unterstützt verschiedene Maschinentypen; für die Grafikausgabe im Desktop-Fenster stehen die X86-Standards Hercules, CGA, EGA, VGA und VESA zur Auswahl. Dabei ist eine Vollbildausgabe möglich. Die Soundausgabe kann mittels virtuellem PC-Lautsprecher oder verschiedener ehemals populärer Audioadapter, wie Midi/Adlib, Soundblaster Pro oder Gravis Ultra erfolgen.

## 5.7 Emulatoren und -Virtualisierer im Vergleich

Die Zahl der kommerziellen und Open-Source-Virtualisierer und Emulatoren ist inzwischen recht umfangreich. Die wenigsten der betrachteten Programme sind primär für die Langzeitarchivierung entwickelt worden, jedoch lassen sich zum jetzigen Zeitpunkt folgende Aussagen treffen:

- Die kommerziellen Produkte haben typischerweise einen höheren Reifegrad erreicht als ihre Open-Source-Pendants. Vielfach zeichnen sich erstgenannte durch bessere Benutzer-Interfaces und einfachere Bedienung aus. Dieses relativiert sich jedoch im Zeitablauf oder bei Überlegungen zur Integration in automatisierbare Umgebungen.
- Open-Source-Emulatoren und -Virtualisierer sind oft deutlich flexibler als Closed-Source-Produkte: Sie können von Haus aus mit verschiedenen Containerformaten umgehen, stehen für mehr Plattformen zur Verfügung und bieten oft eine breitere Palette virtualisierter Hardware. Gerade die Verfügbarkeit sehr alter virtueller ISA-Hardware oder sukzessive veraltender PCI-Hardware nimmt an Bedeutung zu.
- QEMU, Bochs und Dioscuri zeigen, dass der Open-Source-Ansatz im Bereich X86-Emulation funktionieren kann und sogar deutlich bessere Chancen offeriert, eine wirkliche Langzeitperspektive zu eröffnen. Teilweise existieren bereits große Nutzergemeinschaften, die viele, in der Umsetzung oft teure Aufgaben, wie Spezialanpassungen oder umfangreiche Tests und Bugfixes aus eigenem Interesse durchführen. Darüber hinaus lassen sich solche Projekte durch intensives Engagement beeinflussen und um die eigenen Komponenten erweitern. Hier könnte diskutiert werden, eine geeignete Servicelandschaft rund um die bestehenden Projekte aufzubauen und mit kommerziellen Partnern zu realisieren.<sup>87</sup>

<sup>87</sup>Dieses wäre ein ökonomisch bereits erprobtes Modell, wie die Angebote großer IT-Unternehmen, wie IBM oder HP, rund um das Betriebssystem Linux zeigen.



- Kommerzielle Anbieter, wie Microsoft, haben zwar mit ihren Virtualisierern eine längere Verfügbarkeit ihrer älteren Systeme im Sinn. Jedoch greift der Ansatz oft viel zu kurz und beschränkt sich auf zu wenige Nutzungs- und Referenzumgebungen, als dass sich darauf eine erfolversprechende Langzeitarchivierungsstrategie aufbauen ließe.
- Offene Projekte mit vielen Beteiligten oder Partnern haben bereits eine beeindruckende Zahl inzwischen historischer Plattformen emuliert. Die Liste der allein von QEMU und MESS/MAME abgedeckten Systeme ist beachtlich. Anders als bei VMware oder Virtual PC fallen ältere Nutzungsumgebungen im Zeitablauf nicht heraus.
- Mit MESS/MAME liegen interessante Ansätze einer modularen Emulation vor. Schon vorhandene CPU- und Peripheriemodule erleichtern die Integration weiterer ähnlicher Architekturen.
- Open-Source-Emulatoren erlauben es, Softwareabhängigkeiten (Abb. 4.5) zu reduzieren: So lassen sich alle benötigten Bibliotheken statisch einkompilieren, so dass lediglich eine Binärkompatibilität zum laufenden Betriebssystem-Kernel erforderlich ist. Dieses Problem versuchen kommerzielle Virtualisierer durch das Mitliefern eines möglichst vollständigen Satzes erforderlicher Bibliotheken zu umgehen.

Eine ganz neue Richtung erhält die Emulation durch die Entwicklung eines dedizierten Werkzeugs für die Langzeitarchivierung: Dioscuri hat mit einem bisherigen geschätzten Aufwand von drei Mannjahren die Reife eines 286/386 PCs erreicht. Noch fehlt einige Peripherie wie Audioadapter und Netzwerkkarte. Die bisherigen Fortschritte zeigen jedoch, dass der Emulationsansatz mit sowohl beherrschbarem als auch ökonomisch kalkulierbaren Aufwand verfolgt werden kann.

**Unvollständigkeit der Emulation** In der direkten Ausführung der meisten CPU-Befehle auf dem Host-Prozessor liegt die größte Einschränkung der Virtualisierer für die Langzeitarchivierung: Solange sich die darunterliegende Hardware nicht so grundlegend ändert, dass zumindest theoretisch jede X86-Software auf ihr ablaufen könnte, genügen Virtualisierer wie VMware, VirtualBox, Virtual PC und Parallels, um sich vor der Obsoleszenz von Betriebssystemen zu schützen.

So gilt, beispielsweise für die End- und Heimanwenderbetriebssysteme Windows 95, 98 und ME, dass diese nicht mehr von Microsoft gepflegt werden und keine Updates und Patches mehr verfügbar sind. Wenn für diese jedoch Programme geschrieben wurden, die nicht korrekt auf Windows 2000, XP, Vista laufen, bietet Virtualisierung einen Ausweg: Der Betrieb der eingangs genannten Betriebssysteme auf heutiger Hardware mag schon wegen fehlender Hardwaretreiber nicht sinnvoll möglich sein; hinzu kommen die schwerwiegenden Sicherheitsprobleme dieser alten Betriebssysteme. Die Verwendung in einem Virtualisierer erlaubt zweierlei:

1. Vor jedem Start einer Nutzungsumgebung kann ihre virtuelle Festplatte, die Containerdatei im Host-System kopiert und nur mit dem Duplikat gearbeitet werden. So sind Tests und Installationsversuche möglich, ohne die Ausgangsinstallation zu irreversibel zu verändern.

2. Einige Virtualisierer bieten zudem die Verwendung sogenannter "Undo"-Schritte an, die es erlauben, vorgenommene Änderungen an der Containerdatei in Stufen wieder rückgängig zu machen. Das vereinfacht den Arbeitsablauf gegenüber dem ersten Fall.

Aufgrund dieser Rahmendaten erscheinen Virtualisierer zumindest als Archivierungslösung für einen Teil der identifizierten Nutzergruppen<sup>88</sup> für einen gewissen Zeitrahmen. Kleinere Institutionen wären mithilfe von Virtualisierung in der Lage ihren gesetzlichen Vorgaben zur Aufbewahrung ihrer Dokumente<sup>89</sup> durch Zentralisierung auf einer Referenz-Workstation zu genügen.

### 5.7.1 Praktische Implikationen

Installationen in X86-Emulatoren oder Virtualisierern unterscheiden sich nicht von denen auf einer realen Hardware. So wie man beim Wechsel der physischen Hardware das installierte System an das neue anpassen muss, gilt dieses ebenso beim Update oder Wechsel des Emulators oder Virtualisierers. Die Migration kann einerseits horizontal von einer älteren auf eine neuere Version eines Virtualisierers erfolgen oder vertikal<sup>90</sup> durch Wechsel von einem Werkzeug auf ein gleichwertiges anderes. Das sukzessive Update eines Programms ist zur Mitführung mit einer Referenzumgebung unumgänglich, da ältere Versionen nach einer gewissen Zahl von Updates des darunterliegenden Betriebssystems nicht mehr lauffähig sind. Dieses gilt beispielsweise für ältere Versionen der VMware Workstation vor Version 5.5.

Ein weniger offensichtliches Problem besteht in der Weiterentwicklung der Nutzungsumgebungen: Da aktuelle Nutzungsumgebungen auf der verfügbaren Hardware direkt ausgeführt werden können, besteht erst einmal kein großer Bedarf, diese im Emulator oder Virtualisierer auszuführen. So haben beispielsweise ältere Programme Probleme das aktuelle Windows Vista<sup>91</sup> ablaufen zu lassen.

Für X86-Emulatoren und Virtualisierer existiert kein allgemeines werkzeugübergreifendes, standardisiertes Containerformat der virtuellen Festplatten. Zudem gibt es im Laufe der Versionswechsel einzelner Programme den Bedarf nach Aktualisierungen des Formats. Deshalb kommt man um eine Transformation des Containers nicht umhin (siehe Abbildung 5.6).

**Updates der VMware Workstation Version 3.2 bis 6.0** Die VMware Workstation erscheint wegen ihres für ein Softwareprodukt dieses Anwendungsgebiets hohen Alters von knapp zehn Jahren als interessantes Untersuchungsobjekt für Aspekte der Langzeitarchivierung:

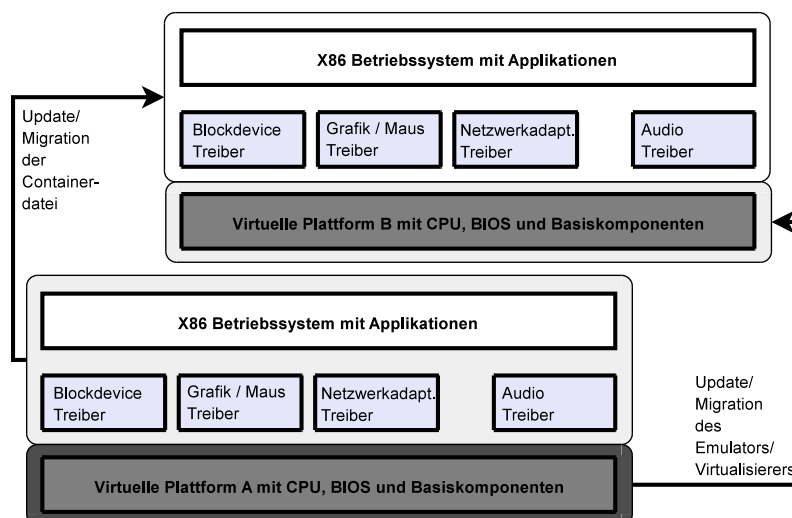
- Beobachtung der Produktpolitik, -pflege und des Funktionsumfanges
- Notwendige Migrationsschritte beim Versionswechsel

<sup>88</sup>Siehe hierzu Abschnitt 4.5.

<sup>89</sup>Texte, Tabellenkalkulationen, Datenbanken unterliegen typischerweise Aufbewahrungsfristen von fünf bis 20 Jahren.

<sup>90</sup>Vgl. hierzu Abschnitt 3.4 und Abbildung 3.6, S. 60.

<sup>91</sup>So bereitete es der VMware Workstation 4.0 Schwierigkeiten, mit dem 2.6er Linux Kernel umzugehen. Solche Probleme werden üblicherweise mit neueren Versionen behoben.



**Abbildung 5.6:** Üblicherweise müssen bei einem Wechsel des Emulators oder Virtualisierers das Containerformat transformiert und die Hardwaretreiber des Gastsystems angepasst werden.

Insgesamt ließen sich drei Problemfelder identifizieren: Im Laufe der Updates wurde das Containerformat gewechselt. Schrittweise Aktualisierungen stellten keine Schwierigkeit dar, jedoch waren Container der Version 3.2 nicht mehr mit der Workstation 6.0 zu öffnen.<sup>92</sup> Weitere Herausforderungen ergaben sich aus den "inneren" Updates. Die Aktualisierung der virtuellen Hardware machte in vielen Fällen eine Reinitialisierung der Nutzungsumgebungen erforderlich. Hier konnte es vorkommen, dass sehr alte Umgebungen "abgehängt" wurden. Das liegt daran, dass die Hardware und Treiber eines bestimmten Betriebssystems nur für bestimmte Perioden gemeinsam gültig sind. Zuguterletzt traten mit neuen Versionen der Referenzumgebungen - Windows Vista oder Linux Kernel 2.6 - Installationsprobleme für ältere Workstations auf.<sup>93</sup>

**Flexibilität und Abhängigkeiten** Mit der zunehmenden Anzahl obsoleter X86-Betriebssysteme kann man feststellen, dass eine virtuelle Peripherie pro Klasse - Audio, Netzwerk, Grafik, Board - langfristig nicht ausreicht. Hierin liegt ein zwangsläufiges Problem des technologischen Wandels. Trotzdem sollte man versuchen eine möglichst kleine gemeinsame Basis für einen definierte Zeiträume anzustreben, um den Pflegeaufwand klein zu halten.<sup>94</sup> QEMU demonstriert die Machbarkeit dieses Ansatzes mit einer größeren Auswahl an Sound-, Netzwerk- und Grafikkarten.

Dieses Open-Source-Werkzeug zeigt den Vorteil offener Ansätze: So verfügt QEMU über ein Monitorinterface, kann verschiedene, auch nicht-eigene Containerformate lesen. Mit der Unterstützung des VMware SVGA II Grafikadapters unterscheidet sich die virtuelle Hardware der VMware Workstation und Player kaum noch von der des aktuellen QEMU.

<sup>92</sup>Dieses gelang erst mit einer Transformation durch "qemu-img", vgl. dazu Abschnitt 5.6.1. Jedoch hat "qemu-img" generell Probleme aufgespaltene VMware-Images zu behandeln (beim Anlegen Option "split into 2 GByte Files" gesetzt).

<sup>93</sup>So sind die Workstations vor Version 6.0 nicht mehr auf Vista ablauffähig. Zudem mußten für Linux spezielle Treiberpakete eingesetzt werden, damit der Workstation-Monitor und die virtuellen Netzwerk-Interfaces übersetzt werden konnten.

<sup>94</sup>Vgl. hierzu die folgenden Kapitel zum Softwarearchiv und dessen Management.

**Zukünftige Entwicklungen** Die Langzeitarchivierung digitaler Objekte kann durch den zunehmenden Trend zur Virtualisierung weitere Unterstützung erfahren. Ältere Ablaufumgebungen bleiben länger präsent und Computernutzer lernen das Konzept Emulation bereits im täglichen Betrieb kennen. Weiterhin helfen die Bemühungen der Hersteller von Virtualisierungslösungen möglichst bequeme Lösungen für den Austausch von Daten<sup>95</sup> zu finden.

Bisher verhielten sich Betriebssysteme weitgehend agnostisch gegenüber Virtualisierung, bis auf die vorgenannten Ausnahmen des Mac-OS-Übergangs und der DOS-Umgebung im Windows NT. Es wären aber generellere Ansätze in der Betriebssystementwicklung vorstellbar: Emulation oder zumindest Virtualisierung abgekündigter Versionen, die durch eigens definierte Schnittstellen in die aktuellen Betriebssysteme und ihre Benutzer-Interfaces leichter eingebunden werden können. Dies setzte eine spezielle Treiberentwicklung und einige Anpassungen – eine Investition, die am Ende eines Produktzyklus anfallen kann – voraus, die sich jedoch unabhängig vom Langzeitarchivierungsgedanken auszahlen könnten: Es erlaubt Betriebssystemherstellern und -programmierern eine größere Unabhängigkeit von Vorgängerversionen. So ließen sich Innovationen oder Sicherheitskonzepte schneller umsetzen. Solange gleichzeitig der Abstand im Bedienverständnis und Art der Benutzerinteraktion nicht zu groß wird, kann der Nutzer ältere Betriebssystemversionen des gleichen Herstellers ebenso wie Varianten anderer Hersteller als Applikation seiner aktuellen Arbeitsumgebung ansehen. Auf diese Weise ließe sich für durchschnittliche Endanwender, wie kleinere und mittlere Unternehmen sowie Privatpersonen ein gewisser Zeitraum überbrücken, der viele Langzeitarchivierungsaufgaben bereits abdecken kann.

Zentral für das Vertrauen in derartige Entwicklungen ist die Unabhängigkeit von einer ganz bestimmten Lösung oder einem Hersteller. Einige Emulatoren oder Virtualisierer verfügen bereits über die Flexibilität mit fremden Containerformaten umzugehen. Für andere existieren Zusatzprogramme, wie der VMware Converter, "qemu-img" oder Angebote von Drittanbietern, wie der VHD-Converter.<sup>96</sup>

---

<sup>95</sup>Vgl. hierzu den späteren Abschnitt 6.8.

<sup>96</sup>Siehe <http://vmtoolkit.com/files/folders/converters/entry8.aspx>.

## Teil III

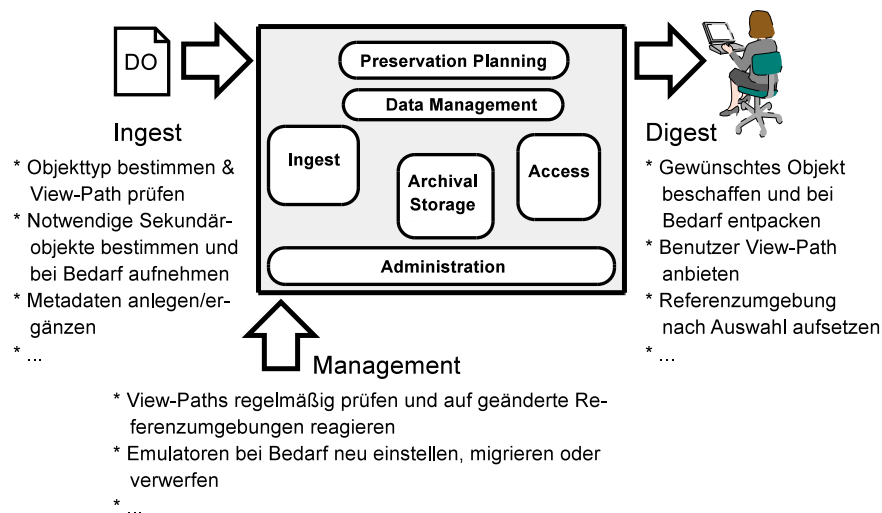
# View-Path, Referenzumgebung und Softwarearchiv



## 6 View-Paths und Referenzumgebungen

Digitale Objekte können nicht aus sich alleine heraus genutzt werden, sondern bedürfen eines geeigneten Kontextes, damit auf sie zugegriffen werden kann. Dieser Kontext, Ablauf- oder Nutzungsumgebung genannt, muss geeignete Hardware- und Softwarekomponenten so kombinieren, dass je nach Typ des Primärobjekts seine Erstellungsumgebung oder ein geeignetes Äquivalent erzeugt wird. Deshalb sind View-Paths, die Darstellungspfade, eines der zentralen Konzepte, die in dieser Arbeit diskutiert werden.

Einem digitalen Langzeitarchiv erwachsen aus diesem Konzept verschiedene Aufgaben, die von der Bestimmung des Objekttyps beim Einstellen in das Archiv, die Generierung und Ablage der benötigten Metadaten, bis zur Überprüfung auf Vorhandensein oder die Herstellbarkeit einer Ablaufumgebung reichen. Für das OAIS-Management leiten sich daraus einige



**Abbildung 6.1:** Im Zusammenhang mit einem digitalen Langzeitarchiv treten bestimmte Aufgaben- und Fragestellungen auf.

Aufgaben ab, die im Laufe dieses Kapitels näher erläutert werden. Eine Rolle spielt beispielsweise die Klassifikation der Objekte, neben der Erhebung und Erzeugung ihrer Metadaten, die anschließend behandelt werden. Dieses beeinflusst nicht unwesentlich die später notwendigen Schritte für einen Zugriff. Ebenso können je nach Benutzergruppe oder Einrichtung

verschiedene Kriterien existieren, nach denen View-Paths bestimmt werden. Es lassen sich drei wesentliche Phasen voneinander unterscheiden (Abb. 6.1):

- Notwendige Arbeitsschritte und Handlungen bei der Objektaufnahme ins Archiv (Ingest als SIP)
- Regelmäßige Schritte und Arbeitsabläufe im Laufe des Archivbetriebs
- Handlungsanweisungen für den Objektzugriff nach der Objektausgabe an den Endbenutzer (Digest als DIP)

Zur Ermittlung des Objekttyps existieren verschiedene Varianten möglicher Arbeitsabläufe sowie Programme und Netzwerkdienste.

Die Auswahl über die in das Softwarearchiv einzustellenden und später zu pflegenden Emulatoren entscheidet wesentlich über die Nachnutzbarkeit der Primärobjekte des Langzeitarchivs und ist damit eine zentrale Erfolgsbedingung für verschiedene Archivierungsstrategien. Deshalb ist bei der Einstellung eines digitalen Objekts in das Archiv zu ermitteln, ob für den gegebenen Objekttyp ein View-Path bestimmt werden kann und auf welche später nachzubildende Rechnerplattform dieser zeigt. Umgekehrt kann das beim Ingest bedeuten, dass ein Objekt eventuell zurückgewiesen oder nur unter dem Vorbehalt einer später verfügbaren Lösung eingestellt wird.

Die Schritte zur Sichtbarmachung von verschiedenen, im Langzeitarchiv enthaltenen Objekttypen sind im Moment der Archiviausgabe festzulegen. An dieser Stelle werden diverse Arbeitsabläufe notwendig, die es erst erlauben, dass ein späterer Archivnutzer tatsächlich mit dem gewünschten Objekt umgehen kann. In diesem Moment können Migrations- und Emulationsstrategien ähnliche Auswirkungen haben: 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. In jedem Fall ist es unumgänglich, dem Archivnutzer eine Reihe von Hilfsmitteln für den Zugriff zu präsentieren. Um dabei den Aufwand für häufig auftretende View-Paths zu verringern, lassen sich Überlegungen zur Aggregation oder Caching vornehmen.

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.

Neben den direkt mit den Objekten verknüpften Aufgaben sind zusätzliche Überlegungen anzustellen, die Behandlung und Transport der Primärobjekte in den verschiedenen Phasen organisieren. So ist ein Datenaustausch zwischen der im Emulator ablaufenden Software und der Software auf dem Referenzsystem zu realisieren. Diese Aufgabenstellung 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 stehengebliebenen emulierten Systems und der Referenzplattform. Zum Teil halten die verfügbaren Emulatoren bereits Werkzeuge oder Konzepte vor, um die Brücke zu schlagen. Zum Ende des Kapitels wird deshalb der Datentransport mittels virtueller Blockgeräte oder geeigneter Netzwerkprotokolle näher ausgeführt.



## 6.1 Anforderungen an die Objektaufnahme

Wenn ein Objekt in ein digitales Langzeitarchiv übertragen wird, erfolgt seine Klassifizierung. Hierzu kann es notwendig sein, dass der Objekttyp umfassend bestimmt wird und fehlende Informationen hinzugefügt werden. Falls das Objekt aus mehreren Teilobjekten besteht, muss diese Bestimmung für jedes Teilobjekt ausgeführt werden. Auf jeden Fall ist der Dateityp zu ermitteln, damit eine geeignete Einordnung in eine oder mehrere Archivierungsstrategien erfolgen kann. Folgende Überlegungen könnten je nach Art des Archivs eine Rolle spielen:

- Typ des Objekts. Je nach Dateityp eignet sich das Objekt für eine Migrationsstrategie oder es muss die komplette Ablaufumgebung aufbewahrt werden. Unter Umständen stellt sich heraus, dass für den gegebenen Objekttyp noch keine Strategie existiert. Hier ist dann zu überlegen, ob das Objekt abgewiesen wird oder die notwendigen Komponenten zur Unterstützung in das Archiv eingestellt werden.
- Wahrscheinlichkeit des Zugriffs. Je häufiger Zugriffe auf ein bestimmtes Objekt oder ausgewählte Objekttypen sind, desto sinnvoller wird es, eine Kopie der interessierenden Objekte früh in die jeweils aktuelle Referenzumgebung zu migrieren oder eine geeignete Ablaufumgebung permanent vorzuhalten.

Selbst wenn sich ein digitales Objekt gut für eine Migrationsstrategie eignet, sollte zur Überprüfbarkeit der erfolgten Migrationsschritte und zum Beweis der Authentizität des Objekts die Erstellungsumgebung erhalten werden. Solcherart Hinweise sind in den Metadaten zu hinterlegen. Ein Modell für eine solche Verknüpfung der Metadaten inklusive der Informationen für den späteren Zugriff wird in [van Diessen und Steenbakkers 2002], Kapitel 6 umrissen.

Für die Ermittlung des Dateityps und weitergehender Angaben bieten sich verschiedene, bereits vorhandene Format *Registries* oder -Validatoren an:

- PRONOM ist eine öffentlich nutzbare Registratur [TNA 2007] des Britischen Nationalarchivs, welches ursprünglich für interne Zwecke eingerichtet, später aber allgemein öffentlich zugänglich gemacht wurde. Für bekannte Dateitypen liefert PRONOM Metainformationen, wie den Ersteller des Dateiformats, Informationen zur Erstellungsapplikation, Versionsnummer und technische Details über den Aufbau des Formats. Die Registratur verwendet sogenannte PUIDs (PRONOM Persistent Unique Identifier), die ein ähnliches Konzept wie Unix Magic Numbers<sup>1</sup> umsetzen. Bisher steht eine Webschnittstelle zur Nutzung offen, die derzeit um einen entfernt einbindbaren Web-Service erweitert wird. Zusätzlich wird mit "DROID" eine separate Java Applikation zur Nutzung angeboten,<sup>2</sup> die diese Registratur benutzt.
- JHOVE (JSTOR/Harvard Object Validation Environment) ist ein Werkzeug zur Formaterkennung und Validierung. Das Programm wurde in Java erstellt, das die Version 1.4 auf dem jeweiligen System voraussetzt. Es lässt sich als Bibliothek in eigene Projekte einbinden. Die von diesem Programm erzeugten technischen Metadaten können sowohl

---

<sup>1</sup>Vgl. hierzu den Punkt zu Unix *file*.

<sup>2</sup>Siehe neben [TNA 2007] speziell auch die Homepage - <http://droid.sourceforge.net>.

in XML mit definiertem XMLSchema als auch Standard-Textform ausgegeben werden. Das Programm ist in der Version 1.1 als Kommandozeilenwerkzeug oder grafisches Tool erhältlich und steht unter der Lesser GPL. Damit können Drittanbieter verschiedene Frontends dazu anbieten. Derzeit kennt JHOVE die Formate AIFF, ASCII, Bytestream, GIF, HTML, JPEG, JPEG 2000, PDF, TIFF, UTF8, WAV, und XML. Es kann diese Formate validieren und auf diese Weise sicherstellen, dass eine Datei, welche in einem fraglichen Format vorliegt, dessen Spezifikation vollkommen entspricht.

- Unix-*file* ist Bestandteil der meisten Unix-artigen Betriebssysteme und ist frei kopierbar.<sup>3</sup> Dieses Kommando liefert anhand einer "Fingerabdruck"-Datenbank den Typ einer Datei anhand von Magic-Numbers zurück.
- FILEExt<sup>4</sup> ist eine Datenbank, in der anhand von Dateiendungen nach Formatmetadaten gesucht werden kann. Die Ergebnisseite listet beispielsweise Erstellungsapplikationen, MIME-Typs und den Identifier auf.
- C.E. Codere's File Format Site<sup>5</sup> listet Standardformate auf, die von Organisationen, wie der ISO aber auch einzelnen Firmen, wie Intel definiert wurden. Sie sind typischerweise nicht an bestimmte Applikationen gebunden.
- Global Digital Format Registry (GDFR) *"... will provide sustainable distributed services to store, discover, and deliver representation information about digital formats."*<sup>6</sup> Diese Format-Registry orientiert sich an den Aspekten der Langzeitarchivierung und nimmt Bezug auf [Hedstrom u. a. 2003].
- Representation Information Registry Repository<sup>7</sup> dient zur Klassifikation von Software, der Auflösung benötigter Bibliotheken und Komponenten. Es können Struktur- und semantische Informationen hinzugefügt werden.
- Library of Congress Digital Formats<sup>8</sup> listet anhand der typischen Dateiendung eine umfangreiche Menge an Metadaten zu jedem Format auf. Hierbei sind neben den klassischen Metadaten, wie Beschreibung, Gebrauchszeitraum, sicherlich die "sustainability factors" von besonderer Bedeutung.
- File Format Encyclopedia.<sup>9</sup> Die Seite ist recht alt und knapp gehalten: Dateien werden anhand ihrer typischen Endungen identifiziert und ihren Erstellungsapplikationen zugeordnet. Die letzte Aktualisierung fand 2001 statt.

<sup>3</sup>Quelle - <ftp://ftp.astron.com/pub/file>. Im Paket sind die diversen Formatkennungen (Magic Numbers - <http://www.magicdb.org>) enthalten.

<sup>4</sup>Siehe dazu <http://filext.com>.

<sup>5</sup>Homepage - <http://magicdb.org/stdfiles.html>. Letzte Aktualisierung erfolgte 2005.

<sup>6</sup>Selbstbeschreibung auf der Homepage - <http://hul.harvard.edu/gdfr>, siehe zudem [JaJa 2006], S. 285.

<sup>7</sup>Homepage - <http://registry.dcc.ac.uk/omar> und siehe zudem das Wiki zu "Representation Information in the DCC Registry/Repository Version 0.4" auf <http://dev.dcc.rl.ac.uk/twiki/bin/view/Main/DCC-RegRepV04>.

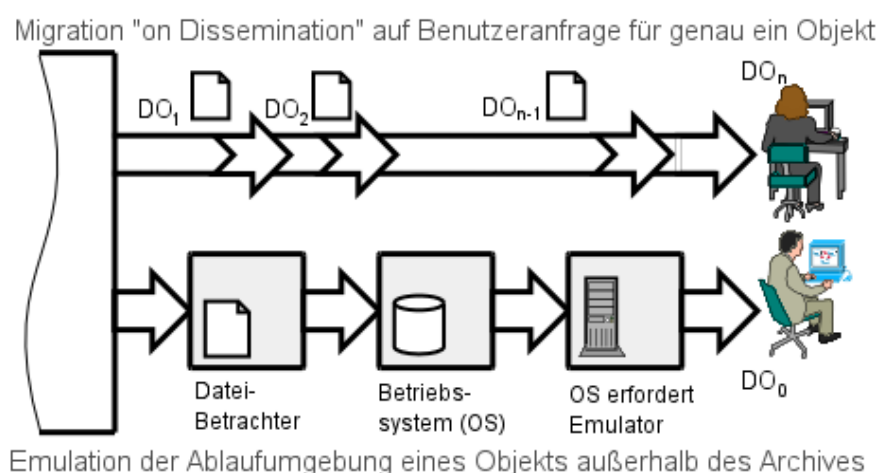
<sup>8</sup>Siehe [http://www.digitalpreservation.gov/formats/fdd/browse\\_list.shtml](http://www.digitalpreservation.gov/formats/fdd/browse_list.shtml).

<sup>9</sup>Homepage - <http://pipin.tmd.ns.ac.yu/extra/fileformat>.

Mit hoher Wahrscheinlichkeit wird sich ein offener, weil von vielen genutzt und aktualisierter Standard wie PRONOM mittelfristig durchsetzen. An diesem Punkt können mit Langzeitarchivierungsstrategien befaßte Gedächtnisorganisationen proaktiv wirken. Das hat dann wiederum den Vorteil, dass auf Standards basierende Werkzeuge und Ergebnisse über Organisationen hinweg ausgetauscht werden können.

## 6.2 Objektzugriff

Zu einem gegebenen Zeitpunkt wird ein Primärobjekt von einem Archivbenutzer nachgefragt werden. In jedem Fall 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 Zustän-



**Abbildung 6.2:** Für jedes dem Archiv entnommene digitale Objekt sind Arbeitsabläufe zu organisieren, die eine Ausführung oder den Zugriff seitens des Archivnutzers erlauben.

digkeit endet nicht mit der erfolgreichen Bewahrung und Fortschreibung der Archivobjekte, sondern er muss zudem den Zugriff auf alle im Archiv enthaltenen Objekte zu jedem Zeitpunkt sicherstellen.

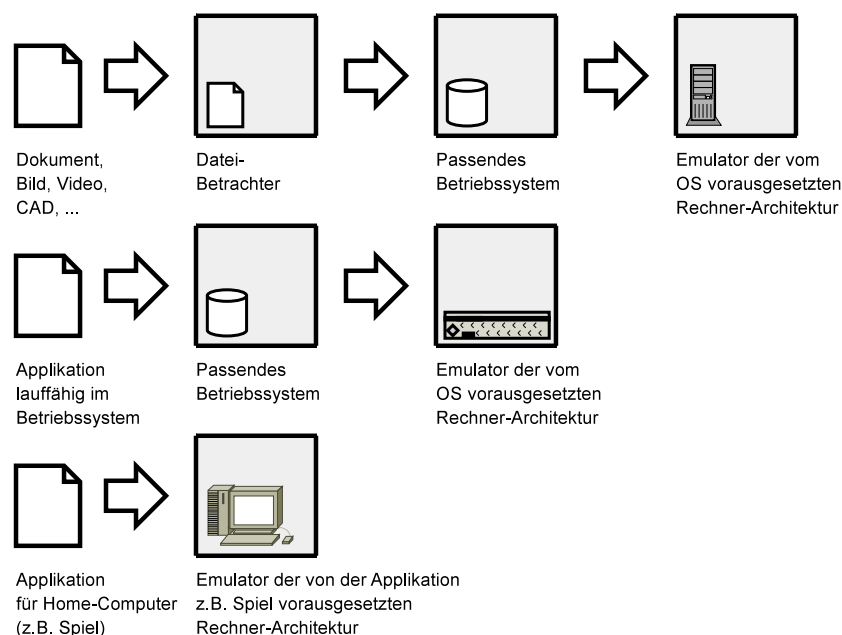
Hierzu wird eine geeignete Umgebung erwartet, in der ein Archivnutzer das Objekt betrachten oder ausführen kann. Die Benutzergruppen einzelner digitaler Archive und Sammlungen werden sich im Grad ihrer Kenntnisse unterscheiden. Da bei einem durchschnittlichen Nutzer nicht zwingend von einem erfahrenen 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 (Abb. 6.2):

- 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, wie in Abschnitt 3.2 dargestellt, nur für statische Objekte. 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 Ansprüchen nicht mehr genügen.

- **Wiederherstellung einer Nutzungsumgebung.** Für alle anderen vorgestellten Objekttypen (Abschnitt 3.2) verbleibt als Darstellungsoption lediglich die Wiederherstellung der ursprünglichen oder einer äquivalenten Objektumgebung. Dieser Vorgang hängt wiederum vom konkreten Objekttyp und seinen spezifischen Anforderungen ab. Die involvierten Prozesse werden in den meisten Fällen eine gewisse Nutzerinteraktion erfordern und zudem eine bestimmte Anzahl zusätzlicher, sekundärer Objekte benötigen (Abb. 6.3). Darüber hinaus stellt sich die Frage des geeigneten Transports des Primärobjekts in die wiederhergestellte Ablaufumgebung.



**Abbildung 6.3:** Erforderliche Schritte zum Rendering/Ablaufenlassen digitaler Objekte in Abhängigkeit ihres Typs.

Die konkrete Gestaltung des jeweiligen Zugriffs hängt vom Archivbetreiber und seinen Richtlinien, als auch von seiner Benutzergruppe ab. Dieses kann "vor Ort" beispielsweise durch eine Referenz-Workstation oder durch einen geeigneten Transport über Kommunikationsnetze geschehen und wird Gegenstand der folgenden Abschnitte sein.

### 6.2.1 Zugriff auf nicht-archivierte Objekte

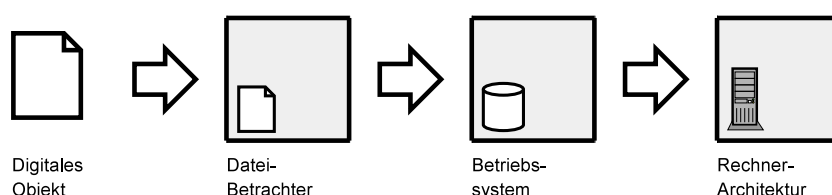
Die bisher getroffenen Überlegungen gelten nicht ausschließlich im Zusammenhang mit einem Langzeitarchiv. So kann es durchaus vorkommen, dass aus unterschiedlichen Quellen digitale Objekte auftauchen, die bisher nicht in einem digitalen Langzeitarchiv gespeichert wurden. Hierzu können die "privaten" Daten von Computernutzern zählen, die im Zuge allfälliger System-Updates nicht berücksichtigt wurden. Auf diese lassen sich die eingangs vorgestellten Arbeitsabläufe der Objekttypbestimmung und der Darstellung für den Benutzer ebenfalls anwenden. Das Objekt würde dann des DIP in die bestehenden Abläufe eingespeist werden. Auf diese Weise könnten etablierte Gedächtnisorganisationen, wie Bibliotheken oder technische Museen, bereits vorhandene Werkzeuge für "Softwarearchäologen" bereitstellen. 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.

## 6.3 Der Weg zum Objekt

Die Wiederherstellung von Nutzungsumgebungen oder entsprechender Äquivalente lässt sich am besten durch sogenannte "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<sup>10</sup> wurde dieses Konzept in die Diskussion eingebracht.

View-Paths sind im Modell von [van Diessen und Steenbakkers 2002], Kapitel 6 konkrete Realisierungen des abstrakten Preservation Layer Model (PLM). Die Abbildung 6.3 zeigt einen typischen View-Path ausgehend vom Primärobjekt, über seine Erstellungssapplikation, zum erforderlichen Betriebssystem bis hin zum daraus resultierenden Hardwareemulator. Während



**Abbildung 6.4:** Genereller View-Path zum Rendering/Ablaufenlassen eines digitalen Objekts.

der Ausgangspunkt des View-Paths durch das Primärobjekt fixiert ist, wird sich, erzwungen durch den technologischen Fortschritt und die sukzessive Obsoleszenz vorhandener Rechnerplattformen, der Endpunkt des View-Path im Zeitablauf verschieben.

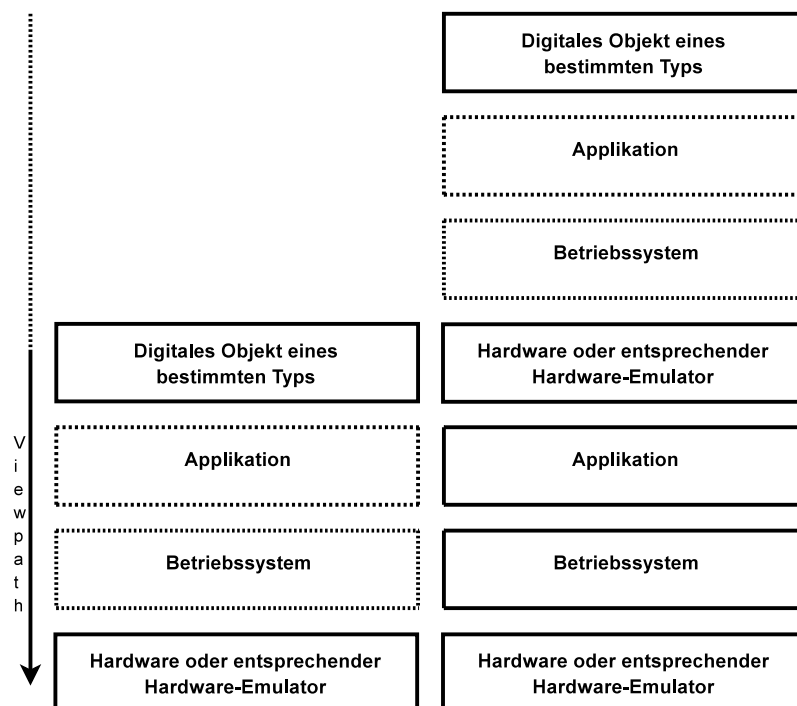
<sup>10</sup>Siehe dazu das Konzept des Preservation Managers in [http://www.kb.nl/hrd/dd/dd\\_onderzoek/preservation\\_subsystem-en.html](http://www.kb.nl/hrd/dd/dd_onderzoek/preservation_subsystem-en.html) auf [KB 2007] und die ausführliche Darstellung in [van Diessen 2002], S. 14 f.

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, die am Ende des Abschnitts erörtert werden.
- Es kann Primärobjekte geben, zu denen zu bestimmten Zeitpunkten keine View-Paths konstruierbar sind.

**Referenzumgebung** Zur sinnvollen Bestimmung der Existenz von View-Paths sollten sie sich auf bestimmte Referenzumgebungen mit jeweils festgelegten Eigenschaften aus Hard- und Software beziehen.<sup>11</sup>

Einen View-Path kann man als Entscheidungsbaum interpretieren, an dessen Wurzel das Primärobjekt steht. Ein Blatt ohne weitere Verzweigungen stellt das Ende des Pfades in Form einer gültigen Referenzumgebung dar. Zur Veranschaulichung des Aufbaus der benötigten Ablaufumgebung lässt sich wiederum ein Schichtenmodell vorstellen, wie es Abbildung 6.5 zeigt. Viele Primärobjekte lassen sich durch mehr als eine Applikation (Viewer) darstellen.



**Abbildung 6.5:** View-Path in der Darstellung als Schichtenmodell. Je nach Objekt muss nicht jede Schicht durch eine Softwarekomponente repräsentiert sein.

Dabei können die Anzeigeergebnisse in Authentizität, Komplexität oder Qualität differieren.

<sup>11</sup>Vgl. dazu auch [van Diessen 2002], S. 16, 17.

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 (vgl. Abb. 6.6). Da Referenzumgebungen extern durch den technologischen Fortschritt 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 Treiber (Abschnitt 7.5) 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 einfache Plattformen, wie Home-Computer, wo keine Trennung zwischen Betriebssystem und Applikation vorliegt. Darüber hinaus können Schichten wiederum gestapelt sein, wenn es für einen bestimmten Emulator erforderlich wird, seinerseits eine geeignete Ablaufumgebung herzustellen, was im rechten Teil von Abbildung 6.5 illustriert wird.

**Metrik** Eine sinnvolle Erweiterung des etwas starren Ansatzes im ursprünglichen DIAS-Preservation-Modell von [van Diessen und Steenbakkers 2002], 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 die Authentizität der Darstellung<sup>12</sup> 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 mit den Objektmetadaten gespeichert und durch das Archivmanagement regelmäßig aktualisiert werden. Hierzu sollte eine Rückkopplung mit den Archivbenutzern erfolgen. So könnten in die Aktualisierungen Bewertungen seitens der Nutzer einfließen, 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, denn der Archivbenutzer, ergibt sich aus unterschiedlichen Lokalisierungen. Damit ist die Anpassung der Software an bestimmte Sprachräume und Einheiten gemeint.<sup>13</sup> Lange Zeit wurde von kommerziellen Softwareanbietern die Landessprache in Menüführung und Benutzerdialogen

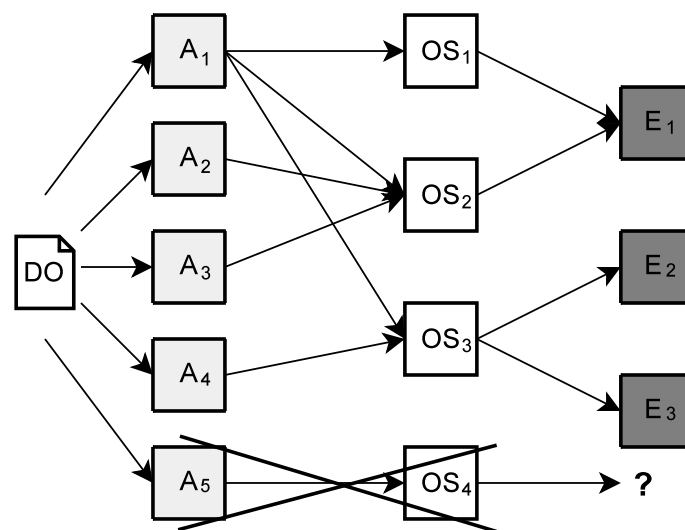
<sup>12</sup>Vgl. hierzu [van Diessen und van der Werf-Davelaar 2002]

<sup>13</sup>Hierzu zählen verschiedene Währungen und ihre Zeichen, Maße oder das Format und Aussehen der Datumsangabe, die Berechnung und Zahl von Feiertagen.

als Distinktionsmerkmal verschiedener Märkte genutzt.<sup>14</sup> 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. Die Präferenzen seitens der Nutzer ließen sich wiederum über eine zusätzliche Metrik abbilden.

## 6.4 Aggregation von View-Paths

Wie soeben dargestellt, 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 sogenannten Office-Pakete, Zusammenstellungen aus verschiedenen Applikationen: Sie können mit einer Vielfalt von Formaten umgehen – nicht nur mit den der enthaltenen Teilkomponenten, sondern darüber hinaus durch Importfilter mit einer Reihe weiterer Dateiformate. Diese Betrachtungen können helfen, eine zusätzliche Referenz-



**Abbildung 6.6:** Redundante View-Pathes zum Rendering/Ablaufenlassen 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.

plattform 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 ablauffähig ist.

<sup>14</sup>Das konnte dazu führen, dass es absurde Preisunterschiede für das funktional gleiche Office-Paket in deutscher und englischer Sprache gab. Dieses ist in den meisten Fällen für die Darstellung der Primärobjekte nachrangig, da sich üblicherweise unabhängig von der Benutzerlokalisierung Module zur Darstellung anderssprachlicher Objekte installieren lassen.



Schwieriger wird jedoch die Aggregation, wenn von den alternativen View-Paths nicht bekannt ist, ob sie ebenfalls zu 100% das gefragte Objekt rekonstruieren. Ein typisches Beispiel ist der Import von Word-Dokumenten in einer anderen als der Erstellungsumgebung. An diesem Punkt könnten ebenfalls die im vorangegangenen Abschnitt vorgenommenen Überlegungen zu Benutzerrückmeldungen in Betracht gezogen werden.

Auf der Ebene der Emulatoren und Virtualisierer lässt sich ebenfalls ein gewisser Austausch untereinander vornehmen. Die Kompatibilitätsmatrix in Abb. 6.7 zeigt, welche Versionen mit jeweils Containerformaten eines anderen Erstellers umgehen kann. So lassen sich Zwischenup-

| Emu/Virt.          | kann lesen/schreiben: |       |     |     |     |     |     |     |     |      |      |      |      |
|--------------------|-----------------------|-------|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|
|                    | (1)                   | (2)   | (3) | (4) | (5) | (6) | (7) | (8) | (9) | (10) | (11) | (12) | (13) |
| VM WS 3.2 (1)      | rw                    | —     | --  | --  | —   | --  | --  | —   | --  | —    | --   | --   | —    |
| VM WS 4.0 (2)      | rw                    | rw    | --  | --  | —   |     | --  | —   | --  | —    | --   | --   | —    |
| VM WS 4.5 (3)      | rw                    | rw    | rw  | --  | —   | --  | --  | —   | --  | —    | --   | --   | —    |
| VM WS 5.0 (4)      | rw                    | rw    | rw  | rw  | —   | --  | --  | —   | --  | —    | --   | --   | —    |
| VM WS 5.5 (5)      | rw(u)                 | rw    | rw  | rw  | rw  | --  | --  | —   | --  | —    | --   | --   | —    |
| VM WS 6.0 (6)      | --                    | rw(u) | rw  | rw  | rw  | rw  | --  | —   | --  | —    | --   | --   | —    |
| Virt PC 04 (7)     | --                    | —     | --  | --  | —   | --  | rw  | rw  | rw  | —    | --   | --   | —    |
| Virt PC 07 (8)     | --                    | —     | --  | --  | —   | --  | rw  | rw  | rw  | —    | --   | --   | —    |
| Virt Server 05 (9) |                       | —     | --  | --  | —   | --  | rw  | rw  | rw  | —    | --   | --   | —    |
| Bochs 2.2 (10)     | --                    |       | --  |     |     |     |     |     |     | rw   | r(w) | --   | —    |
| QEMU 0.9.X (11)    | r                     | r     | r   | r   | r   | r   | r   | r   | r   | r    | rw   | r    | —    |
| Para WS (12)       | --                    |       |     |     |     |     |     |     |     |      |      | rw   |      |
| Virt Box 1.5 (13)  | rw                    | rw    | rw  | rw  | rw  | rw  | --  | —   | --  | —    |      |      | rw   |

**Abbildung 6.7:** Für bestimmte Szenarien kann es interessant sein oder notwendig werden, "fremde" Containerformate mit einem Virtualisierer oder Emulator zu öffnen. Darüber hinaus existieren Konverterprogramme zur Formatwandlung, die hier nicht berücksichtigt wurden.

dates überspringen oder alternative View-Paths für einige Szenarien eröffnen. Der Abschnitt 5.7 enthält zudem einige Betrachtungen zu multiplen View-Paths.

### 6.4.1 Ökonomie und Aufwand

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<sup>15</sup> 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 Windows 3.11 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.

<sup>15</sup>In solchen Fällen könnten bestimmte Datenformate abgelehnt werden, weil deren spätere Rekonstruktion zu hohe Kosten erwarten lässt.

- Wegen der guten und vollständigen Beschreibung des Formats ist es deutlich einfacher, einen Viewer für die jeweils aktuellen Host-Systeme zu migrieren, als alte Nutzungsumgebungen durch Emulatoren zu sichern.
- Dieser Objekttyp ist der einzige, der eine Windows 3.11 Umgebung potenziell verlangt.
- Es gibt kein spezielles Interesse, diese Nutzungsumgebung aufrecht zu erhalten, da dieses nicht im Fokus der Institution 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<sup>16</sup> existiert, könnte ein solcher Zweig im Archiv geschlossen werden.

Ähnlich liegt der Fall in der trivialen Vervielfachung der View-Paths durch unterschiedliche Lokalisierungen. Während Open-Source-Betriebssysteme und Applikationen schon lange mehrere Sprachpakete in einer Installation erlauben, setzte dieses sich bei kommerziellen Systemen erst recht spät durch. Das Bedürfnis von einem Betriebssystem wie Windows 2000 alle verschiedenen Landesversionen aufzubewahren, lässt sich vermutlich besser in Kooperationen ähnlicher Institutionen<sup>17</sup> über Landesgrenzen hinweg erreichen. Dieses würde zudem die Kosten und Aufwand für redundante Lizenzen reduzieren.

Durchaus anders kann der Fall liegen, wenn an anderer Stelle sekundäre Archivobjekte selbst ins Zentrum des Interesses rücken. So sind Computerspiele ganz eigene Zeugen ihrer Epoche oder man möchte nachvollziehen, wie sich die grafischen Oberflächen im Laufe einer bestimmten Periode entwickelt haben. Deshalb beeinflussen Institutionen und ihre Nutzer<sup>18</sup> ebenfalls das Aussehen von View-Paths.

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 (siehe Abschnitt 4.8.1 und 4.8.2) generieren unterschiedliche 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 modulare Ansätze generieren eher kurze View-Paths bei einfacherem Zugriff auf das Primärobjekt. Jedoch liegt der Aufwand im regelmäßigen Update aller benötigter Emulatoren oder ihrer Teilkomponenten.

<sup>16</sup>Weil die Art der Benutzerinteraktion, das Aussehen der grafischen Oberfläche und der Applikation in dieser von eigenständigem Interesse sind. Besonders gut lässt sich dieses 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.

<sup>17</sup>Vgl. Abschnitt 2.6: Inzwischen hat die Kooperation auf dem Gebiet der Langzeitarchivierung Tradition.

<sup>18</sup>Siehe Abschnitt 4.5.

Die erstgenannte Strategie eignet sich daher eher für Objekte mit seltenem Zugriff oder Institutionen mit kleinen, speziell ausgebildeten Nutzerkreisen, wie Archive. Die zweite Strategie ist sicherlich für vielgenutzte Objekttypen und größere Anwenderkreise vorzuziehen.

Eine Reihe von Kostenfaktoren kann durch gemeinschaftliche Anstrengungen und verteilte Aufgaben reduziert werden. Hierzu zählen beispielsweise die Entwicklung des freien X86-Emulators "Dioscuri" oder die Formatregistratur "PRONOM".<sup>19</sup>

## 6.5 Referenzumgebungen

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

Jede historische bis zu den heutigen Rechnerplattformen weist ihre eigenen Komplexitäten auf, die nicht von jedem durchschnittlichen Computeranwender sinnvoll bewältigt werden können. Zudem kann es nicht 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<sup>20</sup> werden deshalb Betreiber digitaler Langzeitarchive nach Lösungen suchen, die es erlauben, geeignet auf ihre jeweiligen Primärobjekte 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:

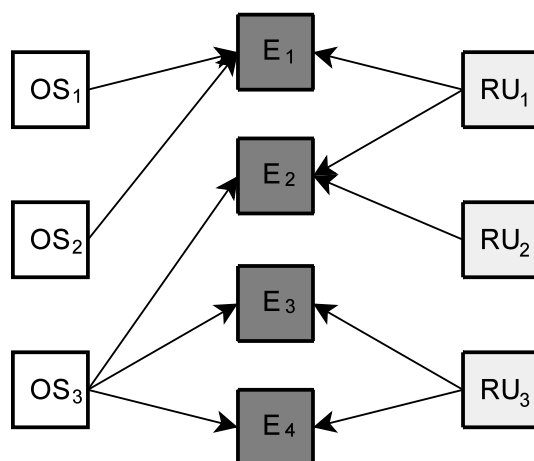
- Referenzplattformen aus einer bestimmten Hardware und Software werden als Endpunkte von View-Paths benötigt. Sie können ortsnahe zum Langzeitarchiv, beispielsweise parallel oder als Erweiterung der üblichen Recherchesysteme der jeweiligen Gedächtnisorganisation, aufgestellt oder in diese integriert 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. Hierdurch erreicht man einerseits eine breitere Auswahl von Referenzumgebungen bei einer stabilen Schnittstelle zur virtuellen Maschine. Andererseits verlagert man damit das Problem eine Schicht weiter nach unten und hängt nun von der Weiterentwicklung virtueller Maschinen ab.

Eine Einordnung von Referenzumgebungen für den lokalen Zugriff und das Angebot über das Internet bietet [van Diessen und van Rijnsoever 2002], S. 7.

<sup>19</sup>Gefördert durch staatliche oder EU-Programme, vgl. hierzu Abschnitt 2.6 im ersten Teil der Arbeit.

<sup>20</sup>Vgl. hierzu Abschnitt 4.5

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, was durch den jeweils eingesetzten Emulator<sup>21</sup> und seine Benutzer-Interfaces in der jeweiligen Referenzumgebung mitbestimmt wird.



**Abbildung 6.8:** Die Auswahl der Emulatoren (E) wird in vielen Fällen nicht nur durch das Betriebssystem (OS) im View-Path sondern auch von der anderen Seite durch die verfügbaren Referenzumgebungen (RU) beeinflusst.

Andererseits können Referenzumgebungen als Endpunkte eines View-Path diesen rückwärts mitbestimmen (Abb. 6.8). Wenn technische, finanzielle oder personelle Restriktionen bestimmte Referenzumgebungen, die potenziell nachgefragt werden, nicht erlauben, kann dieses die Wahl eines Emulators wesentlich beeinflussen. Die Rekursion kann sich unter Umständen noch weiter auf frühere Entscheidungsknoten auswirken.

### 6.5.1 Lokale Workstations

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 machbar, 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 Referenz-Workstation 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

<sup>21</sup>Siehe hierzu "Technische Auswahlkriterien" in Abschnitt 4.7.

Objekte sind eher statischer Natur und wurden typischerweise auf PC's verschiedener Generationen erstellt. Generell muss es sich bei den eingesetzten Referenz-Workstations nicht um die jeweils allerneueste Hardwaregeneration handeln (Abb. 6.9). Stattdessen sollte die Technologie angestrebt werden, die einen optimalen Austausch erlaubt und den Anforderungen der jeweiligen Nutzer gerecht wird.

Darüber hinaus muss ein ausreichendes Bedienungswissen vorhanden sein, 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 Referenz-Workstations überspringen (Abb. 6.9).

Bezogen auf die heutige Zeit und die Verfügbarkeit bereits vorhandener Emulatoren spielen zwei Referenzumgebungen für die X86-32 bit und 64 bit-Architektur eine hervorgehobene Rolle:

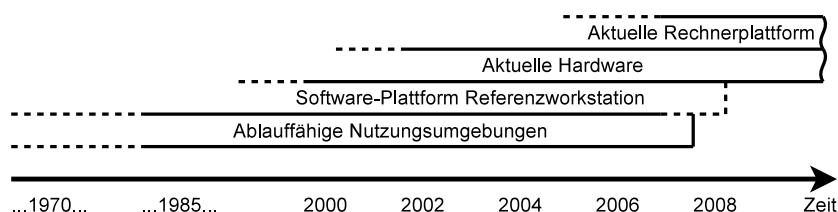
- Windows zählt zu den verbreitetsten PC-Betriebssystemen. Für diese Betriebssystemklasse stehen die meisten Viewer, Emulatoren und Virtualisierer zur Verfügung. Es ist jedoch proprietär und durch aktuelle Verfahren der Lizenzsicherung seitens des Herstellers kann die Gefahr der Nichtverfügbarkeit dann älterer Versionen zu späteren Zeitpunkten steigen. Die Betriebssysteme sind Closed-Source und deshalb unter Umständen schwerer an bestimmte Einsatzszenarien anzupassen. Sie generieren Lizenzkosten, die in jedem Fall aufzuwenden sind.
- Linux ist das derzeit populärste Open-Source-Betriebssystem für den PC. Es existiert eine große Auswahl an Viewern, Emulatoren und Virtualisierern. Da für viele Aufgabenstellungen Software typischerweise erst kompiliert wird, steht eine vollständige Entwicklungsumgebung zur Verfügung, die in Problemlagen Anpassungen an Programmen und Betriebssystem erlaubt. Die Flexibilität ist höher: Durch Backports aktueller Treiber lassen sich auch ältere Linux Systeme auf aktueller Hardware oftmals sinnvoll betreiben.

Nicht in allen Fällen wird es gelingen, sämtliche Emulatoren und Virtualisierer parallel auf einem Host-System zu installieren oder es ist ein höherer Aufwand zu kalkulieren. So lassen sich verschiedene Versionen der VMware-Produkte nicht einfach nebeneinander einrichten. Hier gibt es Konflikte bei den benötigten Kernel-Modulen, installierten Bibliotheken und Konfigurationsdateien.

Je nach angestrebtem Betriebsmodell kann man es Benutzern erlauben, selbst Software in einer Referenz-Workstation zu installieren, da sie das primäre Objekt des Interesses ist. Generell sollte man deshalb den Begriff "Referenz" als eine definierte Plattform auffassen, die möglichst nach jeder Nutzung und Veränderung wieder in den alten Zustand zurück überführt wird. Hierfür bieten sich beispielsweise Stateless-Systeme<sup>22</sup> an, die sich mit Linux oder durch den Einsatz von Virtualisierern<sup>23</sup> leicht realisieren lassen. Performance-Probleme,

<sup>22</sup>Zustandslos meint in diesem Fall, dass bei jedem Neustart der Referenz-Workstation diese komplett neu konfiguriert, eingerichtet und dabei nicht auf lokal vorhandene Konfigurationen zurückgegriffen wird.

<sup>23</sup>Einige Virtualisierer bieten einen non-persistent Mode oder Immutable Disks für Zugriffe auf ihre virtuellen Festplatten an, die keine Veränderungen in der Containerdatei vornehmen, sondern diese in separate Dateien schreiben.



**Abbildung 6.9:** Die Referenz-Workstation kann eine gewisse Zeit den jeweils aktuellen Rechnerplattformen hinterherlaufen, solange noch eine gesicherte Hardwarebasis und Benutzerkenntnisse der Bedienung existieren.

beispielsweise durch die Ausführung mehrerer Instanzen eines Emulators, sind hingegen kaum zu erwarten, da es sich typischerweise um Einbenutzersituationen handelt.

Ein weiterführender Gedanke könnte die Virtualisierung dieser Workstations sein, so dass sie einerseits als zentraler Service eines spezialisierten Rechenzentrums über Weitverkehrsnetze<sup>24</sup> angeboten oder andererseits als vorkonfiguriertes Paket – dieses wird in Abschnitt 6.6 ausführlich behandelt – verteilt werden könnten. Für diesen Fall muss man jedoch beachten, dass einige Virtualisierer keine "Stacked Emulation"<sup>25</sup> zulassen.

Sollen die Referenz-Workstations dem Nutzer erlauben, eigene digitale Objekte zu öffnen,<sup>26</sup> kann es notwendig sein, die Maschine mit den typischen Wechselaufwerken, wie sie im Laufe der Zeit am Markt waren, auszustatten.

### 6.5.2 Virtuelle Maschinen

Was für Softwareentwickler oder Mitarbeiter in Support-Zentren gilt, lässt sich ebenso auf Archive und Bibliotheken übertragen: In vielen Fällen kann es notwendig sein, viele verschiedene Nutzungsumgebungen in einer Maschine abzubilden oder sehr schnell auf diese zugreifen zu können. Diese Anwendung war einer der Bereiche, mittels derer sich die Virtualisierer der X86-Plattform profilieren konnten.

Solche Nutzungsumgebungen können ihrerseits bestimmte Emulatoren enthalten, die genau für diese geschrieben wurden und nicht mehr für die Referenzumgebung verfügbar sind. Das entspricht weitgehend dem im Abschnitt 4.8.2 erörterten Konzept der geschachtelten Emulation. Generell lässt sich auf diese Weise eine definierte Schnittstelle für die Benutzerpräsentation schaffen.

Es kann durchaus sinnvoll sein, bestimmte Nutzungsumgebungen in virtuellen Maschinen vorbereitet verfügbar zu halten. Sie lassen sich schnell zur Benutzung starten oder bequem über Datennetze austauschen, verbreiten und von mehreren Institutionen gemeinsam nutzen. Diese vorbereiteten Systeme werden in dieser Arbeit als "cached" View-Path bezeichnet und in Abschnitt 6.6 ausführlicher behandelt. In Teilen ähnlich gelagert ist das Konzept der entfernten Benutzersteuerung von Emulationsumgebungen.

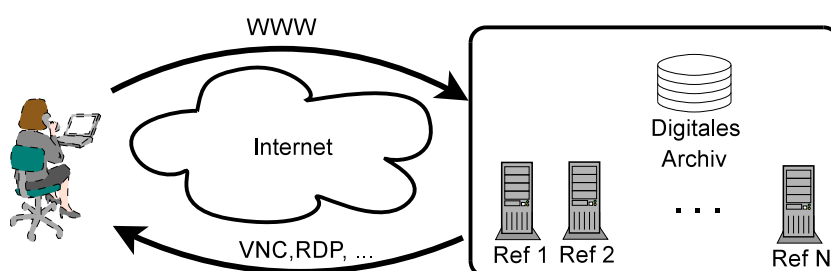
<sup>24</sup>Siehe hierzu die Anmerkungen zum VMware Server in Abschnitt 5.3.9.

<sup>25</sup>Es gelingt zwar QEMU in einer VMware auszuführen, jedoch beispielsweise nicht den VMware Player innerhalb eines VMware Servers oder einer Workstation. Für eine abstrakte Erklärung der Ineinanderschachtelung vgl. Abschnitt 4.8.2, für die Unmöglichkeit des VMware-Stackings Abschnitt 5.2.

<sup>26</sup>Vgl. hierzu die Gedanken in Abschnitt 6.2.1 zu "Softwarearchäologie".

### 6.5.3 Zentralisierte Systeme mit Fernzugriff

Die Digitalisierung vieler Arbeitswerkzeuge hat die Möglichkeiten der Nutzung von Bibliotheken und Archiven revolutioniert. Wo es bis zur allgemeinen Durchsetzung von Computern und Datennetzen durchaus üblich war, sich zur Recherche und Einsicht in Primärobjekte vor Ort zu begeben, können nun etliche Dienste entfernt abgerufen werden. Die bereits bestehende Palette könnte um web-basierte Services zum Laden von Primärobjekten in vorbereitete oder online erstellte Nutzungsumgebungen erweitert werden, die über das Internet angeboten werden. Auf diese Weise ließen sich eine Reihe von Problemfeldern bearbeiten:



**Abbildung 6.10:** Die Interaktion mit Nutzungsumgebungen alter Rechnerplattformen kann über das Internet realisiert werden. Auf diese Art können digitale Objekte bequem auch über große Distanzen präsentiert werden. [Welte 2008]

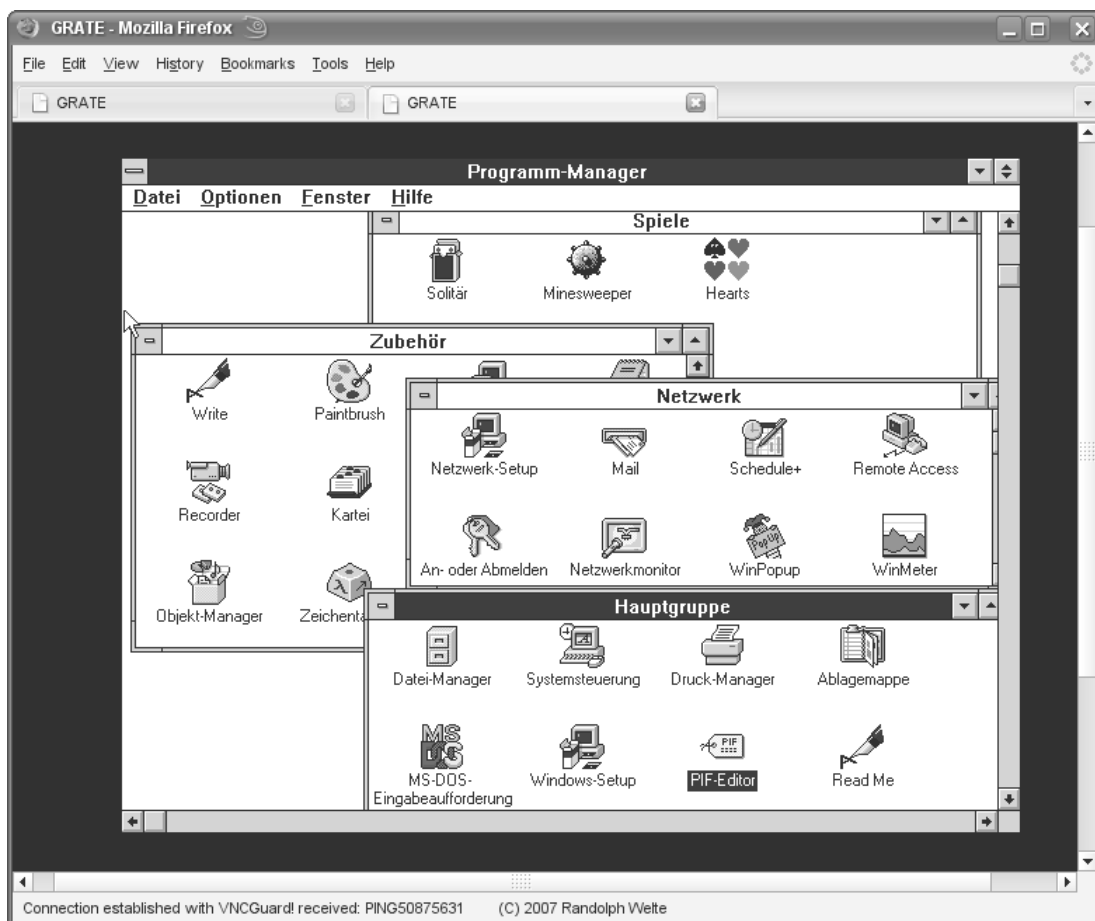
- Der Zugriff auf die Objekte lässt sich einer breiteren Nutzerschaft öffnen, als über die reine Verfügbarmachung an lokalen Ressourcen in Form von Referenz-Workstations.
- Der Benutzer muss über kein ausgewiesenes Wissen zur Installation einer Reihe von Spezialprogrammen, wie Emulatoren und ihrer Hilfswerkzeuge verfügen.
- Das Wissen zum Betrieb eines solchen Dienstes kann zentralisiert werden, da ein Benutzer lediglich bekannte Techniken, wie den Web-Browser und Erweiterungen auf seiner Maschine vorhalten muss.
- Problemen des Lizenz- und Digital Rights Management (DRM) werden aus dem Weg gegangen, da die Primär- und Sekundärobjekte nicht auf die Maschinen der Nutzer übertragen werden müssen.
- Es bleiben Fragen der optimalen Darstellung der Ausgaben, Interaktion, Übertragung von Audioströmen und der Datenein- und -ausgabe zu erörtern, da diese komplexer als bei Referenz-Workstations ausfallen.

Für Ausstellungen wie das Computerspiele-Museum Berlin oder ähnlich gelagerte Informationstechnik- und Rechnervorführungen bietet ein solches Verfahren eine interessante Lösung für die Präsentation ihrer Objekte außerhalb der eigenen Räumlichkeiten (Abb. 6.11). Sie reißen sich damit nahtlos in die inzwischen große Zahl verschiedener internet-basierter Angebote ein und greifen dabei auf ihren Benutzern bekannte Software zurück. Durch den Transport von Ein- und Ausgaben über auch große räumliche Distanzen kann mit einigen Problemen in Bezug auf Geschwindigkeit und Verzögerung gerechnet werden, die aber für die

meisten Anwendungen in diesem Bereich eher nachrangig sind und sich mit der zunehmenden Verfügbarkeit höherer Datenraten bei geringeren Latenzen reduzieren dürften. Mit diesen Fragestellungen befasst sich [Welte 2008] ausführlich und präsentiert einen ersten lauffähigen Demonstrator dieser Ideen.

#### 6.5.4 Problemfelder und Herausforderungen

Emulation oder Virtualisierung bedeutet die Abbildung einer Nutzungsumgebung in eine durchaus stark abweichende weitere, hauptsächlich grafische Arbeitsoberfläche mit nicht nur abweichendem Aussehen, sondern vielfach recht unterschiedlichen Bedienkonzepten. Das Tastaturlayout hat sich beispielsweise im Laufe der Computerentwicklung deutlich geändert und wird auch derzeit über die Architekturen hinweg nicht einheitlich gehandhabt. Deshalb finden sich immer noch bestimmte Sondertasten des PC oder Apple Computers auf unterschiedlichen Tasten. Tastaturen sind nicht die einzige Peripherie, die sich unterscheidet. Frühe Rech-



**Abbildung 6.11:** Präsentation einer Windows 3.11 Nutzungsumgebung im Web-Browser auf der Maschine des Archivnutzers. Der von [Welte 2008] erstellte Demonstrator erlaubt neben der direkten Interaktion mittels VNC das Hoch- und Herunterladen von digitalen Objekten.

ner, wie die diversen Großrechner<sup>27</sup> luden Daten und Programme per Lochkarten, -streifen

<sup>27</sup>Vgl. Abschnitt 4.2.



oder Magnetband, welche eine andere Benutzungsphilosophie als Festplatten und Disketten aufwiesen. Die verschiedensten Datenträger sind für die Ausführung in emulierten Umgebungen als Dateien abzubilden. Dieses muss rechtzeitig erfolgen, solange geeignete Hardware noch zur Verfügung steht und wird für diese Betrachtungen als weitgehend erfolgt vorausgesetzt. Mit dieser Fragestellung setzt sich Abschnitt 7.10 auseinander; mit Datenträgern der X86-Generation beschäftigt sich der spätere Abschnitt A.5. Geschwindigkeit und Effizienzerwägungen spielen bei wachsendem Abstand zwischen Referenzumgebung und jeweiliger Nutzungsumgebung eine zu vernachlässigende Rolle.

**Abbildung von Eingabesystemen** erfordert eine Beschäftigung mit zwei miteinander verwandten Aspekten: Unterschiedliche Tastaturlayouts betreffen Lokalisierungen an die jeweilige Landessprache oder landestypische Vorgaben. So unterscheiden sich französische, englische und deutsche Tastaturbelegungen bis hin zu geometrischen Details, wie die Anordnung bestimmter Sondertasten oder die Größe von "Enter". Zudem gab und gibt es sogenannte "Modifier"-Tasten, die neben der klassischen Großstelltaste aus Schreibmaschinenzeiten weitere Belegungen ermöglichen. Aus diesem Grund erwartet beispielsweise der deutsche PC-Benutzer das "@"-Zeichen auf dem Buchstaben "q" in Zusammenspiel mit der "Alt-Gr" Taste, während es auf englischen Tastaturen mit "Shift" und "2" erreichbar ist. Gerade für Pfad-Delimiter der traditionellen Kommandozeile wie ":", "/" oder "'" geht es um deren Erreichbarkeit.

Home-Computer, wie der ZX-81 oder Spectrum brachten es bis auf fünffache Belegungen einer Taste. Um dieses abbilden zu können, müssen geeignete und für den normalen Anwender gut dokumentierte Übersetzungstabellen erstellt werden, die das jeweilige Layout der Referenzumgebung berücksichtigen. Abhilfe oder Ergänzung können zusätzlich eingeblendete On-Screen-Tastaturen sein, wie man sie bei einigen PDAs oder per Touch-Screen bedienbaren Geräten findet.

Während lange Zeit die zweidimensionale Zeigersteuerung durch Computermäuse dominierte, zeigt die aktuelle Spielekonsole von Nintendo "Wii", wo neue Entwicklungen hingehen können: Die anschließbaren Controller erlauben dreidimensionalen Input.

**Abbildung von Ausgabesystemen** Während die ersten Rechner noch über einzeilige Zahlenreihen, Lochstreifen oder Drucker ihre Ausgaben vornahmen, setzten sich im Laufe der Zeit Bildschirme zur direkten Benutzerinteraktion durch. Anfangs konnte auf diese Weise eine bestimmte Anzahl von Zeichen pro Zeile und Zeilen pro Bildschirmseite untergebracht werden. Da die meisten Displays auf der Technik der damals üblichen Kathodenstrahlröhre der Fernseher beruhten, war die Auflösung begrenzt. Ein typischer Umfang waren 80 Zeichen pro Zeile bei 24 Zeilen.<sup>28</sup>

Frühe Bildschirme konnten lediglich Zeichen einfarbig darstellen. Hervorhebungen geschahen durch Invertierung, Unterstreichung oder eine stärkere Helligkeit einzelner Zeichen oder Zeilen. Mitte der 1970er Jahre begann man, Bildschirme für grafische Ausgaben zu nutzen.

Da sich die Bildschirmauflösungen und Farbtiefen heutiger Systeme ständig weiter entwickeln, überfordern die früher dargestellten Auflösungen heutige Anzeigesysteme nicht. Es kommt lediglich darauf an, dass die Umsetzung alter Ausgaben nicht unbedingt pixelgenau

<sup>28</sup>Diese Anordnung findet man immer noch, da sie sich am VT100 orientiert.

erfolgt, sondern auf eine jeweils geeignete Größe skaliert wird. Das kann zwar einen gewissen Aufwand für das Ausgabesystem bedeutet, welcher bei den heutigen Rechen- und Grafikleistungen zu vernachlässigen sein dürfte.

Spezielle Probleme kann man jedoch für Aufgabenstellungen, wie 3D-Ausgaben erwarten: Entsprechende Programme greifen auf die Programmierschnittstellen der 3D-Hardware für ihre Ausgabe zu. Bisher ist diese Fähigkeit noch nicht oder erst in Ansätzen<sup>29</sup> realisiert worden.

Deutlich problematischer sind hingegen Kopierschutzketten zur Darstellung hochauflösender Filme und Videos zu sehen. Diese Technik im Emulator nachbilden zu wollen, dürfte einen recht hohen Aufwand bedeuten, für eine Zielsetzung, die besser anders erreicht werden sollte.<sup>30</sup>

Interessant werden vielleicht Überlegungen wie weit in der Zukunft liegende Benutzerschnittstellen aussehen könnten. Im Gedankenexperiment könnte man sich vorstellen, wie man alte Hardwareplattformen in Welten wie "Second Life" präsentieren würde. Hier könnte tatsächlich wieder das virtuelle 3D-Modell eines Bildschirms stehen, in das eine klassische 2D-Ausgabe eingeblendet wird, vor die ein Benutzer treten kann. Selbst bei deutlichen Technologiesprüngen kann davon ausgegangen werden – allein schon auf Basis der vorhandenen Menge alter Systeme, dass die neue Technik so viel mehr ermöglicht, dass die alte als Teilmenge abgebildet werden kann.

## 6.6 Vorbereitete Umgebungen

Die verschiedenen Referenzumgebungen in Form von lokal zugreifbaren Workstations oder über Computernetzwerke angebotene Dienste erfordern je nach Objekttyp sich stark wiederholende Arbeitsabläufe.

Für jedes Dateiformat wird es mindestens einen<sup>31</sup> oder mehrere View-Paths geben. In den meisten Fällen liegen jedoch große Anteile dieser View-Paths sehr eng beieinander und unterscheiden sich lediglich auf der Ebene der Applikationen. Durch Aggregation lässt sich die Zahl der resultierenden Nutzungsumgebungen zusätzlich reduzieren. Einige Überlegungen in dieser Richtung finden sich im Artikel [Reichherzer und Brown 2006].

Für populäre Formate wäre es unter Umständen sehr aufwändig, für jedes Objekt oder jede Benutzersitzung den View-Path jedesmal neu zu generieren, da sich das in den wenigsten Fällen komplett automatisieren ließe. Deshalb ist es sinnvoll, häufig benutzte View-Paths zu "cachen"; somit bereits bestehende Installationen aus einem geeignetem Emulator und den darin vorbereiteten Nutzungsumgebungen für eine gewisse Zeitspanne aufzubewahren. Diese Pakete könnten als Bestandteil des Online-Archivs gesehen werden, wie es in den Abschnitt 7.9 dargestellt wird. Das hätte eine Reihe von Implikationen, wie sie Abschnitt 6.7 näher ausführt.

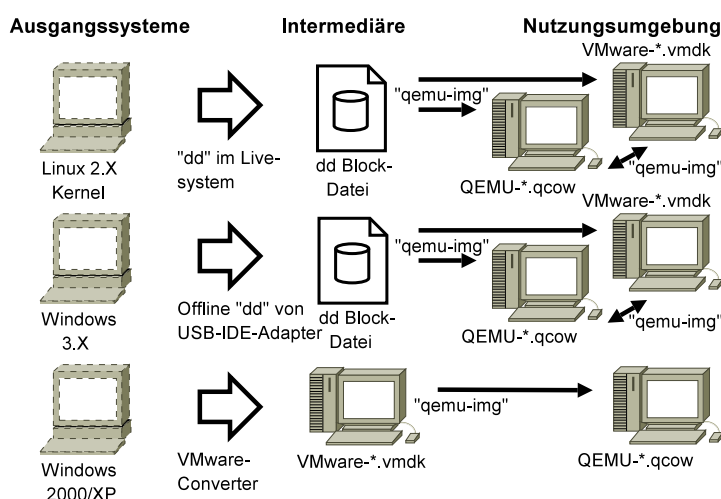
<sup>29</sup>Vgl. VMware "Fusion" [VMware Inc. 2008] und Abschnitt 5.3.10, sowie die Anmerkungen zu QEMUs OpenGL-Patch in Abschnitt 5.6.1.

<sup>30</sup>Deshalb kann VMware nicht auf das Film-DVDs zugreifen und diese in der virtualisierten Umgebung abspielen. Vgl. zudem Abschnitt 7.11.

<sup>31</sup>Dieses muss das Management des Archivs bereits bei der Objektaufnahme (Ingest) sicherstellen.

### 6.6.1 Kombinierte Primärobjekte

Primärobjekte mit bereits im Paket enthaltenen notwendigen Sekundärobjekten, wie bestimmte Metadaten zur Beschreibung, Viewer und Hilfsprogrammen, zusammengefasst in einem Archival Information Package (AIP), können bereits aufbereitet vorgehalten werden. Die Idee hierzu findet sich bereits in [Rothenberg 2002], Abb. 1. Eine weitere Variation beschreiben [Gladney und Lorie 2005], S. 310 ff. Für die Überführung eines physischen Systems in ein



**Abbildung 6.12:** Je nach Ausgangssystem existieren verschiedene Varianten der Archivierung kompletter Nutzungsumgebungen. Kommerzielle Virtualisierer liefern Werkzeuge, wie den VMware Converter, bereits mit oder bieten sie als Erweiterung ihrer Produktpalette an.

virtuelles bieten sich verschiedene Wege an (Abb. 6.12). So kann mit dem Unix-Werkzeug "dd" ein Abzug für QEMU erzeugt werden. VMware bietet für einige Windows-Systeme den VMware Converter an.<sup>32</sup> In den meisten Fällen muss die Treiberausstattung des Systems angepasst und die Boot-Fähigkeit wiederhergestellt werden.<sup>33</sup>

Ein solches Vorgehen lässt sich beispielsweise für sehr spezielle dynamische Objekte, wie Datenbanken von Content Management Systemen (CMS) für WWW-Auftritte, Firmendaten aus Produktionsanlagen oder Entwicklungsabteilungen vorstellen: Statt Daten, Applikation und notwendiges Betriebssystem im Zuge der Archivierung voneinander zu lösen, könnte eine komplette Maschine in eine emulierte Umgebung überführt werden. Damit bleibt die Ser-  
verfunktionalität mit ihren gesamten Einstellungen als eine Einheit erhalten. Dieses wurde im Zuge eines Experiments mit verschiedenen älteren Linux Systemen, die eine MySQL Datenbank oder eine bestimmte Softwareausstattung für den Desktop enthielten, erfolgreich durchgeführt.<sup>34</sup> Das Verfahren kann auf diese Weise sicherstellen, dass ein AIP über Archive hinweg ohne einschränkende Nebenbedingungen ausgetauscht werden kann.

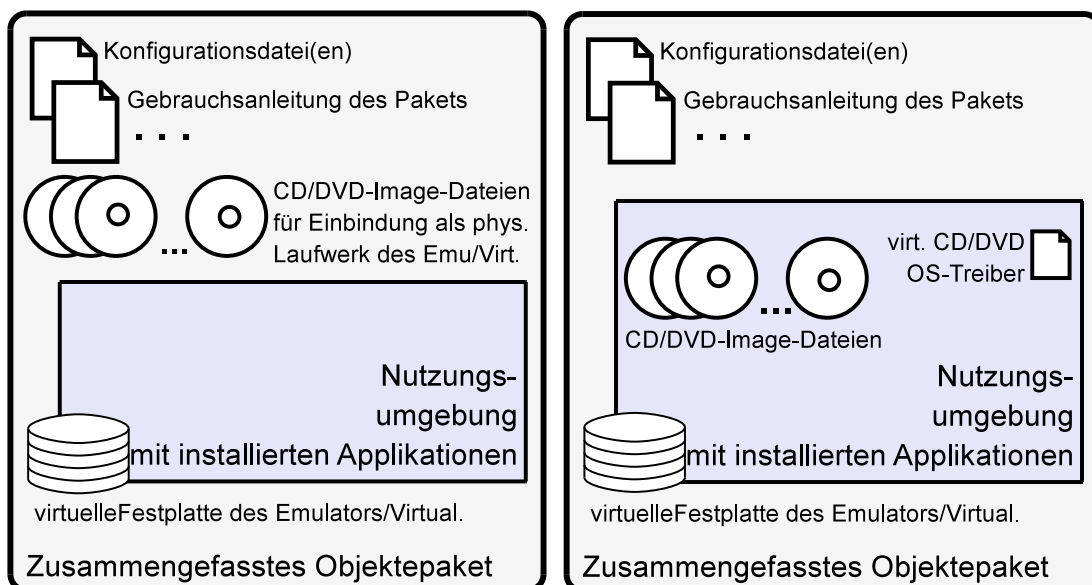
<sup>32</sup> Je nach Werkzeug und System kann der Abzug online vom System selbst oder offline vom (ausgebauten) Datenträger erfolgen. Siehe zu letzterem Abschnitt 7.10.6.

<sup>33</sup> Vgl. hierzu die ähnlich gelagerte Problematik in Abschnitt 5.7.1 und Abschnitt 7.5 zu Treiberfragen.

<sup>34</sup> Es handelte sich bei diesen Untersuchungen und Experimenten um verschiedene Linux-Varianten in Bezug auf Kernel und Softwareausstattung: SuSE 8.2 und 9.0 mit 2.4er Kernel und SuSE 9.3 und 10.2 mit 2.6er Kernel und unterschiedlichen Initial-Ramdisk-Varianten. Enthalten waren verschiedene Datenbanken, CMS oder Desktop-Systeme.

### 6.6.2 Thematische Zusammenfassung

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 zusammenfassen. Dieses erleichtert den Austausch, da oftmals lediglich der Container, die Konfigurationsdatei und Startanleitung des Emulators verteilt werden müssen (Abb. 6.13). 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.



**Abbildung 6.13:** Vorbereitete Objektpakete können unterschiedliche Dateien enthalten. Bei einigen Objekten gelingt es, ISO-Images in die virtuelle Festplatte einzubetten, solange es die zugehörige Applikation und die Nutzungsumgebung erlauben.

Die Idee einer Datenkapsel findet sich in [Rothenberg 1999]. Hier werden relevante Metadaten und Sekundärobjekte, wie Beschreibungen, in einem gemeinsamen AIP zusammengeführt. Eine Anwendung dieses Konzepts schlägt [Huth 2004] vor, um neben alten Computerspielen des Atari VCS 2600 oder C64 auch Bilder der Maschinen, Anleitungen und Benutzungshinweise zu hinterlegen.

Eine etwas anders gelagerte Zusammenfassung bieten [Reichherzer und Brown 2006] an, um auf Datensätze der öffentlichen Administration zu späteren Zeitpunkten zugreifen zu können. 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<sup>35</sup> nach Postscript oder PDF enthalten.

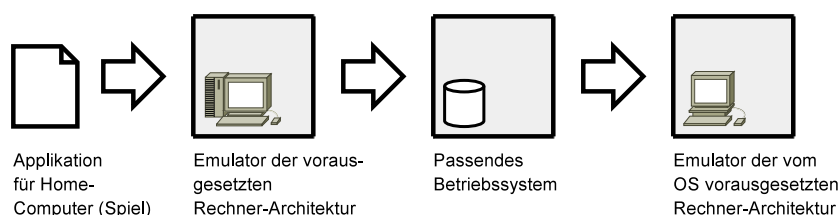
<sup>35</sup>Passende virtuelle Drucker, die durch Postscript- oder PDF-Generatoren repräsentiert werden, können fehlende Exportfilter alter Applikationen ersetzen.

Wiederum für ähnliche Zwecke lassen sich moderne Virtualisierer einsetzen, in denen eine recht aktuelle Nutzungsumgebung installiert ist. Diese Umgebung ist die eigentliche Referenzplattform, in der verschiedene Emulatoren von einer oder mehreren anderen Rechnerarchitekturen installiert sind. Auf diesem Wege können beispielsweise viele Home-Computer-Emulatoren für eine ganze Reihe verschiedener Maschinen gemeinsam mit Primärobjekten wie Spielen oder Anwendungen zusammengefasst sein. Das erspart dem Nutzer oder anderen, an einem Austausch beteiligten Gedächtnisorganisationen, das Zusammensuchen der einzelnen Komponenten und ihrer Abstimmung aufeinander.

Die Pakete können je nach installierter Software oder Datenquellen neben der virtuellen primären Festplatte des Emulators oder Virtualisierers, Konfigurationsdateien oder Kommandozeilen zum Aufruf enthalten. Ebenso könnte eine Reihe ISO-Images mitpaketierte sein, die als CD- oder DVD-Laufwerke konfiguriert werden. Für eine größere Zahl von Images, die beispielsweise für elektronische Datenbanken oder Enzyklopädien notwendig sind, kann man versuchen, diese "innerhalb" der Nutzungsumgebung durch virtuelle CD-Laufwerke einzubinden (Abb. 6.13). Das setzt jedoch ausreichend verfügbaren Speicherplatz im Festplatten-Image und einen passenden Betriebssystemtreiber in der Nutzungsumgebung voraus.<sup>36</sup> Dadurch würde sich die Zahl der im Paket zusammengefassten Dateien verringern und die Konfiguration vereinfachen. Dieses kann jedoch in Abhängigkeit vom Objekt wegen vorhandener Schutzvorrichtungen der jeweiligen Applikation fehlschlagen.

## 6.7 View-Paths im Zeitablauf

Im OAIS-Referenzmodell übernimmt das Management des digitalen Langzeitarchivs eine Reihe von Aufgaben (Abschnitt 3.1.1), die wie in den vorangegangenen Abschnitten 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. Einen guten Einstieg bietet hier [van Diessen und Steenbakkers 2002], Kapitel 6. Ein zentrales Moment ist die



**Abbildung 6.14:** Steht der ursprüngliche View-Path nicht mehr zur Verfügung, da kein Emulator für die aktuelle Host-Plattform existiert, so kann das Problem durch "Stacked Emulation" gelöst werden, was aber den View-Path verlängert.

regelmäßige Kontrolle der View-Paths mit jedem Wechsel der Referenzumgebung als Bezugsgröße. Jeder Plattformwechsel stellt neue Anforderungen für die Wiederherstellung von

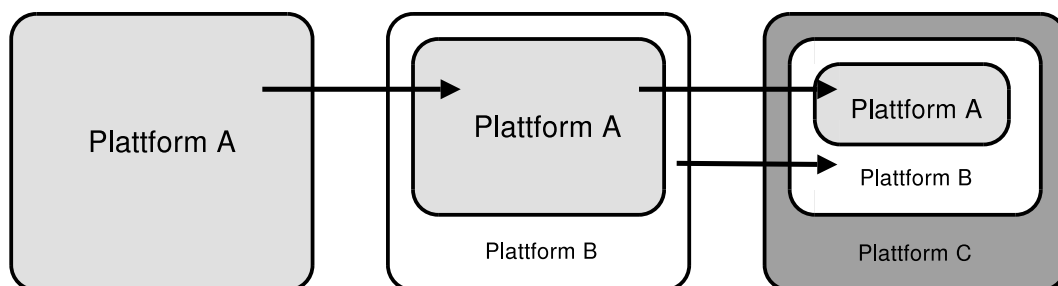
<sup>36</sup>Hierbei kann auf Software wie "VaporCD" - <http://vaporcd.sourceforge.net> oder Ähnliche zurückgegriffen werden, die dann ebenfalls im Softwarearchiv hinterlegt werden sollten.

Nutzungsumgebungen. 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: Neue Formate resultieren in neuen View-Paths.

Da Emulatoren genauso wie alle anderen digitalen Objekte von Veraltung betroffen sind, lassen sich die identischen Strategien anwenden. Die abstrakten Konzepte wurden in Abschnitt 4.8 ausgeführt:

- Migration. Die vorhandenen Emulatoren sind für die neue Umgebung umzuschreiben oder mindestens neu zu kompilieren. Dabei müssen auf jeden Fall die Eigenschaften der emulierten Hardware erhalten bleiben (Abschnitt 4.8.1). Nachteilig wirkt sich bei diesem Ansatz der höhere Aufwand des Archivmanagements aus: Selbst für Nutzungsumgebungen, die vermutlich nie oder sehr selten benötigt werden, muss der Aufwand der Migration des jeweiligen Emulators eingeplant werden.
- Emulation. Wie für Primärobjekte auch, lässt sich eine gegebene Ablaufumgebung durch Emulation nachbilden. Dieses wurde inklusive der möglichen auftretenden Probleme in Abschnitt 4.7 ausgeführt. Diese Strategie hält den Update-Aufwand der im Archiv vorgehaltenen Emulatoren gering, da sie nicht an jeweils neue Host-Plattformen angepasst werden müssen. Jedoch sollte für die Betrachtung oder Ausführung des Objekts seitens des Archivnutzers mit teilweise deutlich höherem Aufwand gerechnet werden. So kann mit höherem Anpassungsaufwand für die Übersetzung der Benutzereingaben gerechnet werden, wenn diese durch mehrere Emulationsschichten hindurchgereicht werden müssen (Abb. 6.15). Ebenso fallen für den Transport der Primärobjekte zusätzliche Arbeitsschritte an, da er rekursiv auf jeder Ebene vollzogen werden muss. Dieses Verfahren bietet sich daher eher für sehr selten nachgefragte Nutzungsumgebungen an.

Für eine Qualitätssicherung des Übergangs sollte je nach Gedächtnisorganisation eine gewisse Testsuite vereinbart sein, die eine abstrakte Überprüfung erlaubt, ohne jedes Mal aufwändige Detailtests mit jedem Objekttyp vornehmen zu müssen. Handelt es sich bei einem Emulator



**Abbildung 6.15:** Die Ineinanderschachtelung mehrerer Emulatoren kann Probleme mit der Benutzerinteraktion durch mehrfache Übersetzung von Benutzereingaben generieren. Zudem erhöht sich der Aufwand des Datentransports, da dieser in mehreren Schritten durchgeführt werden muss.

um proprietäre Software, deren Quelltext nicht offen liegt oder verfügbar ist, lässt sich der Migrationsansatz nicht verfolgen. Ebenso könnte es sein, dass zwar der Quelltext vorliegt, aber der Anpassungsaufwand an aktuelle Umgebungen so hoch wäre, dass Aufwand und

Ergebnis nicht in sinnvoller Relation stehen. Für diese Fälle bietet sich der rekursive Ansatz der ineinandergeschachtelten Emulatoren (Abb. 6.14, 6.15) an.

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 Gedächtnisorganisationen oder Anwender geleistet werden muss.<sup>37</sup> So bleibt ein Abwägen zwischen Transportaufwand, Interaktion und Zahl der Rekursionen ein zentrales Entscheidungskriterium.

### 6.7.1 Archivtypen

Je nach gefordertem View-Path werden weitere digitale Objekte benötigt. Wegen des, je nach Objekttyp, relativ hohen Aufwands wird man Überlegungen anstellen, wie man den View-Path zusammensetzt. Gerade für häufiger angefragte Pfade ist es denkbar, mit vorbereiteten Umgebungen zu arbeiten, wie sie im vorangegangenen Abschnitt 6.6 beleuchtet wurden.

Dadurch 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 AIP 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 aktuellen Dateisystem einer Referenzumgebung, vorzuhalten. Das kann einerseits das Langzeitarchiv entlasten und andererseits zu einem beschleunigten Ablauf der View-Path-Erstellung führen.
- Anlage eines View-Path-Caches - Für oft nachgefragte und aufwändiger zu erstellende View-Paths kann die Vorhaltung vorbereiteter Nutzungsumgebungen den Aufwand für Nutzer und Archivbetreiber reduzieren. Diese Caches könnten als Teil des Online-Archivs oder direkt auf der Referenzplattform abgelegt sein.

Die Betrachtungen für ein Softwarearchiv von Sekundärobjekten nimmt Abschnitt 7.8 des Folgekapitels vor.

### 6.7.2 Nutzergruppen

Weitere Anforderungen ergeben sich aus der Sicht der Institution oder des Einzelnutzers:

---

<sup>37</sup>Dieser Arbeitsaufwand kann unter bestimmten Voraussetzungen durch zukünftige Entwicklungen reduziert werden, wie sie am Abschluss des letzten Kapitels diskutiert werden.

**Archive** (siehe Abschnitt 4.5.1) haben sehr dauerhafte Ansprüche an Zugriff und Erhaltung ihrer Objekte. Typischerweise besteht keine zeitliche Befristung der Aufbewahrung.

Wegen der beschränkten Anzahl von Objekttypen wird eine standardisierte Referenzumgebung ausreichen, die das Spektrum aller zur Erstellung der Objekte benötigten Applikationen oder auch nur die Viewer abdeckt. Aus heutiger Sicht könnte beispielsweise eine Referenzumgebung bestehend aus einem Windows XP<sup>38</sup> mit allen wichtigen Applikationen zur Überprüfung und Darstellung genügen. Zudem sollte kein Bedarf zum Anschluss spezieller Peripheriegeräte zum Auslesen von Datenträgern bestehen. Alle Datenträger wie Disketten, CDs, DVDs sollten als Abbilddatei<sup>39</sup> oder aufgeteilt in einzelne Pakete in das Archivsystem übernommen worden sein. Sollte eine Archivierung als Image erfolgt sein, so sollte das eingesetzte Emulationswerkzeug in der Lage sein, auf diese Images zuzugreifen.

Begründet durch ihren gesetzlichen Auftrag sollten Archive über entsprechend ausgebildetes Personal verfügen, welches auch mit komplexeren Aufgaben eines Emulationssystems umgehen kann und in der Lage ist, einzelne komplexere Arbeitsschritte auszuführen. Dieses könnte sich in neuen Anforderungen an die Ausbildung von Archivaren niederschlagen.

**Bibliotheken** besitzen nur bedingten Einfluss auf erstellte und eingelieferte Formate. Sie sind angehalten, die meisten digitalen Objekte für den permanenten Zugriff seitens ihrer Nutzer bereit zu halten. Sie müssen daher ihren oft wenig technisch 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.

Dabei besteht nicht primär ein technisches Interesse an alten Rechnerplattformen an sich. Das kann bedeuten, dass beispielsweise nicht die Macintosh PowerPC Architektur bereitgestellt werden muss, wenn sich alle Objekte in einer bereits verfügbar gemachten Ablaufumgebung, wie beispielsweise Windows 98 auf einer X86-Emulation, darstellen lassen. Es geht also in erster Linie um die geeignete Sammlung von Applikationen und eine möglichst kompakte Ablaufumgebung für diese. Es kann jedoch vorkommen, dass verschiedene Versionen eines Formates andere Versionen der Erstellungapplikation erfordern, die ihrerseits unterschiedliche Ablaufumgebungen benötigen, weil sie nicht parallel in einer Installation funktionieren.

**Wissenschaft** Der Bereich Wissenschaft und Forschung generiert oft sehr spezielle Objekte, wie Primärdaten. Diese stammen aus verschiedenen Quellen, wie Messinstrumenten, Sensoren oder von Erhebungen. Oft werden zur Auswertung eigens erstellte Applikationen eingesetzt. Deshalb ist mit einer großen Bandbreite verschiedener Objekttypen zu rechnen, die sicherlich mehrere verschiedene Referenzplattformen erfordern werden.

**Computer-, Kunst- und technische Museen** Für diese Institutionen besteht der Bedarf, eine möglichst große Breite von Objekttypen abdecken zu können. Es kommt auf verschiedenste Betriebssysteme und Programme an, die demonstriert werden können. Da dieses in den seltensten Fällen, besonders wenn Besucherinteraktion gewünscht ist, auf der Original-

<sup>38</sup>[van Diessen und Steenbakkens 2002] geht noch von einem Windows 2000 als Referenzplattform aus.

<sup>39</sup>Siehe hierzu Abschnitt 7.10.3.



hardware stattfinden sollte, sind Emulatoren, eingesetzt auf einer aktuellen Hardware, ein guter Kompromiss.

Wenn Computer in vielfältiger Form präsentiert werden sollen, werden entsprechend passende Emulatoren für die jeweils gezeigte Plattform benötigt. Soweit möglich, wird dabei eine gewisse Vollständigkeit, beispielsweise der gezeigten Betriebssysteme und Benutzeroberflächen angestrebt. So wird man die Anfänge grafischer Benutzeroberflächen von Apple und Amiga genauso zeigen wollen wie die Genese dieser Oberflächen auf dem PC mit der ganzen Breite der in einer Periode verfügbaren Betriebssysteme.

**Unternehmen und Organisationen** haben häufig nicht die strengen zeitlichen Auflagen wie Archive. Während vermutlich alle für das Tagesgeschäft wichtigen Objekte in aktuell zugreifbare Formen im Laufe der Zeit migriert werden, so greift auf die Archivbestände mit hoher Wahrscheinlichkeit nur ausgewähltes und besonders geschultes Personal zu.

**Privatpersonen** werden in den wenigsten Fällen ein eigenes Langzeitarchiv betreiben. Jedoch wird im Laufe der Zeit mit einem Bedarf an diesbezüglichen Dienstleistungen zu rechnen sein. So könnte man sich vorstellen, dass kommerzielle Anbieter wie große Internet-Firmen diesen Bereich abdecken könnten und daraus einen eigenen Dienst ableiten.<sup>40</sup> Die Vertrauenswürdigkeit dieser Angebote hinge unter anderem von den realisierten Langzeitarchivierungskonzepten ab, die über eine reine sichere Speicherung der Bitströme hinausgehen müsste. Zumindest für eine Reihe populärer Datenformate wäre ein solches Angebot wegen der steigenden Anzahl nur-digitaler Objekte auch ökonomisch attraktiv. An dieser Stelle wäre zu fragen, welchen Sinn das Angebot eines Medienarchivs hätte, wenn die Haltbarkeit von abgelegten Fotos oder Videos weit geringer ausfällt als die ihrer analogen Pendants.

### 6.7.3 Langfristige Perspektive

Für eine Reihe historischer Computerplattformen ist die technologische Entwicklung zum Stillstand gekommen. Für diese ergeben sich keine neuen Anforderungen mehr an die eingesetzten Emulatoren. Anders stellt sich die Problematik für noch in Verwendung befindliche Architekturen dar: So produziert die permanente Weiterentwicklung des X86-PC regelmäßig neue Herausforderungen an die Programmierer von geeigneten Emulatoren und Virtualisierern. Zwangsläufig entstehen zunehmende Abstände zwischen der Weiterentwicklung der Hardware und der Treiberausstattung alter Betriebssysteme. Eine Idee hiervon können die Abbildungen 7.7 und 7.8 am Beispiel der Netzwerkkarten geben.

Ein ähnliches Problem stellt sich im Bereich der X86-Grafikausgabe: Lange Zeit dominierte der VESA-Standard für die klassische 2D-Ausgabe. Solange ein Standard allgemein anerkannt ist und verwendet wird, erleichtert das die Verwendung alter Nutzungsumgebungen im Emulator. Eine neue Herausforderung kommt durch den zunehmenden Einsatz von 3D-Techniken im Bereich der grafischen Ausgaben. Was lange Zeit eher dem professionellen

<sup>40</sup>So bieten jetzt schon diverse Internet-Provider und Unternehmen wie Google, Microsoft oder Apple Online-Speicherkapazitäten an, wo ihre Benutzer typischerweise Office- oder Medienobjekte hinterlegen können.

Entwurfs- und später dem Spielektor vorbehalten war, dehnt sich nun auf den grafischen Desktop der aktuellen Rechnerplattformen aus.<sup>41</sup>

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. So arbeitet der Hypervisor XEN mit solchen Treibern, um die Softwareschichten zwischen Betriebssystem auf der virtuellen Hardware und dem umliegenden Host-System schmal zu halten. An diese Stelle könnten sogar gesetzliche Verpflichtungen treten, 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. Wie in Abschnitt 2.6 dargestellt, existieren bereits einige erfolversprechender Initiativen, die mit kooperativen Ansätzen arbeiten. Für Zusammenschlüsse wie Nedlib, InterPARES<sup>42</sup> sollte es damit kein (finanzielles) Problem sein, solche Herausforderungen anzunehmen. Kommen bei den angestrebten Entwicklungen offene Standards und Werkzeuge, wie PRONOM oder Dioscuri heraus, können sich einerseits die Belastungen der Einzelinstitution in Grenzen halten und andererseits verbindliche Verfahren entwickeln, die von einer breiten Community unterstützt werden.

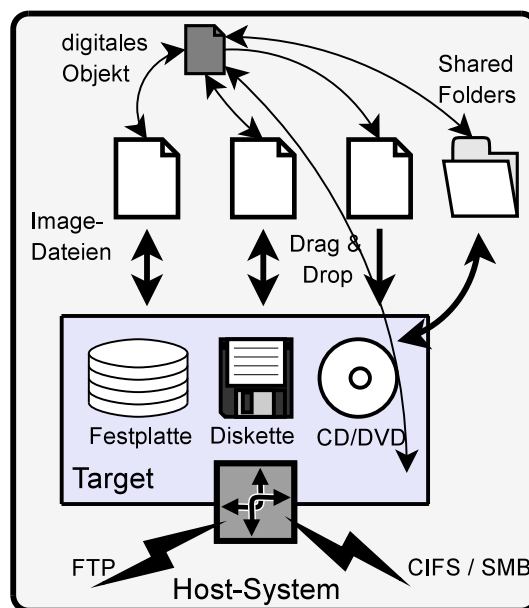
## 6.8 Wege des Datenaustauschs

Nach den generellen Überlegungen zu View-Paths stehen Untersuchungen für einige praktische Implikationen aus: Emulatoren haben die Aufgabe, die technologische Differenz zwischen der jeweils modernen Referenzplattform des jeweiligen Anwenders und der Entstehungsumgebung des digitalen Objekts zu überbrücken. Während die beteiligte Software scheinbar auf einer einzigen realen Maschine läuft, ist es gerade die Aufgabe von Emulatoren oder Virtualisierern mehrere Maschinen in einem Host-System zusammenzufassen und gegeneinander abzugrenzen. Das Datentransportproblem unterscheidet sich deshalb oft nicht von dem, welches beim Einsatz getrennter physischer Hardware auftreten würde.

Typischerweise interagieren Emulatoren für den Datenträgerzugriff nicht direkt mit der Peripherie des Host-Systems, wie Festplatten, optischen oder Diskettenlaufwerken. Stattdessen verwenden sie hierzu virtuelle Hardware. Über diese muss für einen geeigneten Transport von digitalen Objekten gesorgt werden, da der Benutzer diese normalerweise über die Host-Plattform in die gewünschte Nutzungsumgebung einbringen wird. In einigen Fällen wird auch der umgekehrte Weg benötigt: Der Transport von Primärobjekten aus ihrer Ablaufumgebung in das jeweils gültige Host-System. Hierzu stehen eine Reihe von Varianten zur Verfügung (Abb. 6.16), die jedoch von der jeweiligen Referenzplattform und ihrer technischen Ausstattung abhängen.

<sup>41</sup>Hier sind bereits erste Ansätze zur Realisierung zu beobachten, wie im VMware Fusion (Abschnitt 5.3.10) oder QEMU zeigen.

<sup>42</sup>Vgl. <http://www.interpares.org> und <http://nedlib.kb.nl>.



**Abbildung 6.16:** Je nach Host- und Gastsystem stehen verschiedene Varianten des Datentransports zur Verfügung.

Es lassen sich drei grundlegende Klassen von Austauschverfahren unterscheiden:

- Der klassische Weg ohne Eingriffe in die Nutzungsumgebung und daher am universellsten einsetzbar sind wohl Disk-Images. Bei diesen handelt es sich um spezielle Dateien, welche die komplette Struktur und den vollständigen Inhalt von Datenspeichern der verschiedensten Art, wie Disketten, CD-ROMs, DVDs oder Festplatten enthalten. Der wesentliche Vorteil von Images ist der Erhalt der logischen Struktur des vormaligen Datenträgers bei Überführung in eine gut archivierbare Repräsentation (siehe hierzu den Abschnitt 7.10 zu Installationsdatenträgern am Ende des nächsten Kapitels).
- Im Laufe der technologischen Entwicklung begannen sich Computernetzwerke durchzusetzen. Betriebssysteme, die solche unterstützen, erlauben mittels virtueller Netzwerkverbindungen digitale Objekte zwischen Referenz- und Nutzungsumgebung auszutauschen.
- Neben diesen Verfahren bieten einige Virtualisierer und Emulatoren spezielle Wege des Datenaustauschs.

Die notwendigen Arbeitsabläufe zum Datenaustausch können vom Erfahrungshorizont der jeweiligen Nutzer abweichen und sind deshalb geeignet (in den Metadaten) zu dokumentieren oder zu automatisieren.<sup>43</sup>

### 6.8.1 Netzwerkprotokolle

Neuere Betriebssysteme verfügen über die Fähigkeit, Daten über Netzwerke auszutauschen. Viele Emulatoren enthalten deshalb virtuelle Netzwerkhardware und können zwischen Host-

<sup>43</sup>Siehe hierzu die Überlegungen und Umsetzungen in [Welte 2008].

und Nutzungsumgebung virtuelle Datenverbindungen einrichten. Während Ende der 1980er Jahre noch proprietäre Protokolle wie Appletalk, NetBIOS und IPX/SPX neben TCP/IP existierten, hat sich letzteres inzwischen weitgehend durchgesetzt und die anderen verdrängt. Hierbei ist bemerkenswert, dass der jetzige IPv4-Standard, welcher in seiner ersten Fassung am Ende der 1970er Jahre festgelegt wurde, im Laufe der Zeit erstaunlich stabil geblieben ist. Gleichzeitig sieht man hier ein Beispiel und den Vorteil freier, offener und langlebiger Standards.

Für den Austausch von digitalen Objekten sind in erster Linie Protokolle für den Datentransport interessant, die bereits seit vielen Jahren standardisiert sind. Hier stechen besonders das seit den 1980er Jahren definierte File Transfer Protocol (FTP) und das Server Message Block Protocol (SMB) hervor.

Für den Aufbau von Netzwerkverbindungen sind im Host-System privilegierte Zugriffe auf Systemschnittstellen notwendig. Normalerweise können Nutzerprozesse – als solche laufen üblicherweise Emulatoren – nicht selbstständig Datenpakete auf die verschiedenen Netzwerkebenen, wie Data Link, Network oder Transport Layer einschleusen. Hierzu müssen im Host-System Schnittstellen existieren oder geschaffen werden, die ein kontrolliertes Einschleusen von Datenpaketen erlauben. Virtualisierer sprechen hierzu klassischerweise mit einem Kernel-Modul des Host-Systems, welches über diese Privilegien verfügt. In den meisten Fällen muss eine zusätzliche Spezialsoftware, wie TUN/TAP<sup>44</sup> installiert werden.

TCP/IP würde zwar den Datenaustausch auch über Weitverkehrsverbindungen erlauben, dieses sollte man sinnvollerweise vermeiden und nur in geschützten, nicht mit der Aussenwelt verbundenen Subnetzen realisieren, da der Sicherheitsstandard der verschiedenen Gastsysteme bei weitem nicht mehr aktuellen Anforderungen entspricht.

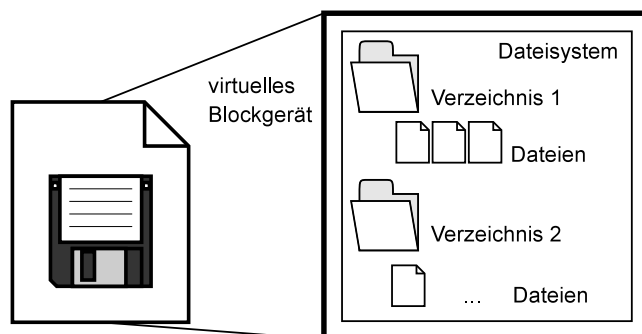
Verglichen mit dem Datentransport über verschiedene Typen blockorientierter Speichermedien, der in den folgenden Abschnitten erläutert wird, ist die Einrichtung von Datenverbindungen deutlich aufwändiger und auf Nutzungsumgebungen mit Netzwerkfähigkeit beschränkt. Jedoch sind die späteren Arbeitsabläufe einfacher. Insgesamt arbeitet diese Art des Datenaustauschs signifikant flexibler, da quasi beliebige Datenmengen in beide Richtungen bei laufender Nutzungsumgebung transportiert werden können.

### 6.8.2 Disketten-Images

Diskettenlaufwerke gehören zur Gruppe der ältesten Datenträger<sup>45</sup> populärer Computersysteme seit den 1970er Jahren. Der überwiegende Anteil der Emulatoren und Virtualisierer ist geeignet, mit virtuellen Diskettenlaufwerken unterschiedlicher Kapazitäten umzugehen. Das Format der virtuellen Disketten ist bei allen identisch, so dass nicht für jeden Emulator ein eigenes Image abgelegt werden muss und ein einfacher Austausch erfolgen kann. Die dahinter stehende einfache Technik sollte es zudem erlauben, sie auch in fernerer Zukunft zu emulieren. Die physikalische Struktur der Diskette wird durch eine logische Blockstruktur in einer Datei ersetzt, indem die Steuerhardware des Laufwerkes und der physische Datenträger weggelassen werden. Aus Sicht des Betriebssystems im Emulator ändert sich nichts. Diskettenzugriffe werden einfach durch Zugriffe auf die Abbilddatei im Host-System umgesetzt.

<sup>44</sup>Hierbei handelt es sich um spezielle Netzwerk-Interfaces, die in Software Ethernet- oder IP-Pakete aufnehmen und an den Stack des Host-Systems weiterreichen können.

<sup>45</sup>Siehe dazu den Abschnitt A.5.1 im Anhang.



**Abbildung 6.17:** Logischer Aufbau einer Datei, die ein Disketten-Image repräsentiert. Die logische Blockstruktur kann mit einem Dateisystem versehen sein, welches Verzeichnisse und Dateien enthält.

Wegen des einfachen Aufbaus einer solchen Datei, gibt es in den meisten Host-Systemen die Möglichkeit den Inhalt als virtuelles Blockgerät mit Dateisystem einzuhängen. Dadurch kann der Container so einfach wie eine normale Festplatte gelesen und beschrieben werden, wenn das Host-System das Dateisystem im Container unterstützt.<sup>46</sup> Für den Zugriff auf Disketten-Images in einer Referenzumgebung sind unter Umständen spezielle Hilfsprogramme notwendig, die zusätzlich archiviert werden sollten.

Disketten-Images können entweder beim Start des Emulators bereits verbunden sein. Fast alle Emulatoren unterstützen zudem das Ein- und Aushängen zur Laufzeit, welches normalerweise vom Benutzer getriggert werden muss. Dieser sollte zudem darauf achten, dass eine Image-Datei nicht mehrfach eingebunden ist, da innerhalb des Containers Blockoperationen ausgeführt werden und kein File Locking stattfindet. Beispielexperimente zur Erzeugung sowie Nutzung von Disketten-Images, hier unter dem Gesichtspunkt des Installationsdatenträgers, sind im folgenden Kapitel im Abschnitt 7.10.4 dokumentiert und ausführlicher erörtert.

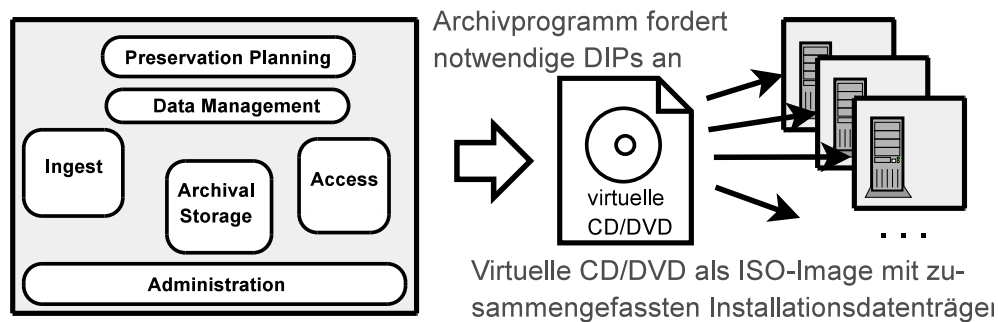
### 6.8.3 Virtuelle optische Wechseldatenträger

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

CD-ROMs als Datenträger für den Einsatz im Computer gibt es erst seit Mitte der 1990er Jahre. Während Disketten quasi nativ in den meisten Plattformen unterstützt sind, muss für die Verwendung von CD-ROMs üblicherweise noch ein Treiber geladen werden.<sup>47</sup> Mit der Verfügbarkeit beschreibbarer optischer Datenträger kamen Programme zur Generierung von

<sup>46</sup>Beispielsweise gelingt das Einbinden eines DOS-, VFAT- oder HFS-formatierten Disketten-Images in Linux-Systemen ohne Schwierigkeiten, solange diese Dateisysteme und das spezielle Loop Blockdevice vom Kernel unterstützt werden. In Zukunft könnten langzeitrelevante Dateisysteme mittels FUSE (Filesystem im Userspace - <http://fuse.sourceforge.net>) realisiert und gepflegt werden.

<sup>47</sup>In neueren Betriebssystemen ist dieser Treiber üblicherweise enthalten. Mit der Treiberproblematik allgemein beschäftigt sich Abschnitt 7.5 im anschließenden Kapitel.



**Abbildung 6.18:** CDs oder DVDs sind die Standard-Installationsdatenträger schlechthin und werden in ihrer virtuellen Ausgabe von jedem gängigen Emulator unterstützt. Das lässt sich gut für die Vereinfachung von Abläufen ausnutzen.

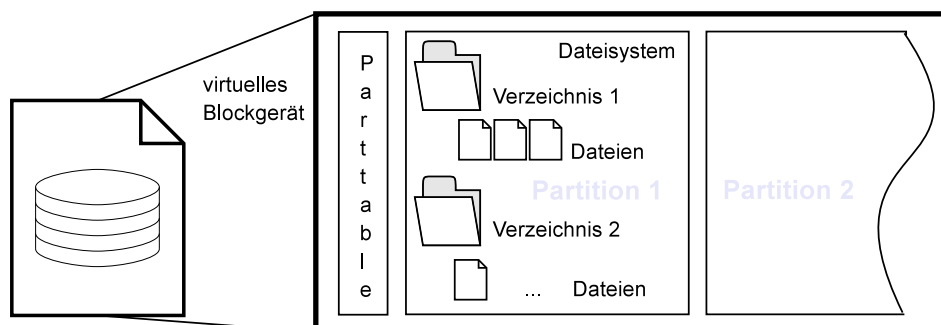
ISO-Images auf. Inzwischen sind sie für jedes gängige Betriebssystem und damit für die relevanten Referenzplattformen verfügbar.

Die Erstellung eines ISO-Images im Hostsystem für den Datentransport in virtualisierte oder emulierte X86-Umgebungen demonstriert Abschnitt 7.10.5.

#### 6.8.4 Virtuelle Festplatten

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

Seit ungefähr 20 Jahren haben sich blockorientierte Festspeichergeräte als dauerhafter Speicher durchgesetzt. Der Inhalt dieser Speichermedien wird als virtuelles Blockgerät in Form von Dateien im Host-System repräsentiert. Ist der Aufbau dieser Dateien bekannt, so können diese ohne laufenden Emulator erzeugt und verändert werden. Wegen des komplexeren inneren



**Abbildung 6.19:** Logischer Aufbau einer Datei, die ein Festplatten-Image enthält. Gegenüber Disketten ist bei Festplatten eine Partitionierung, eine logische Aufteilung der Gesamtkapazität, durchaus üblich.

Aufbaus von Containerdateien wird ein einfaches Mounten im Referenzsystem fehlschlagen, wie es für Disketten- und ISO-Images vorgenommen werden kann. Stattdessen sind spezielle

Programme, wie "vmware-mount.pl" oder "lomount" erforderlich.<sup>48</sup> Der Zugriff unabhängig vom laufenden Emulator oder Virtualisierer erlaubt:

- Automatische Vorbereitung einer Gastplattform ohne Benutzer- oder Administratorzugriff. Das später laufende Gastbetriebssystem gibt die Formatierung des virtuellen Blockgerätes mit einem bestimmten Dateisystem vor, da es später auf dieses zugreifen können muss.
- Bidirektionaler Transport von Primärobjekten in die Gastumgebung hinein und der Ergebnisse aus der Gastumgebung heraus.
- Erstellung von Abzügen laufender Systeme für die Langzeitarchivierung (Abschnitt 6.6.1).

Viele der relevanten Plattformen können mit mehr als einer Festplatte umgehen, so auch ihre Emulatoren und Virtualisierer. Deshalb könnte eine Aufteilung in zwei virtuelle Datenträger vorgenommen werden: Einrichtung einer Systemplatte mit dem Betriebssystem und allen notwendigen Komponenten zur Betrachtung oder Ausführung eines Primärobjekts. Erstellung einer Datenplatte ohne oder mit sehr einfacher Partitionierung und einem Dateisystem, welches sowohl vom Host- als auch vom Betriebssystem der Nutzungsumgebung schreibend in Zugriff genommen werden kann.

### 6.8.5 Spezielle Datentransportpfade

Moderne Virtualisierer bieten neben den klassischen Verfahren des Datenaustauschs zwischen zwei unterschiedlichen oder entfernten Computern zusätzliche Wege an, die durch die Ausführung auf einer gemeinsamen Maschine möglich werden.

Die neueren VMware-Programme, wie die Workstation ab Version 4 und der Player offerieren sogenannte "Shared Folders", die sich bestimmten Gastsystemen als Netzwerkdatenträger präsentieren und analog angesprochen werden können. So können Daten einfach zwischen Host- und Gastsystem hin- und herkopiert werden. Diese Variante setzt jedoch eine relative "technologische Nähe" zwischen Host- und Gastsystem voraus.

Die Variante grafischer Benutzeroberflächen durch "Drag&Drop" Daten zwischen Applikationen auszutauschen, hat zuerst Microsoft für Virtual PC eingeführt. Das bedeutet, dass die virtuelle Maschine aus Sicht des Hostsystems als Applikation arbeitet, die ebenso wie andere Programme der Plattform über bestimmte Betriebssystemschnittstellen Daten empfangen und senden kann. Hier gilt ebenso wie bei den "Shared Folders", dass dieses einen nicht zu großen Abstand von Host- und Gastsystem voraussetzt.

Der Vorteil beider Methoden liegt darin, dass ein Benutzer ohne hohen Aufwand seine Daten in eine virtualisierte Umgebung bringen, dort betrachten und anschließend eventuell transformierte Daten wieder entnehmen kann.

Der gewählte Emulationsansatz geht davon aus Host-Systeme und Nutzungsumgebungen weitgehend voneinander getrennt zu halten, um auf diese Weise den technologischen Abstand überbrücken zu können. Damit stehen jedoch nicht die von modernen Betriebssystemen bekannten Arten des Datentransports in einfacher Weise zur Verfügung.

<sup>48</sup>Vgl. hierzu die Abschnitte 5.3.1 und respektive 5.6.1.

Die Varianten des Datenaustausches mit der Umwelt, das heißt über die jeweiligen Systemgrenzen hinweg, hängen von der Rechnerarchitektur ab. So kannten sehr frühe Plattformen als Austauschmedien nur Magnetbänder, Lochstreifen oder -streifen. Später kamen Disketten, Festplatten und optische Datenträger hinzu. Viele Systeme erlaubten zudem einen Datenaustausch über Computernetzwerke hinweg. Diese setzen jedoch immer eine geeignete Gegenstelle zum Empfang oder Senden von Datenströmen oder -paketen voraus.



## 7 Softwarearchiv der Sekundärobjekte

Nach den ausführlichen Betrachtungen zu Langzeitarchivierungsstrategien digitaler Objekte, der Behandlung der Frage der Auswahl geeigneter Emulatoren und der Bedeutung von View-Paths für den späteren Objektzugriff soll es in diesem Kapitel um die Beschreibung eines ergänzenden Softwarearchivs gehen. Zu den Erfolgsbedingungen der Emulationsstrategie auf der Ebene der Hardwarearchitekturen zum Erhalt möglichst originalgetreuer Ablaufumgebungen für die verschiedensten Typen digitaler Objekte zählen nicht nur die Primärwerkzeuge. Neben den eigentlichen Emulatoren ist eine ganze Reihe weiterer Softwarekomponenten erforderlich.

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

- Ein *Primärobjekt* oder *Zielobjekt* gehört zu den primär interessierenden Objekten eines Archivbenutzers, welches er in irgendeiner Form betrachten will.
- Für die Betrachtung wird Software (zur Herstellung einer Nutzungsumgebung) benötigt. Die notwendigen Komponenten ergeben sich aus dem jeweiligen View-Path und können je nach dessen Wahl variieren. Diese Software in den verschiedenen Ausformungen wird als *Sekundär-* oder *Hilfsobjekt* bezeichnet. Dieses ist für den Benutzer nicht primär von Interesse, wird aber gebraucht, um überhaupt mit dem Zielobjekt umgehen zu können. Je nach Art des Zielobjekts können einige Intermediärobjekte notwendig sein.

Ein Abschnitt widmet sich deshalb der notwendigen Sekundärsoftware, ein weiterer nimmt Überlegungen zur Verteilung und zum Archivmanagement vor. 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 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. Eine besondere Rolle spielen Treiber virtueller Hardware.

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

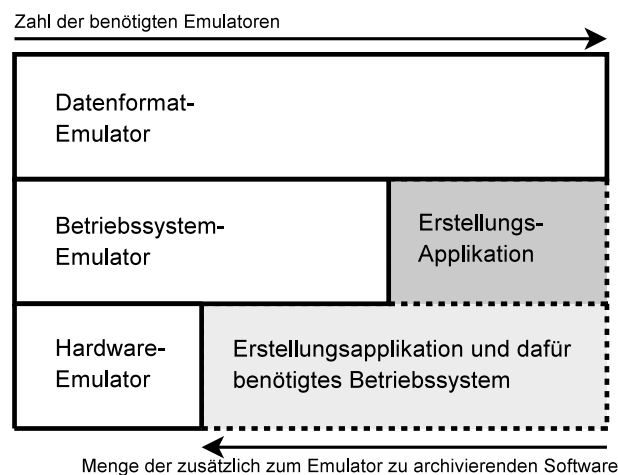
Pendants, häufig mit einschränkenden Nutzungsrechten versehen. Ein weiterer Abschnitt beschäftigt sich mit der Verwaltung von Zugriffsrechten und macht Vorschläge zu ihrer Realisierung.

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 betrachtet der abschließende Abschnitt virtuelle Datenträger und mögliche Formen der Ablage von Sekundärobjekten.

## 7.1 View-Paths und benötigte Software

Je nach gewählter Ebene der Emulation<sup>1</sup> 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-Stack (Abb. 7.1).<sup>2</sup>

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 wohldokumentierten Bildformate.<sup>3</sup> Die Emulation eines



**Abbildung 7.1:** Zahl der benötigten Emulatoren je nach gewählter Ebene und Menge der zusätzlich zu archivierenden Software.

Betriebssystems oder dessen Schnittstellen erlaubt theoretisch unter den im Abschnitt 3.5.3 genannten 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

<sup>1</sup>Vgl. hierzu Abschnitt 3.5.1, speziell Abbildung 3.9 im Überblicksteil der Arbeit.

<sup>2</sup>Vgl. zudem die Diskussion eines "Software Repository" in [Reichherzer und Brown 2006].

<sup>3</sup>Vgl. hierzu Abschnitt 3.6, insbesondere Abbildung 3.10. Jedoch ist die Menge potenziell vorzunehmender Migrationen zu bedenken (Abb. 3.8).

des Betriebssystememulators muss darauf geachtet werden, dass sämtliche im Softwarearchiv eingestellten Applikationen weiterhin ablaufen können.

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 zu sehen, müssen je nach Architektur die Ebenen zwischen der emulierten Hardware und dem Objekt selbst "überbrückt" werden. Die Brücke wird durch die im vorangegangenen Kapitel eingeführten View-Paths realisiert. Diese sehen je nach gewählter Strategie, "Migration-on-Dissemination" oder Wiederherstellung einer Nutzungsumgebung, unterschiedlich aus.

Generell kann ein Archivnutzer nicht auf einer nackten Maschine ein PDF-Dokument öffnen. Er braucht hierfür mindestens ein Programm zur Betrachtung, welches seinerseits nicht direkt auf der Hardware ausgeführt wird und deren Schnittstellen für den Datei- und Bildschirmzugriff, sowie die Abfrage der Benutzereingaben direkt programmiert. Dieses Programm setzt seinerseits ein Betriebssystem als Intermediär voraus, welches sich um die Ansteuerung der Ein- und Ausgabeschnittstellen der Hardware kümmert (Abb. 3.9).

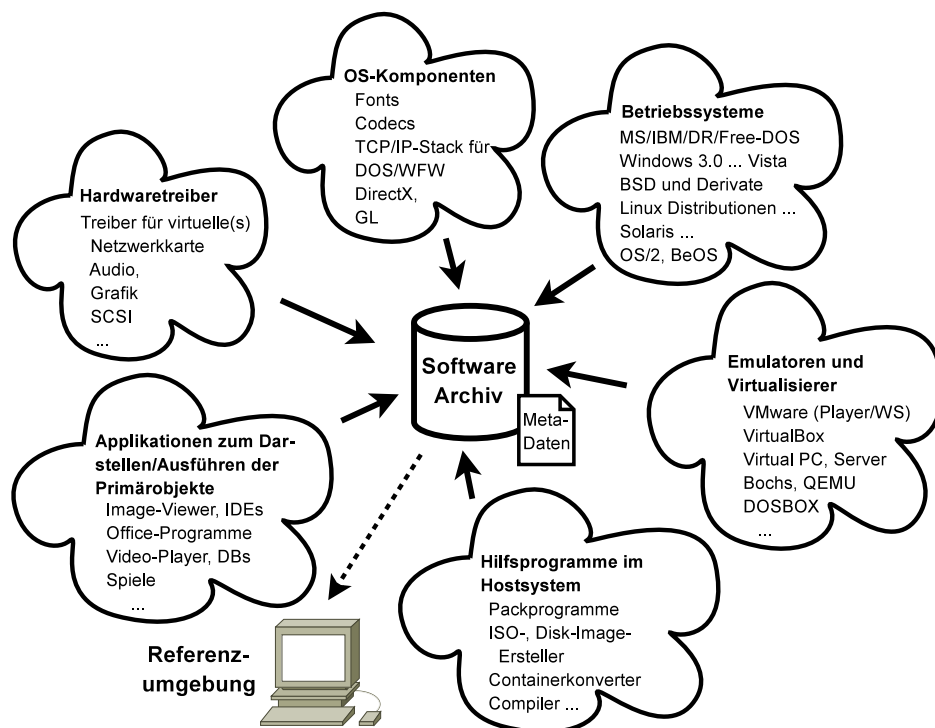
Bei Nachbildung einer kompletten Hardware durch Emulation benötigt man in jedem Fall mindestens eines oder je nach Bedarf mehrere Betriebssysteme, die sich als Grundlage der darauf aufsetzenden Applikationen ausführen lassen. Das bedeutet für ein Softwarearchiv, dass neben dem Emulator für eine Plattform 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.

Dieses Dilemma löst beispielsweise VMware momentan durch die Pflege und Bereitstellung geeigneter Treiber für jedes offiziell unterstützte Betriebssystem. Noch ist die Liste sehr lang und im Sinne der Archivierung fast vollständig. Ein weiteres Problem ist die Art der Prozessornutzung. Virtualisierte 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. Sie erlauben zum einen die Übersetzung für die jeweilige Plattform und zum anderen auch langfristige Anpassungen an völlig neue Architekturen. Zudem kann sichergestellt werden, dass auch alte Peripherie dauerhaft virtualisiert wird und nicht einem Produktzyklus zum Opfer fällt.

Ein weiterer Punkt ist die unter Umständen notwendige Anpassung der Containerdateien, in den 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.

## 7.2 Bestandteile eines Emulationssoftwarearchivs

Wie im Kapitel 6 eingeführt,<sup>4</sup> benötigt ein jedes digitales Objekt einen Betrachter, mit dem es geöffnet oder ausgeführt werden kann. Da es sich hierbei um Software handelt, ist diese auf eine bestimmte Nutzungsumgebung 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



**Abbildung 7.2:** Überblick der ins Archiv aufzunehmenden Emulatoren und weiteren Softwarekomponenten für die X86-Plattform.

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 muss deshalb eine ganze Reihe verschiedener Softwarekomponenten umfassen, so sind:

- die Emulatoren zu speichern, mit deren Hilfe die Wiederherstellung einer Rechnerarchitektur für bestimmte Nutzungsumgebungen erfolgen kann.

<sup>4</sup>Vgl. zur Illustration zudem Abbildung 3.2, sowie [Reichherzer und Brown 2006].

- die Betriebssysteme abzulegen, die je nach Rechnerplattform einen Teil der Nutzungsumgebung ausmachen.
- Treiber der Betriebssysteme zusätzlich abzulegen, da sie dem Betriebssystem überhaupt erst erlauben eine bestimmten Hardware anzusprechen, wenn kein direkter Zugriff via BIOS-API realisiert werden kann.
- die Applikationen zu archivieren, mit denen die verschiedenen digitalen Objekte erstellt wurden. Diese Applikationen sind ebenfalls Bestandteil der Nutzungsumgebung des Objekts. Sie sind in vielen Fällen auf die vorgenannten Betriebssysteme angewiesen.
- die unter Umständen notwendigen Erweiterungen einer Applikationsumgebung, wie bestimmte Funktionsbibliotheken oder Schriftartenpakete zur Darstellung zu ermitteln und bei Bedarf hinzuzufügen.
- 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 mehrere Varianten derselben Software zu archivieren, um beispielsweise die Lokalisierung in eine bestimmte natürliche Ein- und Ausgabesprache zu erreichen.

### 7.2.1 Art der Ablage

Für das Softwarearchiv ist zu entscheiden, welche Art der Ablage für die Einzelkomponenten gewählt wird. Hierbei spielen einerseits die Aspekte einer Langzeitarchivierung und andererseits die Häufigkeit des Zugriffs auf diese eine Rolle. Für eine Langzeitarchivierung ist sicherlich die Ablage der Sekundärobjekte analog zu den Primärobjekten in einem OAIS-Archiv anzustreben. An dieser Stelle könnte überlegt werden, Primär- und Sekundärobjekte miteinander zu verbinden und als gemeinsames Paket abzulegen.<sup>5</sup> Eine Zusammenstellung notwendiger Elemente für Home-Computer zeigt Abbildung 7.3.

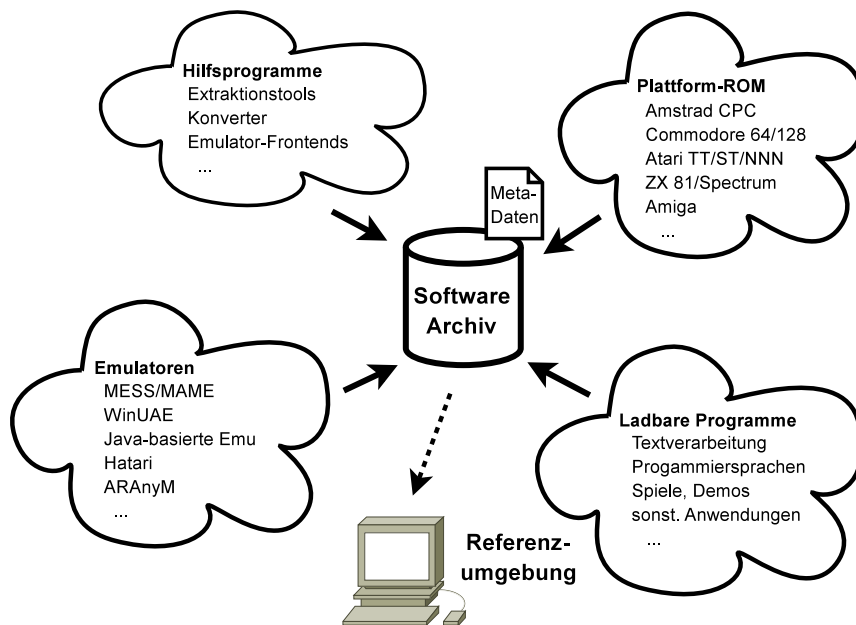
Gerade für ältere Architekturen bleibt der Umfang des Speicherbedarfs einer solchen "Collection" überschaubar. Auf diese Weise ließen sich Metadaten wenig redundant mit einer ganzen Sammlung von Primärobjekten speichern.<sup>6</sup>

Je nach Art des Archivs (Abschnitt 6.7.1) sollte eine Ablage im eventuell zwischengeschalteten Nutzungsarchiv erfolgen, um einen schnellen Zugriff zu erlauben. Ebenfalls könnte es von Interesse sein, spezielle Installationsdatenträger<sup>7</sup> vorzuhalten, um im Ablauf vereinfachte oder automatisierte Setups von Nutzungsumgebungen zu realisieren. Dieses gilt ebenso für die Ablage in vorbereiteten Nutzungsumgebungen, den View-Path-Caches (Abschnitt 6.6).

<sup>5</sup>Vgl. die Gedanken in Abschnitt 6.6 und Vorüberlegungen von [Rothenberg 1995].

<sup>6</sup>Solche Überlegungen hängen vom Sammlungszweck und Blickwinkel der Gedächtnisorganisation und ihrer Nutzer ab: So kann man den Computer ZX Spectrum als ein Objekt mit seinen typischen Applikationen sehen oder ein besonderes Spiel eines Herstellers für diese Plattform betrachten.

<sup>7</sup>Vgl. hierzu Abschnitt A.3.



**Abbildung 7.3:** Zusammenstellung der ins Archiv aufzunehmenden Softwarekomponenten für eine Langzeitarchivierung von Home-Computern. Sie sind in Anzahl und Größe deutlich überschaubar.

### 7.2.2 Softwarevarianten

Die Maschinensprache der jeweiligen Rechnerarchitektur ist einheitlich definiert und kennt keine landeslokalen Unterschiede. Ab einem bestimmten Abstraktionsgrad sind Lokalisierungen, die Anpassungen an die jeweilige Landessprache vornehmen, von Bedeutung. Dieses spielt sich einerseits im Benutzer-Interface ab - bestimmte Zeichensätze der Tastatureingabe und der Bildschirmausgabe sorgen für die Anpassung an den Sprachraum des Benutzers. Für diese Aufgaben zeichnet typischerweise das Betriebssystem verantwortlich.

Während es beim Benutzer-Interface abstrakt darum geht Lokalisierungen vornehmen zu können, muss dieses durch die jeweiligen Applikationen, die mit dem Benutzer direkt interagieren, konkret umgesetzt werden. Das betrifft die Beschriftung von Menüs ebenso wie alle Ausgaben in Form von Hinweisen. Hinzu kommen landes- oder regionstypische Formatierungen von Währungen, Zeitangaben und Maßeinheiten.

In der Computeranfangszeit waren allein aus Speicherplatz- und Zeichensatzgründen die Benutzer-Interfaces lediglich englisch. Diese wurden mit der Verbreitung der Technologie in breitere Nutzerkreise immer stärker an den jeweiligen Sprachraum angepasst. Diese Anpassungen bedeuteten einen höheren Aufwand seitens der Programmierer. Sie mussten in ihren Betriebssystemen und Applikationen dafür sorgen, dass sie Schnittstellen zur Verfügung stellten, die einen transparenten Austausch der Lokalisierung erlaubten oder ihre Software für die jeweilige Lokalisierung neu übersetzen.<sup>8</sup>

<sup>8</sup>Diesen Problemkreis hatten insbesondere kommerzielle Softwarehersteller benutzt, um eine Segmentierung ihrer Märkte zu erreichen und dadurch teilweise unterschiedliche Preisstrukturen durchzusetzen.

## 7.3 Applikationen

Applikationen oder Programme sind Software, die zur Erstellung und Verarbeitung elektronischer Daten geschrieben wurden. Sie treten dem Benutzer entweder in direkter Form über die Benutzerschnittstellen der jeweiligen Rechnerplattform entgegen oder werden mittelbar durch Benutzereingriffe aufgerufen. Sie treten damit an einer definierten Stelle des View-Paths auf. Die letztlichen Anforderungen werden durch das jeweilige Primärobjekt bestimmt.

**Sekundärobjekt** Die meisten digitalen Primärobjekte wurden mit speziellen für ihren Zweck geschaffenen Applikationen erstellt und sind vielfach ohne diese nicht sinnvoll zu erleben oder zu benutzen. In diesem Fall sind Applikationen für die View-Path-Generierung erforderliche Sekundärobjekte. Einige Datenformate können zusätzlich durch Programme von Drittanbietern im Sinne der Betrachtung ausreichend interpretiert werden. Die Güte ihrer Verwendbarkeit ließe sich durch eine Metrik beschreiben (Abb.7.4).

**Primärobjekt** Einige Applikationen sind für sich genommen von direktem Wert als digitales Primärobjekt. Dieses können einerseits Programme, wie Spiele sein, mit denen Benutzer zum Freizeitvertreib interagieren. Sie interessieren aus persönlichem Interesse einzelner Anwender oder späterer Forschung beispielsweise zum Freizeitverhalten einzelner Epochen. Andererseits werden bestimmte Rechnerplattformen aus historischem Interesse aufbewahrt, um sie beispielsweise im Rahmen von Ausstellungen in technischen Museen als Zeitzeugen ihrer Epoche zu präsentieren.

**Metadaten** Programme erklären sich nur in wenigen Fällen aus sich heraus und dann oft auch nur einem speziellen Benutzerkreis. Deshalb sind gemeinsam mit ihnen geeignete Metadaten<sup>9</sup> abzulegen. Klassische Beispiele erweiterter Metadaten sind die Handbücher oder allgemein Anleitungen, die mit einem Softwarepaket verbreitet wurden. Sie erklären Installation und Nutzung der Software, enthalten Hinweise zur Tastenbelegung oder zu zusätzlich zu installierenden Komponenten. Ebenso sind notwendige Lizenzen und je nach Art der Freischaltung oder Aktivierung Lizenzschlüssel zu hinterlegen.

### 7.3.1 Viewer und Editoren

Für eine gewisse Anzahl von meistens proprietären Datenformaten besteht normalerweise eine direkte 1:1-Verknüpfung zwischen Format und erzeugender Applikation. Daneben existiert eine Reihe von generischen, oft allgemein standardisierten Formaten, die von mehreren Applikationen angezeigt und modifiziert werden kann.

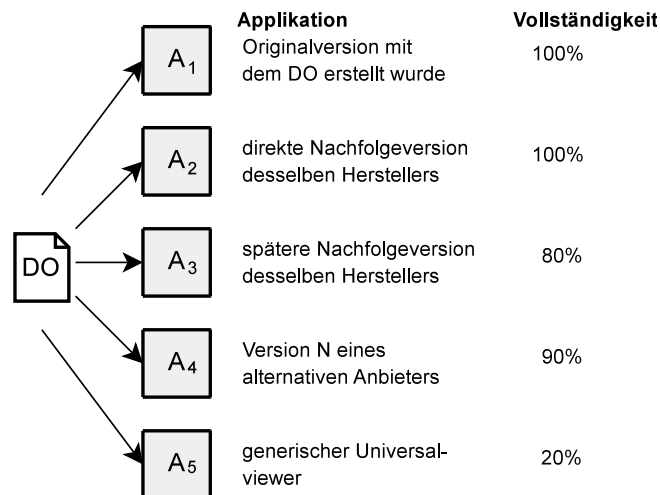
Zu den generischen Formaten zählen einfache Texte in Kodierungen, wie ASCII oder UTF8, diverse Bildformate wie PNG, JPEG oder TIFF. Für letztere gibt es etliche Universal-Viewer. Hier kann es sinnvoll sein, Programme zu wählen, die vielfältige Formate behandeln können, um zu vermeiden, eine große Anzahl von ihnen gleichzeitig archivieren zu müssen.

---

<sup>9</sup>Vgl. hierzu Abschnitt 3.3 zu Metadatenschemata digitaler Objekte und Abschnitt 4.4 zur Anwendung auf Emulatoren und Virtualisierer als spezielle Applikationen.

Diese Programme haben zudem den Vorteil gleichzeitig, zur vertikalen Migration von einem Format in ein anderes einsetzbar zu sein.

Etwas komplizierter liegt der Fall für Audio- und Video-Player. Diese Viewer sind in den meisten Fällen modular aufgebaut: Der Rahmen des Programms übernimmt die klassischen Funktionen des Öffnens der Dateien und nutzt die Schnittstellen des Betriebssystems zur Anzeige auf dem Bildschirm und Ausgabe auf der Soundkarte. Für die Dekodierung des Datenstroms und die Interpretation der Containerformate<sup>10</sup> ist meistens ein Plugin-Konzept realisiert. Auf diese Weise können neue und weitere Codecs<sup>11</sup> zur Laufzeit des Programms eingebunden werden. Diese müssen deshalb zusätzlich zum jeweiligen Viewer abgelegt werden. Durch diese Konstruktion besteht jedoch der Vorteil, dass ein Viewer an eine Betriebssystemversion angepasst werden kann, ohne die Plugins anfassen zu müssen, solange die Rechnerarchitektur nicht geändert wird. Eine große Klasse primärer digitaler Objekte sind sogenannte



**Abbildung 7.4:** Eine Menge eines bestimmten Objekttyps kann mit unterschiedlichen Viewern verschieden vollständig oder korrekt dargestellt werden. (Abstraktes Beispiel)

Office-Dokumente. Sie umfassen klassische Textverarbeitung und -satz, Tabellenkalkulation sowie Präsentationsprogramme. In dieses Gebiet fallen ebenso Desktop Publishing (DTP) und Programme zur Erstellung von grafischen Ausgaben in vektor- oder pixelorientierten Formaten. Die meisten dieser Formate sind proprietär und unterschiedlich gut dokumentiert. Da sie jedoch häufig eine vertikale Migration über verschiedene Softwareprodukte hinweg oder horizontale Migration zwischen unterschiedlichen Versionen desselben Programms erlauben, existiert oft mehr als ein Programm zur Anzeige. Um die Vervielfältigung der View-Paths einschätzen zu können, sollte man Metriken zur Bewertung der Darstellungsgüte eines gegebenen Objekttyps durch verschiedene Viewer einführen und in den Metadaten ablegen (Abb. 7.4). Solche Metriken ließen sich entweder durch eine definierte Menge von Referenzobjekten oder durch Rückkopplung mit den Archivnutzern gewinnen.

<sup>10</sup>Videomaterial besteht aus verschiedenen "Spuren" für Ton und Bild. Es wird deshalb in Containern, wie AVI oder OGG, angeliefert, die für die Verbindung und Synchronisation sorgen.

<sup>11</sup>Abkürzung aus dem Englischen für "coder und decoder", das Verfahren zum digitalen (de)kodieren von Signalen und Daten meint.



Die Liste der Applikationen ist hiermit nicht vollständig umrissen, sondern gibt lediglich populäre Beispiele. Je nach Gedächtnisorganisation oder Anwender können die Anforderungen unterschiedlich ausfallen (siehe hierzu Abschnitte 2.2 oder 4.5). Generell muss bei der Archivaufnahme einer Applikation darauf geachtet werden, dass alle notwendigen Softwarebibliotheken zur Ausführung vorhanden sind. Dieses ist bei kommerzieller Software meistens der Fall, jedoch gibt es Zusatzprodukte oder stark modularisierte Software, die von bereits installierten Komponenten in ihrer Nutzungsumgebung ausgehen. Hierbei handelt es sich oft um Bibliotheken für die Darstellung der grafischen Oberfläche oder abstrakte Ausgabeschnittstellen.

**Weitere Objekte** Eine Reihe von Applikationen zur Dokumenterstellung greift oft auf zusätzliche Ressourcen zurück. Hierzu zählen verschiedene Schriftarten, die entweder direkt im Betriebssystem allen Applikationen zur Nutzung bereitgestellt oder über einen Font-Server im LAN angeboten werden.<sup>12</sup>

### 7.3.2 Skriptsprachen und Runtime Environments

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 Rechnerplattformen ablauffähig sind. Abbildung 3.9 zeigt den Einschub von Interpretern oder virtuellen Maschinen in den typischen Software-Stack moderner Betriebssysteme. Bekanntes Beispiel für Skriptsprachen sind Perl oder Python. Eine bekannte vollständigen Betriebssystemumgebung, das sogenannte Runtime Environment inklusive virtueller CPU ist die Programmiersprache Java.

Alle genannten Beispiele stehen für verschiedene Rechnerplattformen, wie Linux/X86, Mac-OS X/PPC oder Solaris/Sparc zur Verfügung. Sie können Programme, die in den genannten Sprachen erstellt wurden, ohne Veränderung auf unterschiedlichen Plattformen ausführen. Ein ähnliches Konzept verfolgt beispielsweise die Programmiersprache Flash für WWW-Anwendungen. Bei der Einstellung entsprechender Primärobjekte muss das Softwarearchiv entsprechend ergänzt werden.

### 7.3.3 Spiele

Spiele sind eine besondere Klasse von Applikationen (Abschnitt 3.2.5). Sie stellen einerseits oft hohe Ansprüche an ihre Ablaufumgebung und arbeiteten andererseits schon früh mit verschiedenen Typen der Lizenzprüfung. In den Spielen Anfang der 1990er Jahre zählte hierzu der Blick ins Handbuch, um bestimmte Symbolkombinationen zum Start vorzunehmen oder Fragen zu beantworten.<sup>13</sup>

---

<sup>12</sup>Das MIT-X11-System erlaubte Fonts nicht nur von der lokalen Festplatte, sondern auch via TCP/IP einzubinden. So ließ sich sicherstellen, dass alle Maschinen im LAN über das identische Set an Schriftarten verfügen.

<sup>13</sup>Ein typisches Beispiel sind die Lucas Arts Spiele der "Monkey Island"-Serie. Vgl. auch Abschnitt A.1.4.

Wenige Jahre später wurden LAN- und Online-Spiele populär, die über Computernetzwerke gespielt werden konnten.<sup>14</sup> Diese Spieltypen, die teilweise gar nicht mehr als für sich abgeschlossenes Spiel für den Single-Player-Modus produziert wurden, führten eine Online-Verifikation zur Lizenzprüfung ein. Sie stellen ganz neue Herausforderungen an Emulationsumgebungen.

## 7.4 Betriebssysteme

Betriebssysteme bilden für die überwiegende Anzahl von Rechnerplattformen das vermittelnde Glied zwischen den Applikationen und der Hardware. Sie stellen eine Reihe von Standardschnittstellen zum Benutzer und zu den verschiedenen Applikationen bereit. Auf älteren Plattformen, wie den diversen Home-Computern findet man kein Betriebssystem im modernen Sinn, sondern Firmware-ROMs, die Grundfunktionen zum Start des Systems und zusätzlich meistens einen BASIC-Interpreter enthalten.

Nach den Applikationen sind Betriebssysteme der nächste Schritt im View-Path. Genauso wie die auf ihre Schnittstellen zugreifenden Programme müssen sie für die jeweilige Rechnerarchitektur kompiliert sein. Es existieren etliche Betriebssysteme, die für mehr als eine Architektur übersetzt wurden:

- Linux zählt zu den Betriebssystemen, die auf der größten Anzahl verschiedener Rechnerplattformen ausgeführt werden kann. Prominent sind sicherlich X86 in der 32 und 64 bit-Variante. Daneben kann es für PowerPC, Sparc, S390, MIPS, ARM und viele weitere kompiliert werden. Die diversen BSD-Varianten sind auf einer ähnlich hohen Zahl von Plattformen ablauffähig.
- Windows-NT Version 4.0 wurde bis zum Service Pack 3 für DEC Alpha angeboten.
- MAC-OS X wurde ursprünglich für die PowerPC Architektur übersetzt und steht mit dem Wechsel der Architektur nun für X86 zur Verfügung. Jedoch nutzt es nicht die typischen BIOS-Funktionen der klassischen X86-Architektur, so dass es nur auf Apple-Systemen mit EFI gestartet werden kann.

Da nicht alle Rechnerarchitekturen ähnlich weit verbreitet sind oder waren und nicht für alle Emulatoren existieren, bleibt die Auswahl für die verschiedenen Plattformen oft eine theoretische. So gewinnt man beispielsweise nichts, wenn eine Applikation, die auf einem Linux Kernel der Version 2.X lauffähig ist und im Quellcode vorliegt, für die Sparc-Architektur übersetzt wird, wenn sie auch für X86 lauffähig ist. Dieses erhöht zwar die Redundanz in den View-Paths, treibt jedoch den Aufwand in die Höhe. Überlegungen hierzu finden sich in ausführlicherer Form in Abschnitt 6.4.1 im vorangegangenen Kapitel.

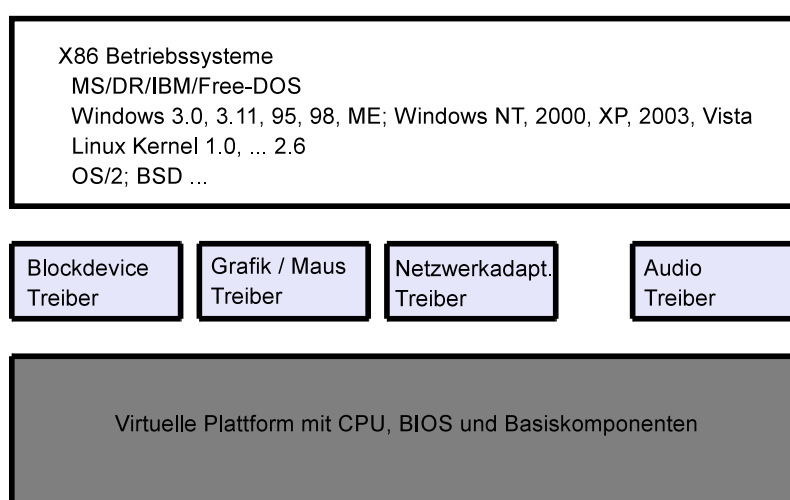
## 7.5 Hardware- und Softwaretreiber

Die Verschiedenheit von Hardware- und Betriebssystemherstellern führt dazu, dass bestimmte notwendige Softwarekomponenten nicht vom Betriebssystem mit ausgeliefert werden, obwohl

<sup>14</sup>Anfänglich nutzten die Spiele IPX/SPX, später nur noch TCP/IP für den Datenaustausch.

sie für den Betrieb einer bestimmten Hardware notwendig sind.

Dieses Problem stellt sich noch nicht für eine Reihe älterer Plattformen. So wurden beispielsweise für die Home-Computer die entsprechenden Programme direkt für eine wohldefinierte Hardware geschrieben und übersetzt. Mit dem zunehmenden Komplexitätsgrad von Software und Hardware wurden jedoch zunehmend Abstraktionsebenen, wie die vorgenannten Betriebssysteme eingezo-gen. Diese übernehmen dann im Auftrag der verschiedenen Applikationen die Kommunikation mit den Einzelkomponenten eines Rechners. Hierzu stellen sie abstrakte Schnittstellen für die Dateispeicherung oder den Zugriff auf das Netzwerk bereit (Abb. 7.5). Hier haben sich verschiedene Prinzipien herausgebildet, wie Treiber erstellt



**Abbildung 7.5:** Benötigte Hardwaretreiber zum Betrieb von X86 Betriebssystemen in realen und virtuellen Umgebungen.

werden. Diese sind in vielen Fällen proprietäre Software, deren Rechte bei den Hardwareherstellern liegen. Ebenso ist zu bedenken, dass es eher schwer ist, Treiber neuer Hardware für alte Betriebssysteme zu programmieren.

### 7.5.1 Benötigte Treiber

Während frühe X86-Betriebssysteme ihre Interaktion mit der Hardware größtenteils über BIOS-Routinen abwickelten und damit eine herstellerübergreifende einheitliche API verwendeten, benutzen modernere Betriebssysteme eigene Hardwaretreiber. Die Menge denkbarer Hardware und Peripherie wuchs derart an, so dass die BIOS-Implementierungen inzwischen nur noch für den initialen Bootvorgang benutzt werden. Für nicht im ursprünglichen BIOS vorgesehene Geräte, wie Netzwerk-Boot, SCSI- oder SATA-Controller gibt es sogenannte BIOS-Erweiterungen, die sich in die Boot-Routinen des X86-BIOS einklinken.

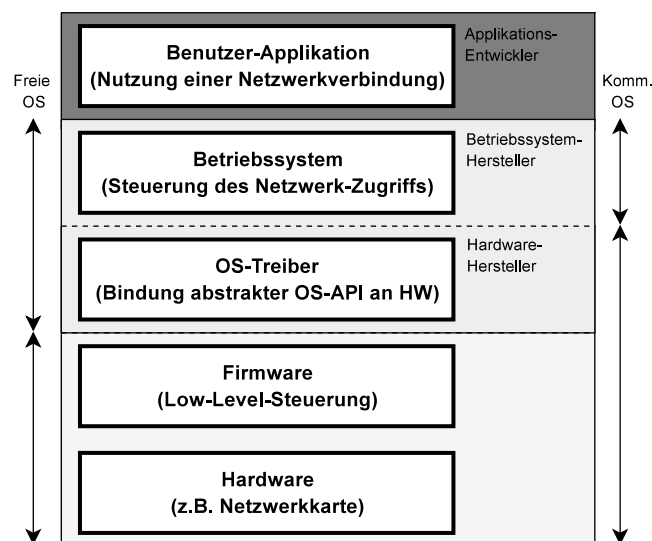
Ein Vorteil des X86-BIOS liegt in seiner seit Jahren beibehaltenen Konstanz der Schnittstellen. Deshalb gelingt es auf modernen Computern theoretisch immer noch, alte Betriebssysteme, wie DOS oder Windows 3.11 zu installieren. Diese seit mehr als 20 Jahren gültige Schnittstelle wird mit hoher Wahrscheinlichkeit durch das EFI (Extensible Firmware Interface) abgelöst werden. Dann besteht nur noch die Möglichkeit, X86-Betriebssysteme, die mit

dieser neuen API nicht mehr klarkommen, im Emulator ablaufen zu lassen oder das alte BIOS selbst zu emulieren.

Zu den Standardschnittstellen eines X86-PCs im Emulator zählen:

- IDE- oder SCSI-Festplatte und Wechselmedium
- Diskettenlaufwerk für 3,5" und 5,25" Disketten verschiedener Kapazitäten (siehe A.5.1)
- VGA-Grafikkarte unterschiedlicher Farbtiefen und Bildschirmauflösungen
- Netzwerkkarte, typischerweise Ethernet 10/100/1000 MBit/s
- Audioschnittstelle
- Serielle, parallele, PS/2 und USB-Schnittstellen

Anders als die Schnittstellen für Diskettenlaufwerke, die so standardisiert sind, dass sie von jedem Betriebssystem unterstützt werden, erfordert der Zugriff auf die Netzwerkkarte oder die Audioschnittstelle spezielle, gerätetypische Treiber. Viele Betriebssysteme sind darauf programmiert, eine generische VGA-Ausgabe zu benutzen. Zunehmend entsteht aber das Pro-



**Abbildung 7.6:** Klassische Aufgabenverteilung der verschiedenen Produzenten bzw. Programmierer bei der Treibererstellung freier und kommerzieller Betriebssysteme.

blem, dass die Bildschirmauflösung für moderne grafische Oberflächen zu gering wird und beispielsweise Dialoge von Applikationen nicht mehr vollständig auf die Anzeige passen. Ebenso können viele Betriebssysteme generisch auf Festplatten zugreifen. Diese Zugriffe sind dann aber sehr langsam und unterstützen keine angeschlossenen Wechselmedien.

**Softwaretreiber** In modernen Betriebssystemen ist es üblich geworden, bestimmte Schnittstellen zur Hardware nochmals zu abstrahieren, um Applikationen einen definierten Zugriff zu gewährleisten. Hierzu zählen höherschichtige Treiber für OpenGL oder DirectX.<sup>15</sup> Auf diese

<sup>15</sup>Vgl. <http://www.microsoft.com/windows/directx/productinfo/default.mspx>.

einheitlichen Schnittstellen greifen beispielsweise Computerspiele zu und sprechen die Hardware nicht mehr direkt über ihre primären Treiber sondern über eine Betriebssystemschicht an. Diese werden typischerweise mit dem Betriebssystem ausgeliefert, oft jedoch in anderen Zyklen als dieses aktualisiert.

### 7.5.2 Hardwaretreiber kommerzieller Betriebssysteme

Kam bei den Home-Computern der 1980er Jahre und den Apple Macintosh der 1990er Jahre<sup>16</sup> die Hardware und die Firmware beziehungsweise das Betriebssystem vom selben Hersteller, handelt es sich bei der X86-Plattform um ein großes "Kooperationsprojekt" mit vielen verschiedenen Marktteilnehmern.

Die Lage war bei Home-Computern noch relativ einfach, da es lediglich einen kleinen Umfang an Hardware gab und die Varianz innerhalb einer Plattform gering ausfiel. Die einzelnen Plattformen waren in sich deutlich geschlossener und inkompatibel zu anderen.

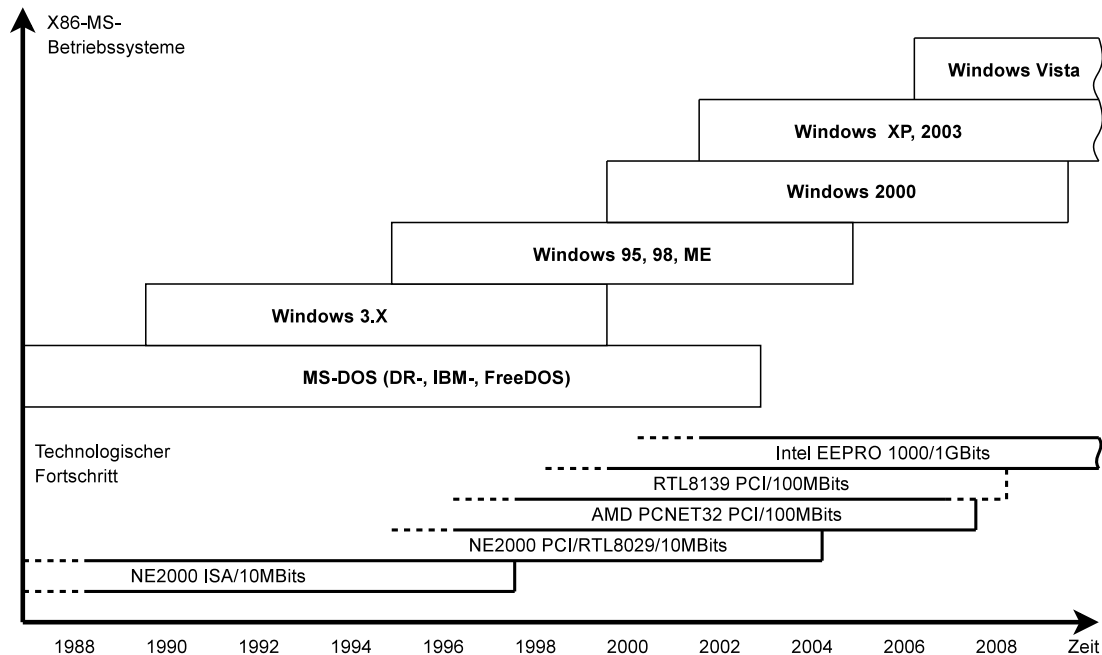
Im kommerziellen Bereich teilen sich Hardwareproduzenten und Betriebssystemhersteller üblicherweise die Aufgabe der Treibererstellung auf (Abb. 7.6). Das Betriebssystem definiert eine bestimmte API, gegen die die Treiber für die jeweilige Hardware entwickelt werden. Das bedeutet für die Hardwareproduzenten, dass sie oft mehrere Treiber für verschiedene Betriebssysteme schreiben müssen. Da dieses die Kosten in die Höhe treibt, sind Hersteller bemüht, die Treiberentwicklung auf das jeweils populäre Marktsegment zu begrenzen. Das bedeutet, dass bei Ablösung oder Abkündigung eines Betriebssystems die Hardwarehersteller aufhören, für ihre neue Hardware noch Treiber zu entwickeln. Abbildung 7.7 illustriert diese Problematik anhand der Microsoft-Betriebssysteme und ihrer Unterstützung typischer von Virtualisierern und Emulatoren nachgebildeter Netzwerkkarten.

Daraus folgt, dass sich keine Hardwaretreiber neuerer Produkte für seltene oder nicht mehr weiterentwickelte Betriebssysteme mehr finden lassen. Umgekehrt gilt ähnliches: Hardwarehersteller reichen nicht mehr unbedingt einen aktualisierten Treiber für ein neu erschienenes Betriebssystem nach, da sich das ökonomisch kaum lohnt und es selten in der oft engen Kalkulation vorgesehen wurde. Das Problem verschärft sich dadurch noch, dass der Source-Code der Treiber nicht offenliegt und von Dritten einfach angepasst werden könnte. Die erste Problematik lässt sich gut an Betriebssystemen, wie OS/2 oder BeOS beobachten. Hier sorgt jedoch oft eine aktive Benutzer-Community noch für Entwicklungen für aktuelle Hardware.

### 7.5.3 Quellen und Beschaffung

Abhängig vom eingesetzten Virtualisierer oder Emulator wird eine bestimmte Treiberauswahl benötigt. Handelt es sich um reale Hardware, die virtualisiert wurde, sollten sich vom ursprünglichen Hersteller dieser Hardware die Treiber für die verschiedenen Betriebssysteme beschaffen lassen. Da oftmals Produkte, die zwar auf demselben Design basieren, von mehreren Herstellern angeboten wurden, kann es mehr als eine mögliche Quelle geben. Umgekehrt kann es auch passieren, dass der ursprüngliche Hersteller nicht mehr existiert.

<sup>16</sup>Zu dieser Zeit waren sämtliche Basiskomponenten der Hardware unter direkter Regie von Apple produziert worden. Später wurden Komponenten, wie Grafikkarten und Netzwerkkadaper von Dritten geliefert, inzwischen sind die meisten Teile keine Eigenentwicklungen mehr.



**Abbildung 7.7:** Treiberunterstützung der Microsoft Betriebssysteme am Beispiel realer oder virtueller Netzwerkkarten. Hardware und Betriebssysteme gehören epochenmäßig zusammen.

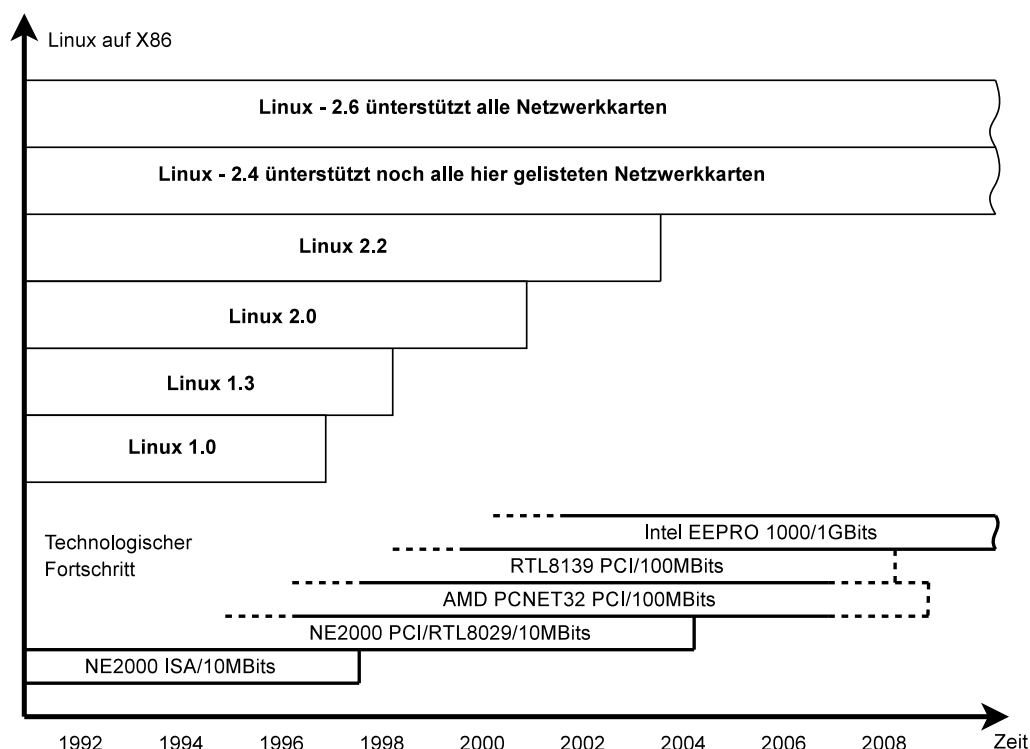
Teilweise lassen sich Treiber finden, die von Dritten erstellt wurden. Hierzu zählen beispielsweise die vor mehr als zehn Jahren populäre Crynwr Packet Driver für eine Reihe populärer ISA- und früher PCI-Netzwerkkarten.

Wenn man nicht als ehemaliger Besitzer der realen Hardware über entsprechende Treiber-Sets verfügt, lassen sich in den meisten Fällen die Treiber im Internet finden und vom Hersteller oder geeigneten Webseiten<sup>17</sup> kostenfrei herunterladen. Im zweiten Fall wäre jedoch die rechtliche Lage zu klären, da es sich um proprietäre Software handelt, deren Nutzungsrecht klassischerweise mit der konkreten Hardware verbunden ist.

Die Beschaffung und Ablage der Treiber ist ein wichtiges Element des Archivmanagements, da wegen mangelnden Interesses nicht davon ausgegangen werden kann, dass die Software auf Ewigkeit im Internet verfügbar sein wird.

Vielfach werden kommerzielle Betriebssysteme mit einer Grundausstattung populärer Hardware ausgeliefert. Dann befinden sich die notwendigen Treiber bereits auf dem Installationsmedium des jeweiligen Betriebssystems. Dieses gilt jedoch selbstredend nur für Hardware, die zum Release-Zeitpunkt des Betriebssystems zur Verfügung stand. Beispiele hierfür sind die Treiber für die NE2000 oder AMD PCNet32 Netzwerkkarten auf den Windows 95 oder 98 Installationsmedien. Jedoch stehen Betriebssystemproduzenten vor einem Problem - ein umfangreiches Treiber-Set muss einerseits auf dem Installationsmedium Platz finden und andererseits gewartet werden.

<sup>17</sup>Es existieren hier eine Reihe mehr oder weniger seriöse, jedoch wechselnde Angebote. Ein Treiber-Set für aktuelle Emulatoren und Virtualisierer wurde im Softwarearchiv hinterlegt.



**Abbildung 7.8:** Treiberunterstützung der Linux-Kernel für populäre Netzwerkkarten. Mit jedem Kernel kommen neue Netzwerkkarten hinzu; bisher wurde noch kein Ethernet-Treiber in der Kernel-Entwicklung aufgegeben.

#### 7.5.4 Hardwaretreiber in Open-Source-Betriebssystemen

Teilweise deutlich einfacher liegt die Situation bei freien Betriebssystemen mit offengelegtem Quellcode. Ebenso wie die Quellen des Betriebssystemkerns stehen die Treiberquellen für unterschiedlichste Hardware bereit. Diese sind entweder von den Herstellern der Hardware bereitgestellt, anhand der Hardwarespezifikationen von Dritten entwickelt worden oder auf dem Wege des Reverse-Engineerings aus kommerziellen Treibern entstanden. Wegen der Quelloffenheit werden Treiber auch für sehr alte Treiber mit der Weiterentwicklung der Betriebssystemkerne mitgeführt. So findet man beispielsweise auch in den aktuellen Linux Kernels immer noch die Unterstützung für die ehemals sehr populären ISA-Bus-Komponenten NE2000-Netzwerkkarte und Soundblaster-16 Soundkarte (Abb. 7.8).

Je nach Hardwarekomponente kann es passieren, dass kein geeigneter Treiber für die freien Betriebssysteme vorliegen. Das betrifft aber eher seltene und recht spezielle Hardware. Bei den Virtualisierern und Emulatoren handelt es sich typischerweise um sehr populäre Hardware mit guter Treiberunterstützung.

## 7.6 Emulatoren

Nach Bestimmung ihrer Auswahlkriterien und Metadaten im Kapitel 4 und einer Übersicht von potenziell für die Langzeitarchivierung geeigneter Projekte und Produkte in Kapitel 5,

geht es in diesem Abschnitt lediglich um ihre Funktion im View-Path und damit ihre Rolle im Softwarearchiv. Sie folgen den Applikationen und Betriebssystemen im View-Path und stellen für letztere die Bezugsumgebung her. Die zweite Dimensionen zur Bestimmung ihrer Auswahl ist die Referenzumgebung, in der sie zum Einsatz kommen sollen (Abb. 7.9). Dabei kann der

| Emu/Virt.      | X86           |             |             |          | PPC    | other |
|----------------|---------------|-------------|-------------|----------|--------|-------|
|                | Linux 2.4/2.6 | Win 2k/XP/V | BSD/Solaris | Mac OS-X | Mac OS |       |
| VM Workstation | x/x           | x/x/-       | -/-         | -        | -      | -     |
| VM Player      | x/x           | x/x/x       | -/-         | -        | -      | -     |
| VM Fusion      | -/-           | -/-/-       | -/-         | x        | -      | -     |
| VM Server      | x/x           | x/x/x       | -/-         | -        | -      | -     |
| Parallels WS   | x/x           | x/x/x       | -/-         | x        | -      | -     |
| Virtual PC/PPC | -/-           | -/-/-       | -/-         | -        | x      | x     |
| Virt PC 04     | -/-           | x/x/-       | -/-         | -        | -      | -     |
| Virt PC 07     | -/-           | x/x/x       | -/-         | -        | -      | -     |
| Virt Server 05 | -/-           | x/x/x       | -/-         | -        | -      | -     |
| Bochs 2.2      | x/x           | x/x/x       | x/x         |          | -      | -     |
| QEMU 0.9.X     | x/x           | x/x/x       | x/x         | x        | x      | x     |
| Dioscuri       | x/x           | x/x/x       | x/x         | x        | x      | x     |
| Virt Box 1.5   | x/x           | x/x/x       | -/-         | x        | -      | -     |

**Abbildung 7.9:** Kommerzielle Virtualisierer sind auf wenige Host-Plattformen beschränkt, freie Emulatoren lassen sich vielfach für viele Plattformen aus ihrem Quellcode übersetzen.

Bezug zur Referenzumgebung durchaus wechselseitig bestehen:

- Existieren mehrere Emulatoren, die für ein Betriebssystem oder bei älteren Plattformen eine Applikation oder generisches dynamisches Objekt geeignet sind, könnte eine Auswahl anhand der bevorzugten Referenzumgebung erfolgen.
- Steht nur ein bestimmter Emulator zur Verfügung und hat der View-Path eine hohe Priorität, so wird man auf weniger präferierte oder seltene Referenzplattformen zurückgreifen.

Diese Entscheidungen liegen beim Archivmanagement. An dieser Stelle erkennt man einen Vorteil des Java-Ansatzes mit einem eigenen Runtime Environment. Er führt eine zusätzliche Unabhängigkeit von der darunterliegenden Rechnerarchitektur ein, unter der Voraussetzung, dass eine geeignete virtuelle Maschine vorhanden ist (siehe hierzu zudem Kapitel 5).

## 7.7 Hilfsprogramme

Mit Hilfsprogrammen sei im Zusammenhang dieser Arbeit Software gemeint, die nicht Basisbestandteil eines Betriebssystems der Referenz- oder Nutzungsumgebung oder des Emulators beziehungsweise Virtualisierers ist. Hierzu zählen einerseits bestimmte Bibliotheken, die zur Ausführung der Emulationssoftware im Host-System benötigt werden. Andererseits werden häufig Programme benötigt, um Objekte in eine Emulationsumgebung hinein oder sie aus dieser wieder heraus zu transportieren.<sup>18</sup> Generell ist für alle zusätzlichen Softwarekomponenten

<sup>18</sup>Vgl. hierzu den Abschnitt 6.8 des vorangegangenen Kapitels.



die Frage der Lizenzierung zu bedenken. Diese ist für Open-Source-Referenz- und Nutzungsumgebungen wegen der GPL oder ähnlicher Lizenzen trivial, für kommerzielle Anwendungen und Bibliotheken hingegen nicht.<sup>19</sup>

### 7.7.1 Softwarebibliotheken für verschiedene Aufgaben

Typischerweise sind Emulatoren als Applikationen für eine bestimmte Host-Plattform nicht in sich komplett. Sie setzen, wie die meisten modernen Anwendungen auch, auf sogenannte "Shared Libraries", gemeinsam benutzte Funktionen beispielsweise für die Ausgabe in grafischen Benutzerumgebungen. Das macht den Programmcode leichter wartbar und den Installationsumfang deutlich kompakter. Deshalb ist bei der Einstellung eines Emulators in ein Archiv festzustellen, gegen welche Bibliotheken dieser dynamisch gelinkt wurde. Virtualisierer, wie die verschiedenen VMware Produkte paketieren benötigte Bibliotheken mit,<sup>20</sup> andere, wie QEMU, überlassen dieses dem Systemverwalter. Jedes Betriebssystem, das mit dynamisch gelinkten Bibliotheken arbeitet, bringt Werkzeuge mit, um festzustellen, welche genau von einer Applikation erwartet werden.<sup>21</sup>

Eine Anzahl Emulatoren setzt für die Interaktion mit verschiedenen Ein- und Ausgabegeräten auf die Simple DirectMedia Layer (SDL) Multimedia-Bibliothek für verschiedene Plattformen.<sup>22</sup> Diese Bibliothek sorgt für eine abstrakte API zum Zugriff auf Sound-, Grafik- und Eingabegeräte. Deshalb setzen sie Entwickler für die Entwicklung von Spielen und Multimediaanwendungen, aber auch für die Programmierung von Emulatoren ein. Beispiele hierfür sind Varianten von QEMU, MAME oder VirtualBox.

### 7.7.2 Host-Plattform

Die Host-Plattform eines Emulators ist typischerweise eine Referenzumgebung, die ein Anwender nutzt, um auf ein bestimmtes digitales Objekt zugreifen zu können. Dieses Objekt wird in dieser Umgebung bereitgestellt und muss geeignet in die jeweils bestimmte Nutzungsumgebung transportiert werden:

- Erstellen von Image-Dateien, die als virtuelle Datenträger durch den Emulator geladen werden können.
- Hinzufügen des digitalen Objekts zu einer virtuellen Festplatte oder Erstellen einer solchen.
- Angebot eines Transportendpunktes für virtuelle Netzwerkverbindungen zum Datenaustausch.

Die Programme zum Erstellen von Image-Dateien, wie Diskettenabbildern oder ISO-Containern unterscheiden sich je nach Referenzumgebung. So eignet sich das Programm "dd"

<sup>19</sup>Vgl. dazu die Überlegungen aus Abschnitt 7.11.

<sup>20</sup>Das ist nicht unbedingt üblich, verhindert jedoch das Problem, dass ein Administrator bei der Installation weitere Pakete hinzufügen muss. Siehe zudem Abschnitt 5.3.

<sup>21</sup>Auf Linux Systemen ermittelt man dieses mit dem Kommando "ldd".

<sup>22</sup>Sie steht unter der LGPL, Homepage - <http://libsdl.org>. Durch die Unterstützung etlicher Plattformen ist bei der Softwareentwicklung eine hohe Portabilität gewährleistet.

unter Linux oder "rawrite" unter Windows für die Erzeugung von virtuellen Disketten (Abschnitt 6.8.2). Die Auswahl der Programme zur Generierung von ISO-Images ist vielfältig: Für alle Referenzumgebungen existieren sogenannte Brennprogramme, die solche Dateien generieren können. Für Linux Systeme bietet sich besonders das Programm "mkisofs"<sup>23</sup> an, welches üblicherweise Bestandteil einer Linux Distribution ist.

Die Containerformate der einzelnen Emulatoren oder Virtualisierer sind typischerweise proprietär. Deshalb beinhalten die jeweiligen Softwarepakete oft Werkzeuge zu ihrer Bearbeitung. Beispiele hierfür sind das VMware Tool "vmware-mount.pl"<sup>24</sup> oder "qemu-img", ein Containerkonverter des QEMU Pakets. Gerade das letztgenannte Programm ist von Interesse, da es Container unterschiedlicher Virtualisierer ineinander umwandeln und Leercontainer erstellen kann. Es erlaubt auf diese Weise, äußere Migrationen<sup>25</sup> vorzunehmen.

Sollen Daten mittels virtueller Netzwerke ausgetauscht werden, müssen die dafür notwendigen Protokolle vorhanden sein. Während der Einsatz von IPv4 sicherlich unproblematisch ist und das auch noch längere Zeit nach einem Umstieg auf IPv6 bleiben wird, kann dieses für ältere Protokolle wie Appletalk oder IPX/SPX problematischer werden. Das Host-System muss die alten Protokolle bereitstellen können, was zunehmend unwahrscheinlicher wird. Ebenso werden darauf aufsetzende Dienste, wie FTP- oder SMB/CIFS-Server benötigt (Abschnitt 6.8.1).

Eine eigene Softwareklasse bilden Programme, die grafische Frontends für sonst kommandozeilengesteuerte Emulatoren abgeben. Sollen Benutzer selbst Einstellungen vornehmen können, sind solche Werkzeuge als Vereinfachung von Interesse. Andernfalls eignen sich gerade kommandozeilenorientierte Emulatoren für eine bessere Einbindung in Frameworks zur Darstellung und Steuerung.<sup>26</sup>

Liegen die interessierenden Primärobjekte in bestimmten Paketierungen vor, so kann es notwendig sein, diese im Referenzsystem behandeln zu können. Dann werden Archivierungs- und Packprogramme benötigt, welche mit den eingesetzten Komprimierungs- und Packalgorithmen umgehen können. Oft ist es zudem leichter, Entpackvorgänge mit komfortablen Werkzeugen in der Referenzumgebung als mit oft sehr einfachen und schwer zu bedienenden Tools in der Nutzungsumgebung vorzunehmen.

### 7.7.3 Nutzungsumgebung

In den jeweiligen Nutzungsumgebungen sind die Endpunkte der Datentransportpfade zu berücksichtigen. Vom Einsatz virtueller Datenträger sieht die Nutzungsumgebung nichts – sie stellen sich wie gewöhnliche Laufwerke der verschiedenen Typen dar. Hingegen sind für den Netzwerktransport üblicherweise eine Reihe von Protokollen und hierfür verschiedene Soft- und Hardwaretreiber notwendig. Wenn diese nicht vom Betriebssystem der Nutzungsumgebung bereits mitgeliefert werden, dieses ist beispielsweise bei DOS und Windows 3.X nicht

<sup>23</sup>Eine ausführliche Beschreibung zur Handhabung liefert man `mkisofs`. Inzwischen existiert ein Nachfolger mit dem Namen "genisoimage".

<sup>24</sup>Dieses Perl-Skript erlaubt das Analysieren von VMDK-Containern und das Mounten darin enthaltener Partitionen. Ein einfaches Loopback-Mount ist wegen der üblicherweise vorhandenen Partitionierung (Abb. 6.19) anders als bei ISO- und Disketten-Images nicht realisierbar.

<sup>25</sup>Vgl. hierzu Abbildung 5.6.

<sup>26</sup>Siehe hierzu [Welte 2008].

der Fall, sind sie zusätzlich bereitzuhalten und zu installieren.

Darüber hinaus können verschiedene Pack- und Kompressionsprogramme benötigt werden, da Softwarekomponenten, aber auch Primärobjekte, in einer solchen Form angeliefert werden könnten.

Einige statische Primärobjekte sollen vielleicht später auf Papier ausgegeben oder in ein besser handhabbares Dokumentenformat umgewandelt werden. Da die wenigsten älteren Viewer oder Editoren über entsprechende Exportfilter verfügen, lassen sich in einigen Nutzungsumgebungen virtuelle Druckertreiber für Postscript und PDF installieren.

## 7.8 Archivmanagement

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 Zielobjekte. Trotzdem kann es von Interesse sein, diese Daten in direkt zugreifbarer Weise oder auch in speziell aufbereiteter Form vorzuhalten.

Der Umfang der Sekundärobjekte wird für die meisten Archivtypen deutlich unter der der Primärobjekte<sup>27</sup> liegen. Deshalb kann man bei Nutzung eines gemeinsamen Speichersystems die Datenmenge des Softwarearchivs vernachlässigen. Nichtsdestotrotz steigt die Menge der zu verwaltenden Daten und Metadaten im Zeitverlauf unaufhörlich an. Selbst ein regelmäßiges Aussondern nicht mehr notwendiger Sekundärobjekte, wie alte View-Path-Caches, wird diesen Trend nicht umkehren.

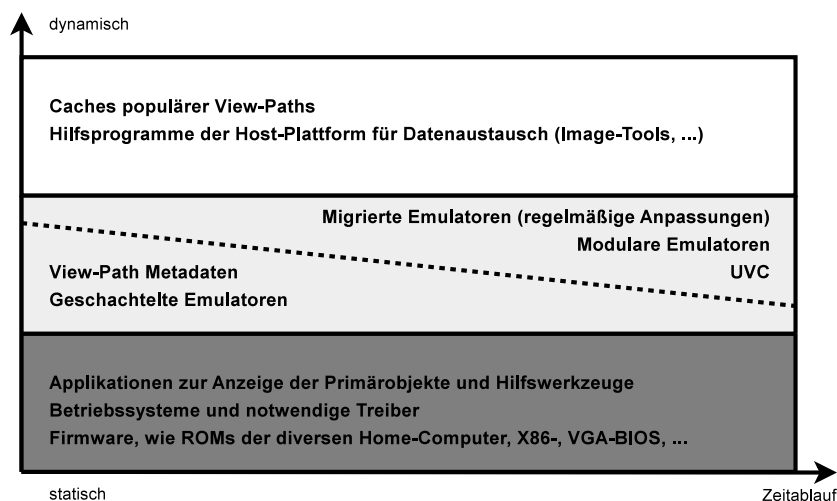
Ein weiterer Problemkreis ergibt sich aus der Art der Sekundärobjekte. Anders als bei den meisten Zielobjekten werden im Laufe der Zeit Änderungen oder Ergänzungen notwendig, die im Archiv berücksichtigt werden sollten.<sup>28</sup> Der Archivinhalt besteht aus unterschiedlichen statischen und veränderlichen Softwarekomponenten (Abb. 7.10):

**Betriebssysteme und Applikationen** alter Nutzungsumgebungen und ihre Hilfskomponenten ändern sich nicht mehr. Hierin liegt ein wesentlicher Vorteil vergangener Rechnerplattformen: Die Weiterentwicklung ist stehengeblieben und es kommen keine neuen Programme oder Aktualisierungen hinzu. Änderungen an Betriebssystemen oder Treibern lassen sich nicht mehr vornehmen, da in den seltensten Fällen ein Zugriff auf den Quellcode, die Entwicklungsumgebung und mit zunehmendem Alter die notwendige Erfahrung für den Umgang damit realistisch ist.

**Emulatoren und Komponenten** sind je nach gewählter Strategie mit der Entwicklung der Referenzumgebung mitzuführen. Bei geschachtelter Emulation bleiben Emulatoren unverändert und ihre Nutzungsumgebung wird nachgebildet. Dann kann es jedoch notwendig sein, weitere Metadaten hinzuzufügen.

<sup>27</sup>Vergleiche hierzu die umgedrehte Pyramide in Abbildung 3.8.

<sup>28</sup>Siehe hierzu auch die Aufgabenstellungen an das Archivmanagement aus Sicht der View-Paths in Abschnitt 6.7.



**Abbildung 7.10:** Während Betriebssysteme, Platform-ROMs und Applikationen unveränderlich bleiben, sind je nach Emulationsstrategie Aktualisierung von Emulatoren oder ihren Teilkomponenten notwendig. Vorbereitete Nutzungsumgebungen müssen mit jedem Update der Host-Plattform erneut auf diese angepasst werden, ebenso wie eine Reihe von Hilfsprogrammen.

Sieht die gewählte Strategie eine Migration von Emulatoren vor, so muss diese mit jedem Wechsel der Referenzumgebung durchgeführt werden. Für modulare Ansätze genügt es hingegen die veränderlichen Komponenten, neu zu erstellen.<sup>29</sup>

**Vorbereitete Nutzungsumgebungen** beziehen sich immer auf eine ganz bestimmte Referenzumgebung. Da diese nicht dauerhaft existiert und zwangsläufig dem technologischen Wandel unterliegt, muss bei jeder Änderung eine Anpassung oder Neuerstellung erfolgen.<sup>30</sup>

## 7.9 Verteilung und Redundanz

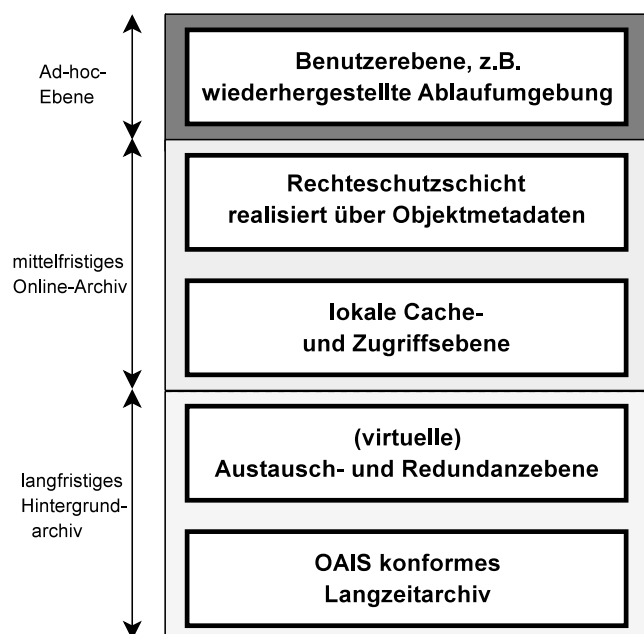
Die Aufbewahrung der Sekundärobjekte sollte mit einer ähnlichen Sorgfalt wie die der Primärobjekte erfolgen. Das bedeutet, dass für den Archivteil der Sekundärobjekte ein ähnlicher Aufwand eingeplant werden muss. Anders als viele Primärobjekte werden die Sekundärobjekte häufig nachgefragter View-Paths 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 die folgenden Punkte abdecken sollten:

- Schaffung einer hohen, auch räumlichen Redundanz der Daten
- Geeignete Sicherung der Objekte und Vergabe der Zugriffsrechte
- Zügiger und komfortabler Zugriff der Archivnutzer auf oft benutzte View-Paths

<sup>29</sup>Siehe hierzu die Diskussion im Abschnitt 4.8 und 6.7.

<sup>30</sup>Vgl. Abschnitt 6.6.

- Sinnvolle und kosteneffektive Verteilung des Aufwands zur Archivpflege



**Abbildung 7.11:** Vorschlag logischer Schichten für das Langzeitarchiv der Sekundärobjekte zur Realisierung von Zugriff und Verteilung.

**Langfristiges Hintergrundarchiv** Das wird bei den meisten Institutionen das vorhandene OAIS-konforme Langzeitarchiv der Primärdaten sein (Abb. 7.11). 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 Ebene kann dazu genutzt werden, je nach Archivierungsrichtlinien eine Auswahl der zu verteilenden Objekte zu treffen. Diese Auswahl sollte aus Redundanzgründen möglichst nicht durch einen Rechtsschutz eingegrenzt werden. Diese Beschränkung sollte sich im Hinblick auf eine Beschränkung auf das Archivmanagement hinnehmen lassen.

Die Austauschschicht lässt ebenso eine Verteilung der Aufgaben zwischen Institutionen oder ihren Einheiten zu: So können sich einzelne Einheiten auf bestimmte Bereiche der Sammlungen, wie einzelne Plattformen spezialisieren, ohne damit den Anschluss in anderen Bereichen zu verlieren. Dadurch lassen sich die Kosten einer aufwändigen Pflege bei trotzdem hoher Archivqualität senken. Zudem kann auf diese Weise das Problem nicht überall verfügbarer und teilweise sehr selten nachgefragter Spezialisten reduziert werden.

Der Austausch der Daten könnte mittels Peer-2-Peer-Technologien erfolgen, wie sie in diesem Zusammenhang in [Cooper und Garcia 2001], [Cooper und Garcia-Molina 2002] oder mit LOCKSS (Lots Of Copies Keep Stuff Safe, [Maniatis u. a. 2003]) diskutiert werden. Generelle Überlegungen zur Absicherung elektronischer Objekte über autonome Institutionen hinweg diskutieren [Cooper und Garcia-Molina 2002]. Sie klassifizieren verschiedene Management-Schemata unter einer Reihe von Gesichtspunkten. Ein Beispiel für ein traditionelles digitales

Langzeitarchiv, welches auch auf räumliche Datenredundanz setzt, ist Kopal.<sup>31</sup>

**Mittelfristiges Online-Archiv** Dieser Archivteil befindet sich deutlich in Benutzernähe. Über diesen greifen Archivnutzer oder das entsprechende Archivpersonal auf Sekundärobjekte zur Wiederherstellung von Nutzungsumgebungen zu. Auf dieser Ebene wird ein Rechtemanagement sinnvollerweise mittels Metadaten, wie sie beispielsweise Dublin Core oder PREMIS vorsehen,<sup>32</sup> realisiert. Das erlaubt einerseits Rechte auf den jeweiligen Benutzer oder die Institution zuzuschneiden und verhindert andererseits Probleme beim Datenaustausch zwischen den Institutionen. Es vermeidet unter Umständen riskante Änderungen an den im Archiv enthaltenen Softwarekomponenten. Für häufig nachgefragte View-Paths sollte ein Objekt-Cache (Abschnitt 6.6) angelegt werden. Das entlastet das Langzeitarchiv von wiederholten Anfragen. Solche Caches bekommen nicht unbedingt den Status, wiederum selbst langzeitarchiviert werden zu müssen. Ist die Situation des Rechtemanagements wegen einfacher Richtlinien unkritisch, so ließen sich die beiden Schichten tauschen, was den Zugriff auf vorbereitete View-Paths nochmals vereinfachen und beschleunigen kann.

**Ad-hoc-Ebene** Hierbei handelt es sich um die Schicht, auf der der Archivbenutzer agiert. Er hat hier die Möglichkeit, alle zum Zugriff auf die ihn interessierenden Primärobjekte notwendigen Schritte einzuleiten. Das erfolgt innerhalb der von der Gedächtnisorganisation vorgegebenen Referenzumgebung.<sup>33</sup> Sinnvollerweise wird diese Referenzumgebung jeweils erneut hergestellt, um einen sauberen Bezugspunkt zu bieten.

## 7.10 Archivierung von Datenträgern

Wie bereits in Abschnitten 2.4 und 2.5 ausgeführt, löst die Aufbewahrung der originalen Hardware das Problem der Langzeitarchivierung digitaler Objekte nicht langfristig. Deshalb kann man nicht davon ausgehen, dass zu jeder Zeit geeignete Geräte und Schnittstellen zum Auslesen von Datenträgern zur Verfügung stehen. In diesem Umfeld sind daher zwei Fragenkomplexe zu klären:

- Wie überführt man die verschiedenen Installationsdatenträger, wie Disketten, CDs, DVDs, Magnetbänder in eine langzeitarchivierbare Form und
- Wie erfolgt der spätere Zugriff auf die archivierten Daten?

Die Softwarewerkzeuge zum Behandeln von Datenträgern sind nicht Bestandteil späterer View-Paths. Sie sollten jedoch trotzdem im Softwarearchiv vorgesehen und bei Bedarf zu aktualisiert werden, damit auch eine spätere Extraktion von Datenträgern möglich bleibt.<sup>34</sup>

Aufgrund des technologischen Wandels und der oft unklaren "Haltbarkeit" der aktuell verwendeten Datenträger und ihrer Lesegeräte bleibt es für den Anwender unumgänglich, in

<sup>31</sup>Vgl. hierzu die Homepage [kopal 2008].

<sup>32</sup>Vgl. hierzu Abschnitt im vorderen Teil.

<sup>33</sup>Vgl. hierzu Abschnitt 6.5.

<sup>34</sup>Vgl. hierzu Abschnitt 6.2.1 zur Softwarearchäologie.

bestimmten Zeitintervallen einen Wechsel des Mediums vorzunehmen. Dieser ist, wenn die Zeitabstände nicht zu groß werden, oft recht einfach und kostengünstig möglich:

1. Der Inhalt einer kleinen Festplatte wird auf eine größere Festplatte kopiert. Hat sich die Schnittstellenspezifikation des Gerätes geändert, hilft oft der Umweg über ein Netzwerk, welches zwei Rechner miteinander verbindet.
2. Der Inhalt von Disketten wird als Image auf CD oder Festplatte abgelegt. Bei letzterem erfolgen Updates einfach mit dem Wechsel der Festplatte.
3. Der Inhalt von CDs wird als Image auf Festplatte abgelegt und in den Update-Zyklus von Punkt eins eingespeist.
4. DVDs können als Abbilddatei auf Festplatte gespeichert und damit in den erstgenannten Update-Zyklus aufgenommen werden.
5. Der Inhalt von Magnetbändern kann eingelesen und als Einzeldatei mit dem Gesamt-datenstrom oder als die ursprünglichen Einzeldateien auf Festplatte abgelegt werden.

### 7.10.1 Rechtzeitige Datenträgermigration

Vielfach liegen die zu archivierenden Objekte noch nicht als einfach zu transportierende Bitströme im Archiv vor, sondern sind oft noch an ihr Ursprungsmedium gebunden. Dieses gilt insbesondere für die Installationsmedien aller Arten von Software: Sie kommen üblicherweise bei der Einrichtung einer Maschine zum Einsatz und werden dann mehr oder weniger geeignet archiviert.

Das generelle Bild, welches sich bei Datenträgern für Computersysteme bietet, ist uneinheitlich:

- Disketten im 3,5" Format gibt es seit mehr als 20 Jahren. Die Speicherkapazitäten haben sich von 720 kByte, 1,44 MByte auf 2,88 MByte bei LS120-Disketten auf 120 MByte pro Medium<sup>35</sup> gesteigert, wobei sich Laufwerke mit der "hohen" Kapazität kaum durchgesetzt haben. Fast alle Hochkapazitätslaufwerke waren darauf ausgelegt, Disketten niedrigerer Kapazität zu lesen.
- CDs haben sich befördert durch den bereits existierenden Standard der Musikindustrie seit Mitte der 1990er Jahre als Datenträger durchgesetzt. Die Kapazität erhöhte sich im Laufe der Zeit von ursprünglich 650 MByte auf 700 MByte und teilweise 800 MByte. Versuche, die Kapazität auf 1,3GByte zu verdoppeln, konnten sich am Markt nicht durchsetzen.
- DVDs, in Zukunft auch HD-DVDs und Blu-ray-Disks lösen sukzessive CD-ROMs als Datenträger ab, ohne sie auf längere Sicht komplett zu ersetzen. Ihre Grundkapazität liegt bei 4,7 GByte.

---

<sup>35</sup>Kurze Beschreibung in Abschnitt A.5.1.

- Zwar sind bei Festplatten die Datenträger verglichen mit Wechsellaufwerken am besten geschützt, jedoch sorgt die feine Mechanik und die feste Verbindung von Steuerungslogik und Datenträger dafür, dass die Ausfallwahrscheinlichkeit gerade bei einfachen und sehr preiswerten Festplatten recht hoch ist. Anders als bei Wechsellaufwerken kann nicht trivial bei Ausfall der Steuerungslogik der Datenträger in einem weiteren Laufwerk ausgelesen werden.

Die meisten genannten Systeme lassen sich als Wechselmedien bezeichnen. Ihnen ist gemeinsam, dass Abspielgerät und Datenträger sich nach dem Lese- oder Schreibvorgang voneinander trennen lassen. Die Datenträger können über denselben Typ Abspielgerät frei ausgetauscht werden.

Während Disketten-, CD-, und DVD-Laufwerke mit der Definition des Standards nahezu unverändert bleiben, stellt sich das Bild bei Festplatten etwas anders da. Die Kapazitäten produzierter Festplatten steigt beständig an. Während zu Beginn der 1990er Jahre 40 - 200 MByte üblich waren, sind inzwischen Größenordnungen von 200 - 1000 GByte erreicht. In nur 15 Jahren hat sich das Fassungsvermögen dieser Datenträgerkategorie mehr als vertausendfacht, wobei der Preis des Einzelgerätes dabei noch gefallen ist. Einen kurzen technischen Überblick zu Festplatten bietet Abschnitt A.5.2 im Anhang.

Es erscheint nicht vernünftig, ein Medium erneut aufzufrischen, das am Ende seiner technologischen Spanne angekommen ist. So ist beispielsweise derzeit ein Aussterben der 3,5" Disketten und ihrer Lesegeräte zu beobachten. Da die auf einer Diskette enthaltene Datenmenge von 1,44 MByte aus heutiger Sicht lächerlich gering ist, entsteht der Wunsch auf kompaktere Speicherung gerade von größeren Beständen dieses Datenträgers. Hier bietet sich das Kopieren auf andere, technologisch jüngere Medientypen an. Hier entsteht jedoch möglicherweise ein Problem: Bestimmte Systeme, wie beispielsweise ältere Home-Computer können nur mit dem Laufwerkstyp Diskette umgehen. Sie stellen keinerlei Treiber für Festplatten oder optische Wechsellaufwerke zur Verfügung, womit Daten, die auf neueren Medien gespeichert wurden, für diese Rechnerarchitektur quasi "unsichtbar" werden.

Gleiches gilt, wenn sich die Zugriffsformen auf Daten verändern. Auf Magnetbändern werden Daten sequenziell als Bitströme abgelegt und mit speziellen Start-, Trenn- und Endmarken gekennzeichnet. Festplatten und optische Laufwerke speichern Daten blockorientiert, auf die einzeln, unabhängig voneinander zugegriffen werden kann. Veränderungen des Datenzugriffs dieser Art setzen neben oft notwendigen Schnittstellenanpassungen der Hardware erhebliche Modifikationen im Betriebssystem der jeweiligen Rechnerarchitektur voraus.

**Datenträger umkopieren** An dieser Stelle kann keine generelle Migrationsstrategie angegeben werden. Jedoch treffen die Überlegungen für ein Softwarearchiv von Sekundärobjekten auf viele ähnlich gelagerte Fälle zu. Auf eine Standard-CD passt insgesamt der Inhalt von 486 3,5" Disketten. Eine DVD kann knapp sieben Standard-CDs aufnehmen. Bei allen Angaben wurde davon ausgegangen, dass alle Datenträger in ihrer Maximalkapazität mit Daten bestückt sind. Dieses ist jedoch oft nicht der Fall, so dass es gerade bei kleineren Datenträgern einen unverhältnismässig hohen Verschnitt gibt. 200 MByte freie Kapazität sind auf einer CD schon ein Viertel der Gesamtkapazität, auf einer 250 GByte-Festplatte hingegen zu vernachlässigen.

Beim Kopieren von Datenträgern gibt es eine weitere Herausforderung: Es ist nicht zu



| Datenträger   | Kapazität  | n-fach auf 500 GByte |
|---------------|------------|----------------------|
| 3,5" Diskette | 1,44 MByte | 345.000              |
| Daten-CD      | 700 MByte  | 700                  |
| Daten-DVD     | 4,7 GByte  | 100                  |

**Tabelle 7.1:** Kapazitätsvergleich verschiedener Datenträger im Verhältnis zu einer Festplatte der durchaus üblichen Fassungsgröße eines halben Terabytes (gerundet).

unterschätzen, dass unter Umständen wichtige Metadaten, wie Diskettenbeschriftungen oft vergessen werden oder aufgrund der schier Menge sich nicht einfach auf das Beschriftungsfeld von CDs, DVDs oder externe Festplatten übertragen lassen. Mit der steigenden Anzahl der Dateien, nimmt der Aufwand für ihre Organisation zu, ein Umstand dem bei der Einrichtung eines Softwarearchivs Rechnung getragen werden sollte.

### 7.10.2 Dateien direkt auslesen

Die gewünschte Langzeitexistenz von Installationsdatenträgern hängt von ihrer späteren Nutzung ab. Diese Datenträger sind zusammengefasste digitale Objekte, die aus mehr als einem Teilobjekt bestehen und erst in dieser Kombination verwendbar sind. Für die spätere Ablage im Archiv kommt es darauf an, ob eine bestimmte Anordnung der Einzelkomponenten zueinander auf dem Datenträger notwendig ist oder ob diese aufgelöst werden kann. Aus diesen Prämissen ergeben sich zwei Ablageformen:

- Stapel von Installationsdisketten liefern ein klassisches Beispiel: Für Betriebssysteme verfügt die erste Diskette typischerweise über eine Boot-Fähigkeit. Die weiteren Datenträger enthalten alle Dateien, die nicht für den Boot-Vorgang benötigt werden oder keinen Platz auf der ersten Diskette fanden. Damit ist nur für die erste Diskette die Blockstruktur im Gesamtzusammenhang entscheidend und die anderen Daten können je nach Software als Dateien kopiert werden. Trotzdem kann es unter Authentizitätssichtspunkten interessant sein, alle Datenträger in ihrem blockweise logischen Aufbau zu archivieren. Diese Option bleibt die einzig realisierbare, wenn das Dateisystem des Datenträgers im Host-System nicht bekannt ist.
- In Abhängigkeit der Installationsroutinen<sup>36</sup> von Software lassen sich viele kleine Installationsdatenträger auf einem gemeinsamen großen zusammenfassen. Ein Beispiel ist Windows 3.11, welches auf neun Disketten ausgeliefert wurde, aber auch aus einem einzigen Unterverzeichnis einer CD eingerichtet werden konnte.<sup>37</sup> Grundbedingung dieses Vorgehens ist die Unterstützung des Dateisystems auf dem Datenträger.

Daher bleibt es bei der Archivaufnahme durchaus freigestellt, ob einzelne Dateien bereits im Moment des Einlesens von Disketten kopiert werden. Dieses lässt sich zu einem späteren

<sup>36</sup>Üblicherweise kümmert sich der Installer um die Aufforderung an den Nutzer den nächsten Datenträger einzulegen und nach Ausführung das Laufwerk erneut einzubinden. Einige Software erwartet unbedingt eine nächste Diskette, andere sind deutlich flexibler.

<sup>37</sup>Dieses wurde im Zuge der PLANETS-begleitenden Experimente zur Erzeugung eines zusammengefassten ISO-Images zur Installation verschiedener X86-Betriebssysteme erfolgreich in der Arbeitsgruppe getestet und genutzt.

Zeitpunkt nachholen. Dann lässt sich der Vorgang zudem komplett automatisieren und auf viele Datenträger nacheinander anwenden, da keine Datenträgerwechsel in einem physischen Laufwerk vorgenommen werden müssen.

In bestimmten Fällen kann es sinnvoll sein, Dateien direkt vom Datenträger zu kopieren. Während die Klasse der block-orientierten Leseprogramme blind gegenüber dem auf dem Datenträger enthaltenen Dateisystem ist, gilt dieses nicht für direkte Kopien. Deshalb besteht eine wichtige Voraussetzung in der Fähigkeit des Host-Systems das auszulesende Dateisystem zu "kennen". Auf diese Weise können Dateien von alten Dateisystemen gelöst werden, bei denen in Zukunft die Gefahr besteht nicht mehr direkt in einer aktuellen Referenzumgebung gelesen werden zu können.

Während die aktuellen Linux-Kernel der 2.4 und 2.6er Versionen fast alle gebräuchlichen Dateisysteme kennen, da die Treiber bereits im Basiskernel enthalten sind, sieht das Bild bei Microsoft-Betriebssystemen deutlich düsterer aus. Bei letzteren gilt zudem oft, dass für bestimmte Dateisysteme weitere Lizenzen erworben werden müssen.

### 7.10.3 Virtuelle Datenträger

Der überwiegende Teil der im Archiv abzulegenden Sekundärobjekte, wie Betriebssysteme und Applikationen, wurde im ursprünglichen Zustand auf Installationsdatenträgern<sup>38</sup> ausgeliefert. Die zu Anfang genannten Problemkreise manifestieren sich beim direkten Umgang mit der jeweiligen Rechnerhardware und den plattformtypischen Datenträgern.

Deshalb sollten rechtzeitig alle interessierenden Datenträger in ihre Abbilder umgewandelt werden, die sich als Dateien im jeweils verfügbaren Festspeichersystem ablegen lassen. Diese Virtualisierung der Datenträger, das Übertragen der Bitströme in ein Langzeitarchiv, vereinfacht eine spätere Migration erheblich, wenn statt vieler Datenträger, die teilweise manuell gehandhabt werden müssten, lediglich Dateien kopiert werden. Dann liegen Installationsdatenträger in einem für ein Archivmanagement unproblematischen Zustand vor. Einen abstrakten Überblick zur Archivmigration und den damit verbundenen Problemstellungen liefert zudem [van Diessen und van Rijnsoever 2002].

Solange dabei die vereinbarten Standards<sup>39</sup> eingehalten werden, bereitet die Umwandlung eines physischen Blockgeräts in ein logisches Blockgerät in Form einer Datei keine Schwierigkeiten. Sind jedoch an den physikalischen Datenträger Kopierschutzmaßnahmen gebunden, die sogar teilweise durch bewusste Verletzung von Standards das korrekte Auslesen der Datenträger unterbinden, kann es unmöglich werden, Datenträger ins Archiv zu übertragen (Abschnitt 7.11).

---

<sup>38</sup>Obwohl der Käufer von Software eine Garantie auf die Funktionsfähigkeit der Datenträger erhält, so ist diese jedoch nach einem gewissen Zeitraum erloschen. Ein Ersatz seitens des Herstellers lässt sich üblicherweise nur in den Zeitspannen des Produktsupports erwarten.

<sup>39</sup>Diese sind allein schon notwendig dafür, dass Datenträger in unterschiedlichen Geräten gelesen und geschrieben werden können.

### 7.10.4 Installationsdisketten

Nach Magnetbändern gehören Disketten<sup>40</sup> zu den ältesten Installationsdatenträgern überhaupt. Das bedeutet einerseits, dass viele Betriebssysteme und Anwendungsprogramme auf diesen Medien ausgeliefert wurden und andererseits einige Betriebssysteme direkt von diesem Medium aus arbeiteten, wenn keine weiteren Datenträger in der Maschine vorgesehen waren.

Disketten haben ein seit vielen Jahren einheitliches Blockformat, welches sich direkt in eine Abbildungsdatei überführen lässt. Die Voraussetzung hierfür besteht lediglich darin, dass die Standards des Zugriffs eingehalten und keine Kopierschutztechniken am Medium angebracht wurden. Disketten konnten in unterschiedlicher Form gebraucht werden:

- Zu startender Code wurde einfach blockweise auf die Diskette geschrieben. Es gibt kein darunterliegendes Dateisystem oder eine Partitionierung. Dieses wurde beispielsweise für das direkte Booten von Linux Kernels angewandt, ist aber eher die Ausnahme.
- Die Diskette enthält ein Dateisystem, jedoch ohne Partitionstabelle. Handelt es sich nicht um ein Boot-Medium und ist das Dateisystem dem Host-System bekannt, so können enthaltene Dateien und Verzeichnisse direkt kopiert werden. Hierzu wird die Diskette oder ihr Abbild im direkten oder sogenannten "Loopback Mount"<sup>41</sup> in das Dateisystem des Hosts für den direkten Zugriff eingehängt.
- In sehr seltenen Fällen enthält die Diskette eine Partitionstabelle (vergleiche hierzu Abbildung 6.19). Dann kann sie nur als Abbild archiviert werden.

Bei der Übertragung von Disketten als komplettes Abbild entsteht eine neue Datei, die nun problemlos im Dateisystem oder Archiv abgelegt werden kann. Hierbei sollte man jedoch beachten, dass im Zuge des Kopiervorganges alle Metadaten vom Datenträger (Beschriftungsfeld der Diskette) ebenfalls geeignet übernommen werden.

Die notwendigen Laufwerke sind für 3,5" Disketten noch ohne Probleme, für 5,25" Disketten jedoch nicht mehr neu beschaffbar. Eine gute Alternative sind USB-Laufwerke, die es für den kleineren Diskettentyp gibt. Unabhängig von der Art der Schnittstelle lassen sich Disketten 1:1 blockweise in ein Disketten-Image überführen.

Das Unix-Tool "dd" gehört zu den Basiswerkzeugen eines jeden freien Unix-Systems<sup>42</sup>. Es kann lesend und schreibend auf die entsprechenden Blockdevice-Dateien des Kernels zugreifen. Solange am Originaldatenträger kein physischer Kopierschutz angebracht wurde, der auf einer proprietären, nichtstandardkonformen Nutzung von Laufwerkeigenschaften beruht, kann das Programm Inhalte unabhängig vom verwendeten Dateisystem von Disketten lesen und auf diese schreiben.

Für DOS- oder Windows-Plattformen steht das Programm "rawrite" zur Verfügung. Dieses kann nach demselben Prinzip und den gleichen Einschränkungen Disketten in ihre Images überführen. Frühe Linux- und BSD-Anwender kennen dieses Programm schon seit mehr als

<sup>40</sup>Siehe hierzu auch Abschnitt A.5.1.

<sup>41</sup>Ein direkter Zugriff auf die Daten in einer Image-Datei ist nicht möglich. Auf Linux oder BSD Unix Systemen wird durch das Loopback aus Sicht des Betriebssystems aus einer Datei ein Datenträger, auf den dann wie gewohnt zugegriffen werden kann: `mount -o loop floppy.img /mnt`.

<sup>42</sup>Es ist Bestandteil der Grundinstallation der diversen Linux- und BSD-Varianten.

zehn Jahren. Es bot die Möglichkeit, unter der DOS-Betriebssystemklasse bootfähige Disketten mit einem für DOS nicht bekannten Dateisystem zu schreiben. Inzwischen existiert von diesem Programm ebenso eine komfortabel zu bedienende Windows-Variante, die alle Betriebssystemvarianten bis zu Windows Vista abdeckt.

Unabhängig vom Kopierprogramm stellte sich heraus, dass auch mehr als 15 Jahre alten Disketten, die von einem Hersteller oder Distributor beschrieben worden waren, noch zu nahezu 100% gelesen werden konnten. Eine hohe Fehlerquote trat hingegen bei älteren Datenträgern auf, die in einem Endgerät (Diskettenlaufwerk im System des Benutzers) erzeugt wurden. Die erzeugten Images sind untereinander austausch- und in sämtlichen Emulatoren und Virtualisierern einsetzbar. Die genannten Kommandos sind agnostisch gegenüber dem Inhalt der Diskette, so dass ohne Schwierigkeiten auch Disketten ohne der Referenzplattform bekanntes Dateisystem ausgelesen werden können.<sup>43</sup>

### 7.10.5 Optische Datenträger

CDs und zunehmend auch DVDs sind die klassischen Installationsdatenträger, die seit knapp zehn Jahren Disketten abgelöst haben. Sie sollten als mindestens für Betriebssysteminstallationen Image-Dateien archiviert werden, da eine ausreichende Langlebigkeit der Datenträger und Lesegeräte in den Zeiträumen einer wirklichen Langzeitarchivierung nicht sichergestellt werden kann.<sup>44</sup> Ein zunehmendes Problem für die Überführung der Datenträger in Image-Dateien stellen Kopierschutzmaßnahmen am Datenträger dar (Abschnitt 7.11).

Anders als die teilweise sehr speziellen Dateisysteme der einzelnen Betriebssysteme handelt es sich beim ISO 9660 um einen weitgehend vereinheitlichten Standard, der nur über einige betriebssystemspezifische Erweiterungen verfügt und deshalb in einer sehr großen Menge von Umgebungen gelesen werden kann.

Das Auslesen von Daten-CDs und DVDs kann auf verschiedenen Wegen vorgenommen werden. Zum einen steht mit dem zuvor vorgestellten *dd* ein Universalprogramm zum Auslesen von blockorientierten Geräten unter Unix-Systemen zur Verfügung. Weiterhin bietet jedes halbwegs aktuelle Programm zum Brennen optischer Datenträger die Option, Image-Dateien von der Quelle zu erstellen. Als Ergebnis dieser Arbeitsschritte erhält man ISO-Images, die sich als virtuelle CD/DVD-Laufwerke an die verschiedenen Emulatoren hängen lassen.<sup>45</sup> So können, solange nicht ein Kopierschutz am Datenträger diesen Vorgang verhinderte, erfolgreich Image-Dateien von verschiedenen Datenträgertypen<sup>46</sup> erzeugt werden.

<sup>43</sup>Durchgeführte Experimente mit IBM-, MS-, DR-DOS, VFAT- und HFS-Disketten konnten das sowohl für eine direkte Kopie auf einen anderen Datenträger gleichen Typs, als auch für die Überführung in Image-Dateien bestätigen.

<sup>44</sup>Derzeit werden neue Generationen von optischen 5,25"-Laufwerken rückwärtskompatibel angelegt, so dass beispielsweise aktuelle Blu-ray-Lesegeräte auch mit CD-ROMs und DVDs umgehen können. Jedoch könnte dieser technologische Entwicklungspfad wegen Miniaturisierung der Datenträger oder einer anderen, überlegenen Speicherart abreißen.

<sup>45</sup>Sämtliche der im Kapitel 5 vorgestellten Emulatoren und Virtualisierer (bis auf Dioscuri) unterstützen virtuelle optische Laufwerke, so diese in der jeweiligen Hardwarearchitektur vorgesehen waren. Vielfach kein Zugriff besteht auf Video-DVDs aus einer Emulations- oder Virtualisierungsumgebung heraus.

<sup>46</sup>Hierzu zählen Originaldatenträger und Backups auf einmal und mehrfach beschreibbare Medien für CD und DVD. Blu-ray Datenträger traten noch nicht als Installationsmedium für die X86-Architektur auf.

In den jeweiligen Referenzumgebungen, wie Linux oder Windows stehen verschiedene Methoden zur Einbindung der erzeugten ISO-Images zur Verfügung:

- Ebenso wie Disketten-Images lassen sich ISO-Images unter Linux per "Loopback" einbinden.
- Für Windows existieren spezielle Programme für virtuelle CD-Laufwerke, die ihre eigenen Treiber mitbringen.

**Virtuelle Windows-CD-Laufwerke** Virtuelle optische Laufwerke können sowohl im Host-System als auch in der Nutzungsumgebung von Bedeutung sein. Im Host-System erlauben sie den Zugriff auf einzelne Dateien des Datenträgers. Da die Zahl der in einem Emulator oder Virtualisierer konfigurierbaren virtuellen Laufwerke beschränkt ist, kann es günstiger sein, CD-ROMs oder DVDs innerhalb einer Nutzungsumgebung direkt einzubinden. Die Images liegen dann in der Datei der virtuellen Festplatte, die entsprechend größer wird. Das hat zudem den Vorteil, dass die Zahl von Dateien in einem Paket (Abschnitt 6.6) sich verkleinert und für den Nutzer die unter Umständen aufwändige Bedienung des virtuellen optischen Laufwerks entfällt.

Die Entsprechung des "Loopback Mounts" unter Linux findet sich in Windows-Umgebungen mit speziellen Tool-Sets, die einerseits eine grafische Oberfläche zum Ein- und Aushängen von ISO-Images bereitstellen und andererseits die notwendigen Treiber installieren, um dem Betriebssystem weitere optische Laufwerke vorzugaukeln. Ein einfaches Tool, das für Tests zum Einsatz kam, war CDSpace5.<sup>47</sup> Es zeigt die klassischen Probleme, die auftreten, wenn Applikationen mit Schutzmaßnahmen arbeiten. Etliche Spiele wie das bekannte Command & Conquer 2 lassen sich auf diese Weise nicht nutzen. Die Einbindung desselben Images via Virtualisierer funktionierte hingegen ohne Probleme.<sup>48</sup>

Ein weiteres Programm dieser Klasse ist VaporCD<sup>49</sup>, welches unter der GPL steht. Es ist jedoch seit 2002 nicht mehr aktualisiert worden.

Eine sehr gute Alternative besteht in den "Daemon-Tools".<sup>50</sup> Diese Werkzeugsammlung steht für eine breite Palette von Windows-Betriebssystemen zur Verfügung und kann mit den gängigen Schutzmechanismen, wie Safedisc, Securom, Laserlock oder RMPS umgehen. Wie auch bei den vorgenannten Tools lässt sich mehr als ein virtuelles optisches Laufwerk definieren.

### 7.10.6 Virtuelle Festplatten

Anders als Disketten oder optische Wechseldatenträger zählten Festplatten nicht zu den traditionellen Installationsmedien. Computergeschichtlich wurden sie nach der Magnetdiskette eingeführt (Abschnitt A.5.2).

<sup>47</sup>Produkt-Homepage - [http://en.cdspace.com/product/product\\_01.asp](http://en.cdspace.com/product/product_01.asp). Dieses Werkzeug ist Shareware und 15 Tage lang kostenlos testbar.

<sup>48</sup>Damit ist nicht die Langzeitarchivierung des Objekts generell gefährdet, jedoch würde der Einsatz von virtuellen CDs innerhalb der Nutzungsumgebung die thematische Zusammenfassung (Abschnitt 6.6.2) und die Bedienung durch den Archivbenutzer vereinfachen.

<sup>49</sup>Homepage - <http://vaporcd.sourceforge.net>.

<sup>50</sup>Produkt-Homepage - <http://daemon-tools.cc>, kostenlose und kostenpflichtige Versionen sind erhältlich.

Da etliche dieser Datenträger inzwischen ein für digitale Medien recht hohes Alter erreicht haben, liegt der kritischste Punkt oft schon im Auslesen vom ursprünglichen Datenträger.<sup>51</sup> Neben der Alterung des Datenträgers selbst, verschwinden die Laufwerke für bestimmte Datenträgertypen vom Markt. Darüber hinaus sind die Peripherieschnittstellen zum Anschluß verschiedener Laufwerkstypen vom technologischen Wandel betroffen.<sup>52</sup>

Ebenso wie für die blockorientierten Disketten und optischen Laufwerke erlaubt das Werkzeug "dd" Komplettabzüge von einer physischen Festplatte vorzunehmen. Ein solcher Abzug kann direkt von QEMU genutzt werden, wenn die passende Treiberausstattung innerhalb der Nutzungsumgebung vorliegt.<sup>53</sup> Da immer der gesamte Datenträger oder zumindest eine Partition gelesen werden muss, kann das Verhältnis von Nutzdaten zum gesamten benötigten Speicherplatz ungünstig ausfallen, was sich jedoch mit zunehmendem zeitlichen Abstand relativiert. Kommerzielle Virtualisierer stellen spezialisierte Tools zur Umwandlung physische Installationen in virtuelle Maschinen bereit. Ein solches Produkt ist der VMware Converter, der einige neuere Windows-Betriebssysteme, jedoch keinerlei Linux-Installationen, konvertieren kann.

Bei der Überführung einer Festplatte in eine Image-Datei entsteht ein "Stapel" von Blockdevices und Dateisystemen: Die Containerdatei bestehend aus virtuellem Blockdevice und Dateisystem der Nutzungsumgebung liegt auf einem realen Blockdevice im Dateisystem der Referenzumgebung. Man vergleiche hierzu den Abschnitt 6.8.4 im letzten Kapitel.

### 7.10.7 Problemstellungen

Die vorgestellten Programme arbeiten auf dem Host-System, sind also von dessen Veränderungen betroffen. Sie zählen in die Klasse der Hilfsprogramme der Referenzumgebung und müssen deshalb bei dessen Änderungen gegebenenfalls migriert werden. Die Anwendung dieser Werkzeuge aus einer emulierten oder virtualisierten Umgebung kann nur solange funktionieren, wie es möglich ist, reale Wechsellaufwerke in einem Zielsystem einzubinden.

Ein Problem des Ansatzes von Festplattenabzügen ergibt sich, wenn die Nutzungsumgebung auf der "neuen Hardware" der virtuellen Maschine wegen einer Lizenzbeschränkung, wie sie beispielsweise Microsoft für seine modernen Systeme wie XP und Vista vorsieht, nicht mehr startet. Images müssen also rechtzeitig erzeugt werden, um solche Nebenbedingungen behandeln zu können.

Die Verfügbarkeit von "Brennprogrammen" zum Erstellen von Abbildern hängt von der Existenz der Klasse optischer Datenträger ab. Bisher bot sich in diesem Bereich bis auf ein paar Irritationen bei der Einführung neuer Technologien ein recht erfreuliches Bild: Aktuelle Lesegeräte für neue Datenträgertypen sind rückwärtskompatibel und in der Lage ältere Medien zu lesen. Dieses gilt sowohl für die Klasse der DVD als auch der Nachfolgeprodukte Blu-ray-

<sup>51</sup>Vgl. hierzu die Anmerkungen in Abschnitt 2.4.4 zu Datenträgern.

<sup>52</sup>So bekommt man immer noch eine Reihe von Mainboards im Handel, die über parallele IDE- und die typischen 34-poligen Diskettenlaufwerksschnittstellen verfügen. Jedoch sind diese in neueren Kompakt- und Business-PC aus Kostengründen bereits gestrichen worden.

<sup>53</sup>Dieses kann im laufenden Betrieb beispielsweise für Linux-Installationen erfolgen. Betriebssysteme, die kein "dd" kennen, können im Offline-Modus durch dieses behandelt werden. So kann eine QEMU-Umgebung quasi extern vorbereitet und dann erst angestartet werden. Je nach Art des Abzuges ist eine Wiederherstellung des virtuellen Master Boot Records (MBR) durch die Werkzeuge des jeweiligen Betriebssystems notwendig.

Disks.

Neben der Übernahme der Installationsdatenträger in ihrer virtuellen Form als Image-Datei kann es sinnvoll sein, Dateien unabhängig vom logischen Aufbau des Datenträgers abzulegen. Das setzt jedoch ein anderes Verfahren zum Auslesen voraus.

Generell hatten Disketten im Verhältnis zu ihrer Speicherkapazität viel Platz, um Beschriftungen, mithin Metadaten anzubringen. Dieses gilt in ähnlicher Form für industriell bedruckte CDs. So wurden beispielsweise Installationshinweise oder Lizenzschlüssel direkt auf dem Datenträger angegeben. Diese Informationen sind bei der Einstellung von Datenträgern ins Archiv geeignet zu übertragen und in einem passenden Metadaten-set abzulegen.

Ähnlich liegt die Fragestellung für Dateinamen und die Wahl von Verzeichnissen. Während auf einzelnen Datenträgern aufgrund des begrenzten Platzes immer nur eine bestimmte Anzahl von Dateien unterzubringen war, so gilt diese Restriktion nicht mehr für das Softwarearchiv. Während im Langzeitarchiv die Dateinamen der digitalen Objekte keine Rolle spielen, kann dieses in Zwischenstufen, wie virtuellen Installationsdatenträgern wieder von Bedeutung sein, da Endanwender direkt mit diesen zu tun bekommen könnten.

## 7.11 Zugriffssteuerung und Rechtemanagement

Ebenso wie viele klassische Objekte sind digitale Objekte das Ergebnis eines menschlichen Schöpfungsprozesses und damit wie ihre analogen Pendanten im Augenblick der Erstellung mit bestimmten Schutzrechten ausgestattet. Sie sind deshalb gleichfalls als geistiges Eigentum zu betrachten.

Schutzrechte gelten üblicherweise für einen bestimmten begrenzten Zeitraum. Klassische Schutzrechte sind je nach nationalem Rechtssystem auf sehr viele Jahre angelegt.<sup>54</sup> Sie überschreiten damit weit den intendierten Lebenszyklus eines durchschnittlichen digitalen Objekts. Das kann bedeuten, dass zum Zeitpunkt des Ablaufs der Rechte und einem unproblematischen Zugriff seitens späterer Nutzer keine Technik oder kein Wissen mehr hierfür 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 ausgestiegen ist.

Anders als bei analogen Objekten, wo die Vervielfältigung mit einem unter Umständen auch hohen Aufwand verbunden ist und bei jedem Vervielfältigungsschritt mit einem graduellen Qualitätsverlust zu rechnen ist, lassen sich digitale Objekte sehr einfach unter Beibehaltung all ihrer Eigenschaften duplizieren. Die Problematik trifft sowohl auf Primär- als auch Sekundärobjekte zu. Eine gute Zusammenstellung des Problemfeldes Langzeitarchivierung und Recht gibt [Beger 2005].

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. Dieses geschieht typischerweise mittels Kopierschutzverfahren oder durch eine später notwendige "Freischaltung" der Software.

<sup>54</sup>Beispiele für Schutzfristen: 70 Jahre, 99 Jahre, siehe [Goebel und Scheller 2004], [Beger 2005] S. 77.

In den überwiegenden Fällen unproblematisch erweist sich die Verwendung von Open Source Software, die normalerweise unter der GPL<sup>55</sup> oder ähnlichen Lizenzmodellen angeboten wird. Das bedeutet einerseits die kostenfreie Verwendung in beliebigen Installationen und andererseits den vollständigen Zugriff auf den Source-Code. Damit besteht das Recht, die Software anzupassen, wenn diese Änderungen zurückfließen. Solche Lizenzmodelle sind am ehesten für eine Langzeitarchivierung und damit verbundene Standards geeignet.<sup>56</sup>

### 7.11.1 Schutzvorrichtungen auf Datenträgerebene

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:

- Installationsdatenträger werden üblicherweise einmal benötigt, um eine bestimmte Software auf einen Rechner aufzubringen.
- Laufzeitdatenträger werden während des Ablaufs der jeweiligen Applikation auf ihr Vorhandensein überprüft und es werden von ihnen diverse Laufzeitdaten nachgeladen.

Beide sind Verteilmedien, mit denen Hersteller ihre Software in den Umlauf bringen. Sie machen aus der körperlosen Software ein klassisches "verkauft" physisches Objekt.<sup>57</sup>

Schutzvorrichtungen an Datenträgern sollen verhindern, dass das Medium einfach digital kopiert werden kann. Das bedeutet, dass sich kein Bitstrom vom Datenträger auslesen lässt, der sich in bitweise identischer Form auf einen typgleichen Datenträger übertragen lässt. Hierbei kommen oft fragwürdige Verfahren zum Einsatz, die sich Abseits von definierten Standards bewegen. Ein klassisches Beispiel hierfür sind kopiergeschützte Audio-CDs, die nicht mehr dem CD-Standard<sup>58</sup> entsprechen. Ähnliches versuchen Computerspielerhersteller mit scheinbar defekten oder speziellen leeren Dateien auf Datenträgern, auf deren Existenz geprüft wird.

Teilweise nutzen Kopierschutzverfahren nichtstandardkonforme Spezialfunktionen der Le-sehardware. Hierzu zählt die Verwendung sogenannter "Fat-" und "Fattracks" bei alten Commodore-Disketten.<sup>59</sup> Damit entsteht das Problem, dass Primärobjekte unter Umständen nicht adäquat gesichert und in ein digitales Langzeitarchiv überführt werden können. Der Aufwand für diese Objekte ein Hardwaremuseum betreiben zu müssen, ist für die meisten Gedächtnisorganisationen zu hoch. Das Unterfangen muss als riskant und nicht für langfristige Horizonte geeignet eingestuft werden.<sup>60</sup>

<sup>55</sup>General Public License, siehe hierzu <http://www.gnu.org/licenses/gpl-2.0.html>.

<sup>56</sup>Dioscuri und PRONOM nutzen schon ein solches offenes Lizenzmodell. [Coy 2006] spricht sich ebenfalls dafür aus.

<sup>57</sup>In Zeiten schneller Datennetze nimmt die Bedeutung dieser Medien zum Verteilen von Software ab. Sie haben oftmals eine hohe psychologische Bedeutung und werden mit Packung und Handbuch zu einem manifesteren Besitztitel.

<sup>58</sup>Phillips/Sony Redbook, vgl. zudem <http://www.research.philips.com/newscenter/dossier/optrec/beeethoven.html>; "kaputte" Rootkit-CDs von Sony/BMG - <http://www.heise.de/newsticker/meldung/65602> und etliche weitere Meldungen auf [HZV 2007], eine kurze Einordnung und weitere problematische Seiten des Konzepts erörtert [Tsai u. a. 2006] S. 11, "The Sony Rootkit Controversy".

<sup>59</sup>Siehe hierzu das C64 Preservation Project - <http://c64preservation.com/dp.php?pg=protection>.

<sup>60</sup>Vgl. hierzu Abschnitt 2.5 zu Beginn dieser Arbeit.



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 und Herstellern auszutarieren sind: Verliert ein Anwender mit dem Unbrauchbarwerden des Datenträgers das Recht ein Produkt weiter nutzen zu können oder steht der Hersteller zumindest theoretisch in der Pflicht im Rahmen sinnvoller Grenzen für eine unbefristete Nutzbarkeit seiner Produkte heranziehbar zu sein.

### 7.11.2 Schutzvorrichtung an Software und statischen Objekten

In der Diskussion der letzten Jahre befanden sich in erster Linie die Verfahren zum Schutz von Audio- und Videodaten der Content-Anbieter.<sup>61</sup> Der Schutz an Primärobjekten ist für diese Arbeit insofern von Interesse, als dass es nur für wenige Schutzmechanismen gelingt, die notwendigen Komponenten mitzuarchivieren. Ein entsprechendes Problem trat beim Versuch auf, das ISO-Image des Spiels "Myst" zu virtualisieren (Abschnitt 7.10.5).

Da die meisten modernen Verfahren mit online verifizierten und aktualisierten Schlüsseln arbeiten, eignen sich diese nicht für eine Langzeitarchivierung. Deshalb sollten Primärobjekte generell ohne einen Zugriffsschutz ins Langzeitarchiv eingestellt werden.

Eine direkte Herausforderung stellen geschützte Anwendungsprogramme dar. Hier muss ebenfalls sichergestellt sein, dass sie zu einem beliebigen späteren Zeitpunkt erfolgreich ausgeführt werden können, wenn ihre Ablaufumgebung wiederhergestellt wurde. Neben der Freiheit von allen Online-Schutzkomponenten müssen die Metadaten die notwendigen 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.

### 7.11.3 Rechtemanagement im Softwarearchiv

Jegliche Schutzvorrichtungen stellen für ein digitales Langzeitarchiv ein nicht unerhebliches Problem dar. Kopierschutzvorrichtungen behindern die Übernahme der Inhalte für eine langfristige Erhaltung. Sie sind nach ihrer Entsicherung oder Überwindung jedoch keine langfristige Schwierigkeit im Archiv. Schutzverfahren am Objekt selbst sind wie eben dargestellt eine größere Herausforderung: Diese Schutzvorrichtungen bergen eine große Gefahr für den Archivbestand, da sie mit jedem Kopier- und Auffrischungsschritt des Archivs mittransportiert werden. Die folgenden Lösungen wurden in diesem Umfeld bereits diskutiert oder sollten dafür in Betracht gezogen werden:

- Bruch der jeweiligen Schutzvorrichtung im Augenblick der Archivaufnahme als Bestandteil der Arbeitsabläufe rund um das SIP. Ein Ansatz hierfür bietet die gesetzliche Lösung in den USA als Bestandteil des "Digital Millenium Act". Trotzdem handelt es sich hier um eine noch auszulotende Grauzone, da Verfahren zum Bruch von Kopierschutzeinrichtungen rechtlich problematisch sind und Archive sich verbotener Techniken

---

<sup>61</sup>Schutz von Musikstücken durch Schutzverfahren direkt am Objekt, wie beispielsweise durch Microsofts Multimediaformate oder die Absicherung hochauflösenden Videomaterials durch den HDMI-Standard vom digital übertragenen Video Broadcast bis auf das hochauflösende Display.

bedienen müssten. Umgekehrt könnte das Wissen um bestimmte Kopierschutztechniken zu einem deutlich späteren Zeitpunkt nicht mehr verfügbar und damit auch für Archive unerreichbar sein.

- Eine denkbare, aber eher unbefriedigende Option wäre die Reduktion digitaler Schutzfristen: Es bestehen aber für verschiedene Softwaretitel sehr unterschiedliche "kommerzielle" Haltbarkeiten. So sind einige Computerspiele besonders erfolgreich und ein Revival erfolgt nach 25 Jahren auf aktuellen Plattformen, wie Mobiltelefonen. Andere besitzen nur wenige Jahre Relevanz und wurden oft nur noch einen kurzen Recyclingzyklus lang in Spielesammlungen verramscht. Ähnliches gilt für Anwendungsprogramme und ihre veralteten Vorgängerversionen.
- Eine befriedigende Lösung könnte in einer gesetzlichen Vorschrift die Zurverfügungstellung schutzfreier Objekte zur Archivierung bestehen: Es gibt mit der Pflichtabgabe von Druckwerken und Musikmedien an die Nationalbibliothek<sup>62</sup> bereits Beispiele für Verfahrensweisen. So könnte diese Pflichtabgabe um die Forderung nach Freiheit von Schutzmechanismen erweitert werden.

Der Nestor-Aufsatz [Goebel und Scheller 2004] bezeichnet die folgenden Fragestellungen als urheberrechtlich relevant:

- Herstellung einer Archivkopie
- Recht zur Durchführung verändernder Eingriffe (Datenmigration),
- Aufhebung von Einschränkungen durch DRM (Kopierschutz, Individualisierung auf Einzelnutzer)
- Konditionen des Zugriffs auf die archivierten Ressourcen.

Jede Art von Rechtemanagement sollte deshalb auf der Ebene der Metadaten stattfinden. So definiert bereits das Dublin Core Metadatenschema, welches in Abschnitt 3.3.1 vorgestellt wurde, schon etliche Felder für die Beschreibung diverser Aspekte der Rechteausszeichnung und -festlegung. Aus Sicht des Archivbetriebs könnte die Umsetzung der Zugriffsrechte als eigene (virtuelle) Schicht im Archiv realisiert werden (Abb. 7.11).

Ein interessanter Randaspekt besteht in der Problematik kommerzieller Betriebssystemtreiber für virtuelle Hardware: So verwendet man Software, die nur im Zusammenhang mit dem Besitz einer bestimmten Hardware sinnvoll war. Jedoch kommen mit der Virtualisierung nun für die kommerziellen Betriebssysteme Grafik-, Audio- und Netzwerkkartentreiber zum Einsatz, ohne dass der Hersteller auf dem Wege des Hardwareverkaufs eine Abgeltung erhalten hätte.<sup>63</sup>

<sup>62</sup>In Deutschland im Gesetz zur DNB geregelt, siehe hierzu [DNB 2006]. Die Ausweitung auf digitale Objekte bringt eine jedoch eine Reihe neuer Problemstellungen mit sich. Siehe hierzu beispielsweise [Muir 2001].

<sup>63</sup>Vielfach haben die betroffenen Institutionen ja tatsächlich die reale Hardware selbst mal beschafft und für eine gewisse Zeitspanne genutzt, so dass sich die Fragestellung auf einen geeigneten Nachweis reduzieren ließe, von Mengenfragen vielleicht einmal abgesehen.

## 8 Fazit

Der Einsatz der Langzeitarchivierungsstrategie "Emulation" erreicht, fast beliebige digitale Objekte über lange Zeiträume zu bewahren. Am erfolgversprechendsten ist derzeit 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, sind 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" ein Toolset 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 und Bedienungsanleitungen oder Lizenzschlüssel vorhalten muss.

Wegen ihrer Eignung für dynamische Objekte kann die Emulationsstrategie den Belangen von Museen, Galerien und Ausstellungen dienen. Wenn diese beispielsweise Computerspiele, typische Vertreter der Klasse interaktiver dynamischer Objekte, als Teil des "Cultural Heritage" präsentieren wollen, müssen sie nicht ein zunehmend schwerer zu erhaltendes und aufwändig zu wartendes Hardwaremuseum vorhalten. Gleichzeitig werden eventuell besonders wertvolle oder seltene Objekte einer solchen Sammlung nicht der direkten Benutzerinteraktion und einem dadurch erhöhten Risiko und Verschleiß ausgesetzt. Selbst unter dieser Einschränkung kann der Benutzer einen guten Einblick erhalten, wie Spiele einer Epoche ausgesehen hatten. Zudem können Kunstwerke, wie spezielle Animationen, die objektbedingt an eine bestimmte Rechnerhardware gebunden waren, von dieser gelöst weiterhin gezeigt werden.

Trotzdem kann die Emulation die Migration nicht ersetzen, sondern sie stattdessen sinn-

voll 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.

**OAIS** Das Referenzmodell erweist sich mit der Unterstützung seitens einer Emulationskomponente durch seine offene und erweiterbare Anlage als tragfähiger Standard für ein digitales Langzeitarchiv. In dieser Arbeit wurde untersucht, welche Anforderungen hierzu an das Archivmanagement und Preservation Planning gestellt werden sollten. Zudem wurden Ideen präsentiert, wie Archivbenutzer geeignet auf ihre angeforderten DIPs zugreifen können. Dazu erfolgte eine Erweiterung des View-Paths als flexibler und möglichst redundanter Weg vom DIP in die jeweils gültige Referenzumgebung.

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 Bestimmungsprozesse spielen einerseits geeignete Format-Registries und andererseits Metadaten-schemata zur Aufnahme der zusätzlich 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. Diese zusätzlichen Informationen sollten geeignet mit dem Primärobjekt des Softwarearchivs verknüpft werden. Da die Bewahrung des digitalen kulturellen Erbes nicht mehr lediglich auf der Ebene einzelner Gedächtnisorganisationen oder im nationalen Kontext umfassend gelöst werden kann, sind Offenheit, Standardisierung und Transformierbarkeit von Metadaten von nicht zu vernachlässigender Bedeutung. Hierbei lassen sich die Auswahlkriterienkataloge durch Anwendung und Rückkopplung gemeinsam verfeinern.

Ähnliches gilt für Formatregister: Sie sollten, wie es beispielsweise PRONOM demonstriert, als offene Standards und Schnittstellen angelegt sein, die von einer breiten Benutzergemeinschaft getragen, weiterentwickelt und genutzt werden. Metriken können zur Bewertung von Wahrscheinlichkeiten des späteren Zugriffs auf unterschiedliche Objekte helfen und die Güte bestimmter View-Paths unter verschiedenen Gesichtspunkten bestimmen.

**Emulatoren** sind das Herzstück der vorgestellten Langzeitarchivierungsstrategien. Sie übernehmen die Brückenfunktion zwischen der historischen Nutzungsumgebung des gewünschten Primärobjekts und der jeweils aktuellen Referenzumgebung. Ihre Aufgabe ist dabei die Imitation der echten Maschine. Diese kann jedoch nie ganz vollständig sein, da sich bestimmte Peripherie, Datenträger oder auch der haptische Eindruck nicht sinnvoll nachbilden lassen.

Viele Emulatoren wurden von engagierten Einzelpersonen oder durch "Open Source Project Communities" entwickelt, die nicht primär das Ziel einer funktionalen Langzeitarchivierung im Auge hatten. Hier sollten Gedächtnisorganisationen versuchen, diese Gruppen und Einzelpersonen stärker zu integrieren. Die populäre X86-Plattform hat einige Firmen an den Markt gebracht, die Virtualisierer anbieten. Diese Programme eignen sich nicht unbedingt für eine sehr langfristige Strategie, erlauben jedoch einerseits das Studium von Problemen

der Weiterentwicklung im Zeitablauf und können andererseits bereits für zeitlich beschränkte Archivierungspflichten kleinerer Institutionen ausreichen.

Emulatoren sind ihrerseits Applikationen, die für eine definierte Nutzungsumgebung erstellt wurden. Sie unterliegen damit denselben Problemen wie alle digitalen Objekte. Die Ansprüche an Integrität und Authentizität werden jedoch nur an die "innere" Hälfte gestellt. Die emulierte Umgebung darf sich bei einer Anpassung nicht ändern. Hingegen kann es notwendig werden, ihre äußere Hälfte mit der jeweils aktuellen Referenzumgebung mitzuführen. Dieses ist an bestimmte Voraussetzungen, wie die Existenz geeigneter Programmiersprachen und Compiler geknüpft. Eine andere Strategie könnte in der Ineinanderschachtelung von Emulatoren liegen. Ältere Emulatoren werden mit ihrer Nutzungsumgebung ihrerseits in einem Emulator ausgeführt. Während Performance-Erwägungen sicherlich nicht dagegen stehen, kann jedoch der Transport von Primärobjekten in die innerste Umgebung und die Benutzerinteraktion vor neuen Herausforderungen stehen.

Deshalb werden seit einiger Zeit weitergehende Konzepte erforscht. Hierzu zählen modulare Emulationsansätze oder die Universal Virtual Machine. Das langfristige Ziel ist sicherlich eine wohl-definierte "emulation virtual machine" Plattform, die für jede zukünftige Rechnerarchitektur implementiert wird. Damit kann erreicht werden, dass alle bis dahin geschaffenen Emulatoren für diese Plattform "bis in die Ewigkeit" weiterbenutzt werden können. Die Forschungen auf diesem Gebiet haben bereits Prototypen hervorgebracht, die jedoch noch nicht den funktionalen Umfang ihrer traditionellen Pendants erreicht haben. Selbst bei Verfügbarkeit eines dedizierten modularen Langzeitemulators sollte es weiterhin im Sinne redundant ausgelegter View-Paths sein, zusätzlich traditionelle Emulatoren bei geeigneter Verfügbarkeit weiter zu erhalten.

An dieser Stelle stellt sich die Frage, wie eine bessere Reaktion auf zukünftige Entwicklungen erreicht werden könnte: Hierzu könnten beispielsweise Verpflichtungen der Hardwarehersteller stehen, Emulatoren für ihre Plattform OAIS-Archiven zur Verfügung zu stellen oder Komponenten für die UVM zu liefern. Auf seiten der Betriebssystemproduzenten könnten Vereinbarungen zu generischen Schnittstellen für Netzwerk, Audio oder Grafiksysteme stehen, die abstrakte, von konkreter Hardware unabhängige Ansätze liefern, die einfacher und leistungsfähiger im Emulator abbildbar sind.

**View-Paths** formalisieren die virtuelle Verbindung von einem bestimmten Primärobjekt zu einer Referenzumgebung, in der je nach Typ ein Zugriff, Abspielen oder Ansehen erfolgt. Durch sie lassen sich eine Reihe von Aufgaben des Archivmanagements bestimmen und aus ihnen Prognosen zu einzelnen Objekttypen für den Preservation Manager ableiten. View-Paths werden auf diese Weise zu einer zentralen Entscheidungsgröße für die Bewertung des Langzeitzugriffs auf archivierte Primärobjekte. Wegen der Vielfalt der digitalen Objekte kann es für die Verbesserung des Darstellungsprozesses sinnvoll sein eine Rückkopplung zu den Archivanwendern einzubeziehen. Sie könnten Anhaltspunkte für die Ausgestaltung von Metriken liefern, die helfen automatisiert eine optimale Nutzungsumgebung wiederherzustellen.

**Softwarearchiv** schafft die Voraussetzung für die Generierung von View-Paths zur Anzeige oder zum Ablauf von Primärobjekten. Es ergänzt die archivierten Emulatoren um notwendige Komponenten, wie Betriebssysteme, Hilfskomponenten und Applikationen zur erfolgreichen

Wiederherstellung von Nutzungsumgebungen. Das Archiv besteht aus statischen und dynamischen Komponenten: Einmal eingestellte Programme und Betriebssysteme verändern sich nicht mehr. Das gilt im Besonderen, wenn sie historisch geworden sind. Das Archiv muss schrittweise um jene Applikationen erweitert werden, die zur Unterstützung neuer DIPs notwendig werden. Hier könnte analog zur Lieferpflicht bei Büchern eine solche für alle relevante Software erwogen werden.

Einer stärkeren Beobachtung müssen die Emulatoren unterliegen. Sie zählen je nach Art ihrer Langzeitarchivierungsstrategie zu den dynamischen Komponenten. Hier ist dafür zu sorgen, dass die einmal definierten View-Paths nicht mit dem Obsolet-Werden eines bestimmten Emulators abreißen. Deshalb sollte im Sinne einer hohen Redundanz dafür gesorgt werden, dass für als "wichtig" eingestufte Rechnerplattformen und Gruppen von Primärobjekten mehrere verschiedene Emulatoren für verschiedene Host-Plattformen archiviert werden.

Das in dieser Arbeit vorgeschlagene Softwarearchiv ist zu großen Teilen im Sinne des OAIS kein allgemein frei zugreifbares System. Ein überwiegender Teil der eingestellten Software, wie Applikationen und Betriebssysteme unterliegen einer proprietären Lizenz. Diese erlaubt nur einem wohldefinierten Archiv-Benutzerkreis den Zugriff. Dieser lässt sich beispielsweise mittels in den Räumen der beteiligten Gedächtnisorganisationen aufgestellten Reference Workstations realisieren. Diese Systeme könnten beispielsweise zudem in schon bestehende Online-Recherche-Maschinen integriert werden.

Die Referenzumgebungen werden in den meisten Fällen auf vorbereitete View-Paths zurückgreifen, die als Bestandteil des Softwarearchivs zur Vereinfachung des Zugriffs seitens der Endbenutzer vorgehalten werden können. Sie veralten mit ihrer Bezugsumgebung. Sie lassen sich unter bestimmten Umständen auf neue Umgebungen migrieren, jedoch gelingt dieses am ehesten beim Übergang auf eine neue Version desselben Anbieters und nur bedingt beim Wechsel auf andere Projekte oder Hersteller.

An dieser Stelle wären Untersuchungen anzustellen, welche Möglichkeiten es gibt, analog zu einer Katalogrecherche über ein Weitverkehrsnetz, den Zugriff auf das Softwarearchiv in einer Weise zu erlauben, der die Rechte wahrt. Hier gibt es sicherlich schon Erfahrungen im Umgang mit E-Journals etc. was den Zugriff anbetrifft. Jedoch sind gerade für das Interface, welches dem Archiv-Benutzer präsentiert wird, noch einige Überlegungen vorzunehmen.

indexAnforderung

**Neue Herausforderungen** Vertrauenswürdige digitale Langzeitarchive bringen neuartige Anforderungen an Gedächtnisorganisationen mit sich. Die Tätigkeiten und Fähigkeiten digitaler Archivare weichen zwangsläufig von traditionellen Aufgabenbeschreibungen ab: "New Skills Call for New Jobs".<sup>1</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.

<sup>1</sup>Vgl. [van der Werf 2002], S. 61. Eine Erhebung und kurzen Überblick zu Aufgabenbeschreibungen und zukünftigen Entwicklungen in diesem Gebiet liefert [Choi und Rasmussen 2006], zur Ausbildung und Studienprogrammen [Pomerantz u. a. 2007].

Ein großes Problem in der Langzeitarchivierung digitaler Objekte, sowohl der Primär- als auch Sekundärobjekte kann das Digital Rights Management bedeuten. 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 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.

Digitale Langzeitarchive beginnen mit dem Aufbau eigener Communities, wie die inzwischen beeindruckende Liste nationaler, europäischer und internationaler Initiativen zeigt. Gemeinsame Anstrengungen können helfen, die Kosten des Betriebs der eigenen Archive zu beherrschen. Neben spezialisierten Entwicklern und Anwendergruppen sollten zudem Endbenutzer einbezogen werden, um durch Rückkopplungsprozesse die Qualität der Darstellung und der angebotenen View-Paths zu verifizieren und zu verbessern. Verteilte Ansätze über die Grenzen einzelner Institutionen hinweg erlauben notwendige Spezialisierungen, ohne die die schiere Menge verschiedener Objekttypen nicht dauerhaft zu beherrschen sein wird.

Diese Arbeit hat einige theoretische und praktische Grundlagen diskutiert, welche die Nachhaltigkeit digitaler Objekte verbessern helfen. An diese sollten sich weitere Forschungen und Aktivitäten anschließen: Hierzu zählen sicherlich die Automatisierung des Erstellens von View-Paths, geeignete Benutzerinteraktionen bei steigendem zeitlichen Abstand zwischen historischer und aktueller Nutzungsumgebung und die Integration in Frameworks.

Die voranschreitende Virtualisierung, die in vielen Bereichen große Berührungspunkte zur Emulation aufweist, sorgt sicherlich für eine größere Popularität und besseres Wissen um Konzepte zur Langzeitarchivierung. Trotzdem bleibt die Frage, wie neuere Paradigmen, wie stark vernetzte Webservices, Agenten und spezialisierte Geräte in Zukunft langzeitarchiviert werden können. Sie sind ähnlich zu den Inhalten des World Wide Web ein "Moving Target" und damit nicht mehr trivial innerhalb einer klassischen Hardwareemulation abzubilden. Wenn auch einige Emulatoren bereits erlauben, ganze virtuelle Netzwerke in Software nachzubilden, so kann dieses auch wegen der höheren Komplexität und Dynamik nicht mehr ausreichen.

**Weitere Aspekte** Unabhängig von den bisherigen Überlegungen existieren kommerzielle Entwicklungen, die älteren dynamischen Objekten ein "Zweites Leben" bescheren können. So eignen sich beispielsweise Emulatoren dazu, alte Spiele auf ganz neuen Plattformen, wie Mobiltelefonen, portablen Spielekonsolen oder Set-Top-Boxen "weiterleben" zu lassen. Sie stellen keine hohen Ansprüche an Rechenleistung oder Grafikausgabe und bieten trotzdem kurzweilige Unterhaltung. Dieses geht soweit, dass kleine Hardwaremodule erhältlich sind, die eine ganze Auswahl von Spielen eines Home-Computers, wie des Commodore 64 bündeln und die simpel über Joystick am heimischen Fernseher zu betreiben sind. Darüber hinaus könnten Emulatoren Softwareherstellern die Möglichkeit bieten, den Übergang von einem Betriebssystem auf ein neues zu gestalten, ohne im neuen Kompromisse zur Rückwärtskompatibilität eingehen zu müssen. Generell werden für zukünftige Rechnerplattformen Überlegungen angestellt, durch Virtualisierung Betriebssysteme sicherer und Maschinen besser wartbar zu machen.

Das Engagement einiger größerer Internet-Dienstleister im Bereich Online-Storage wird



**Abbildung 8.1:** "World of Warcraft" ist eine neue Generation der Online-Spiele. Zwar läuft auf den leistungsfähigen Rechnern der Teilnehmer jeweils eine lokale Kopie des Spiels, die Interaktion findet aber auf zentralen Netzwerkservern statt.

vielleicht langfristig dazu führen, dass nicht nur die Bitströme der (zahlenden) Kundschaft aufbewahrt werden. Angebote werden in Zukunft vielleicht danach bewertet und genutzt werden, inwieweit sie Aussagen zu einer wirklichen Langzeitverfügbarkeit der hinterlegten Inhalte machen können.

## 8.1 Grenzen und neue Herausforderungen

Die Geschichte der Computerentwicklung begann mit wenigen (Groß-)Rechnern, an die zur besseren Auslastung des teuren Gerätes oft einige bis viele Arbeitsplätze angeschlossen waren. Trotzdem produzierten diese Maschinen aufgrund ihrer sehr geringen Zahl und sehr spezialisierten Einsatzbereiche keine signifikanten Mengen an langzeitarchivierungs-relevanten Objekten. Das begann sich erst mit einem Paradigmenwechsel Mitte der 1970er Jahre langsam zu ändern, als Rechner in Form von Home-Computern für den breiteren Massenmarkt



der Privatanwender aufkamen. Zeitgleich nahm die Zahl der in Unternehmen und öffentlichen Einrichtungen eingesetzten Maschinen bei Dezentralisierung zu. Das Paradigma relativ autonomer Workstations, Computer an jedem Arbeitsplatz, ist derzeit immer noch das vorherrschende Modell. Jedoch scheint sich mit der Durchsetzung des Internets ein erneuter Wechsel anzudeuten: Durch immer leistungsfähigere Datennetze zeichnet sich eine zunehmende Tendenz ab, Applikationen teilweise oder vollständig "ins Netz" zu verlagern.

Seit einigen Jahren wird eine völlig neue Generation der Multiplayer- und Online-Spiele entwickelt (Abb. 8.1), die ganz neue Herausforderungen an die digitale Langzeitarchivierung stellen, da sie das Konzept des "dynamischen digitalen Objekts" noch einmal erweitern.

Die Spielidee wird erst durch Interaktion vieler tausender Benutzer erreicht, die sich weltweit vernetzt auf zentralen Servern einfinden. Darüber hinaus entstehen weitere neue Formen digitaler Objekte, die nur bedingt in den Betrachtungen dieser Arbeit eine Rolle gespielt haben: Große, verteilte Applikationen wie Datenbanken, Grid- oder Cluster-Computing, ERP-Systeme, die über etliche Maschinen distribuiert arbeiten, fordern weitergehende Anstrengungen.

Wenn inzwischen der anschwellende Hype um neuartige Plattformen der Interaktion und Präsentation wie "Second Life" wieder abgeklungen ist, so zeigt diese Entwicklung zweierlei: Wohin sich zukünftige Benutzer-Interfaces und Referenzumgebungen entwickeln könnten und welche neuartigen Herausforderungen damit auf zukünftige Emulatorentwickler zukommen. Dieses gilt darüber hinaus für Datenein- und -ausgaben. Neben diesen Phänomenen entstanden ganz neue Formen des digitalen Gedankenaustauschs, wie Web-Logs<sup>2</sup> oder sogenannte Soziale Netzwerke.<sup>3</sup>

Das alles zeigt, dass sich auch mit einer digitalen Langzeitarchivierung das generelle Problem nach dem "was ist sinnvoller Weise alles zu archivieren" ebenfalls nicht lösen lässt. Zwar sind die Möglichkeiten zur Archivierung fast unbeschränkter Datenmengen scheinbar da, doch wird man auch weiterhin nicht um eine geeignete Festlegung von Sammelrichtlinien umhinkommen, da man andernfalls in den Datenfluten zu ertrinken droht.

---

<sup>2</sup>Das sogenannte Blogging ist ein Massenphänomen seit Beginn der 2000er Jahre, ist jedoch eher im Bereich des Web-Archivings einzuordnen, vgl. hierzu auch [Hank u. a. 2007].

<sup>3</sup>Dienste wie FaceBook, OpenBC/Xing, LinkedIn, StudiVZ sind sicherlich für spätere Forschergenerationen von Interesse. Sie fallen ebenso eher in den Bereich des Web-Archivings.



# Teil IV

## Anhang



# A Anhang

Der Anhang dient der Ergänzung und Untermauerung der in der Arbeit gemachten Aussagen. Der nächste Abschnitt gibt eine knappe Zusammenfassung der zu Beginn des Projekts durchgeführten Recherchen. Die große Anzahl von Emulatoren für verschiedene Rechnerarchitekturen zeigt die lange Fundierung des Emulationsansatzes. Viele der aufgelisteten Projekte sind nicht mehr existent oder der Code ist in aktuellen Referenzumgebungen nicht mehr ausführbar. Zudem sind viele dieser Programme im Laufe der Zeit überholt und durch andere Ansätze abgelöst worden. Darüber hinaus werden mit ReactOS oder ScummVM interessante Emulationsansätze verfolgt, die jedoch außerhalb des Fokus dieser Arbeit liegen.

Der darauffolgende Abschnitt beschäftigt sich mit einer direkten Anwendung des vorgestellten Softwarearchivs. Für die X86-Architektur erfolgt die (unvollständige) Zusammenfassung von Installationsdatenträgern älterer Betriebssysteme zu einem universellen ISO-Image. Dieses kann von den verschiedenen Emulatoren und Virtualisierern unkompliziert als virtueller optischer Datenträger eingebunden werden.

Am Beispiel einer Auswahl von Computerspielen, als typische Vertreter interaktiver Applikationen, wird die Anwendung verschiedener View-Paths zur Ausführung und Benutzerpräsentation gezeigt und einige Grenzen diskutiert.

Es schließen sich technische Überblicksabschnitte zu gebräuchlichen Datenträgern neuerer Rechnerarchitekturen an. Diese kommen als Installationsdatenträger oder in abstrakter Form als virtuelle Laufwerke der diversen Emulatoren zum Einsatz.

Den Abschluss des Anhangs bildet ein Glossar mit einer kurzen Erklärung häufig gebrauchter Begriffe.

## A.1 Emulator-Recherche: Erste Ergebnisse

Eine erste Iteration der Suche nach bereits existierenden oder referenzierten Emulatoren führte die Arbeitsgruppe Langzeitarchivierung am Lehrstuhl bereits im Jahr 2004 durch. Hierbei fiel die erstaunlich große Zahl der Emulatoren für Home-Computer und alte Großrechner auf. Letztere waren besonders im wissenschaftlichen Umfeld im Einsatz. Das Vorhandensein vieler Emulatoren ist zum Teil dem Phänomen geschuldet, dass das Verschwinden von Maschinen und Rechnerarchitekturen nicht erst vor ein paar Jahren einsetzte, sondern die Computergeschichte seit ihrem Anbeginn begleitet.

Im Zuge der ersten Recherchen erfolgte eine Gruppenbildung von Rechnerklassen, wie sie in Abschnitt 4.2 dargestellt ist. In dieser Zusammenstellung kommen aufgrund ihres Alters

nur die Klasse der Großrechner, Home-Computer und Arcade-Maschinen vor.

Die folgende, unvollständige Liste liefert einen Überblick zu den verschiedenen Maschinen und den dazu erstellten Emulatoren. Nur in einigen Fällen konnte die reale Verfügbarkeit und Funktionsfähigkeit tatsächlich überprüft werden. Die meisten der genannten Programme verfügen über keine Homepage und sind nicht mehr erhältlich. Vielfach gingen sie in Nachfolgeprojekten auf oder wurden durch generischer angelegte Ansätze wie MESS/MAME [MESS 2008] abgelöst. Dieses stellt kein größeres Problem dar, sondern illustriert die Vitalität des Emulationsansatzes einerseits aber auch die zu bedenkenden Probleme der Obsoleszenz von Software andererseits.

### A.1.1 Großrechner

Sie sind die ältesten Computer. Das "groß" bezieht sich dabei eher auf die Baugröße, denn auf die Rechenleistung. Abbildung 4.2, S. 81 illustriert dieses sehr gut: Hier füllt ein Großrechner, dessen Leistungsfähigkeit von einem heutigen Mobiltelefon weit übertroffen wird, den halben Maschinsaal des Freiburger Universitätsrechenzentrums aus.

**Digital Equipment Corporation** - DEC spielte mit den PDP-Systemen und "VAXen" eine nicht unwesentliche Rolle am Markt der ersten kommerziell erfolgreichen Computer. Die PDP-1, 4, 7, 8, 9, 10, 11, 15 sowie VAX lassen sich mit Computer History Simulator nachbilden. Weiterhin existieren ein PDP 8/11 Emulator für Unix, PDP8/E Emulator für Mac-OS, PDP-8 Computer für JavaVM, TM PDP-8 für DOS und Unix mit X11, "Ersatz-11" für DOS.

**EDSAC** - kann durch den Warwick EDSAC Simulator für Mac-OS oder Windows 95 nachgebildet werden.

**Hewlett-Packard** - für die Maschinen HP-41 beziehungsweise HP-48 findet man den Emu48 für DOS, Windows 3.1, x48 für Unix mit X11; für die 2114, 2115, 2116, 2100, 21MX steht mit dem Computer History Simulator.<sup>1</sup>

**IBM** - die "International Business Machines" zählten mit den Maschinen 1401, 1620, 1130, 7090/7094, System 3 zu den dominanten Mitspielern im Marktsegment der Großcomputer. Software dieser Systeme kann durch den Computer History Simulator ausgeführt werden.

**MITS Altair** - konnte mittels Computer History Simulator oder dem Altair 8800 Emulator für Windows 3.1 emuliert werden.

**Scientific Data Systems SDS 940** - wird ebenfalls durch den Computer History Simulator nachgebildet.

**Universal Turing Machine** - gibt es als Turing für DOS; Turing Machine für Windows oder als Turing's World für Macintosh und Windows.

### A.1.2 Home-Computer

Sie hießen im damaligen Sprachgebrauch oft "Microcomputer", um sie in ihrer Größe und Leistungsfähigkeit von den Großrechnern abzugrenzen. Aus dem verbraucherfreundlichen Preis der Geräte folgte die schnelle Verbreitung dieser Plattform an Privatanwender, die das erste Mal

<sup>1</sup>Der Computer History Simulator, siehe [CHSP 2007] geht in die Anfänge der kommerziellen Computergeschichte der Großrechner zurück und emuliert populäre Maschinen dieser Zeit, siehe hierzu weiter [CHM 2008].

in größerem Stil die Chance bekamen, sich mit Computern zu befassen. Auf diesem dadurch mitbegründeten Enthusiasmus von Programmierern gründet die erste Welle der Emulatorentwicklungen.

Soweit ältere Emulatoren als Open Source oder im Quellcode verfügbar vorliegen, können sie als Beispiel und Anleitung zur Erstellung aktueller Programme dienen.

**ABC80** - ABC80 für DOS.

**Acorn Atom** - Acorn Atom Emulator für Unix(-ähnliche) mit grafischer X11-Oberfläche.

**Amiga** - UAE für Unix mit X11, Macintosh, DOS, NeXTstep, BeBox.

**Amstrad CPC** - A-CPC, Ami-CPC/PC-CPC, CPE für Amiga, DOS; !CPC, !CPC\_Demo, CPC-Emulator für Acorn Archimedes, CPC++ für SunOS, Mac-OS; CPCEMU, No\$CPC für DOS, EmuCPC für Amiga, Richard Wilson's CPC Emulator für DOS und Unix mit X11, Caprice32,<sup>2</sup> WinAPE.<sup>3</sup>

**Atari ST/TT** - Amtari, Chamaeleon II, Emplant, Medusa, ST4Amiga, STEmulator für Emulation auf der Amiga-Plattform; PaCifiST and STiMuL für DOS; STonX für Unix mit X11, DOS, Windows; TOS2WIN für Windows 95 und NT.

**BBC** - !6502Em, !65Host, BBCEm, für Acorn Archimedes; BBC, BBC386, BPC, TOL für DOS; BBC-Basic, Horizon/MacBeebEm für Mac-OS, BBC-Environment auf dem Atari ST; (X)Beeb für Linux / Unix; BeebEm für Unix, Windows NT, Windows 95; Owl für NeXTstep, THE EMULATOR für Amiga.

**Commodore 64/128** - JaC64, Java-basierter Emulator.

**CHIP8** - Chip8 und VChip-8 für DOS, S-CHIP für HP-48.

**Coleco Adam** - ADAMEm für DOS, Unix mit X11, Linux.

**Colour Genie** - CGenie and Colour Genie Emulator für DOS.

**CoCo 2, Dragon 32/64** - CoCo 2, PC Dragon II, T3 für DOS; Dream für Amiga.

**(frühe) Apple Macintosh** - A-Max, Emplant, ShapeShifter für Amiga, Aladin, Spectre für Atari ST, Basilisk für BeBox, MagicSac für Atari ST/TT.

**MSX** - AmiMSX für Amiga, MSX-1 Emulator für Atari ST, PC MSX-1, 2 Emulator für DOS; fMSX für Unix, PowerMac, DOS und Amiga.

**Oric** - Amoric für Amiga, Euphoric für DOS and Linux; Oric 48K für Unix mit X11.

**Philips P2000** - M2000 für DOS, Unix/Linux mit X11.

**R2000/SAM Coupe** - SimCoupe für Unix mit X11.

**Sinclair 1000/ZX81** - Extender, ts1000 für DOS, ZX81.PRg für Atari ST.

**Sinclair ZX/Spectrum** - !MZX, !Speccy, Speculator für Acorn Archimedes; Atari-Speccy für Atari; Elwro 800-3 Jr, JPP, SPECTRUM/VGASpec, SpecEM, Jarajevo, X128, Z80, ZX Spectrum-Emulator, ZX Spectrum für DOS; ZX Spectrum Emulator für Java; KGB, zx-spectrum, ZXAM, zxspec für Amiga; MacSpeccy, MacSpectacle für older Mac-OS; Power-Spectrum für PowerMac; Spectrum 48 für Commodore 64; Spectrum für Amiga; WSpecem für Windows; X 128 für Unix mit X11; xz80, xzx für Unix mit X11; xzx für Unix mit X11; !z80Em für Acorn; ZX SP für Atari; zxlin386 für Linux.

**Sinclair Z88** - Win Z88, Z88dream für Windows; Z88EM, TI-81 für DOS.

**TI-81** - TI-81 Emulator für Mac-OS.

**TO7** - FunzyTo7 für DOS, Unix mit X11, FunzyTo7-70 für DOS, Unix mit X11.

<sup>2</sup>Homepage auf Sourceforge - <http://sourceforge.net/projects/caprice32>, letzte Version 4.2.0 aus 2005.

<sup>3</sup>Homepage - <http://www.winape.net>, siehe auch [EMUU 2008].

**TRS-80 Models I-IV** - model1-d, TRS-80 Model I Emulator, TRS-80 Model III emulator, TRS-80 Model III simulator für DOS; trs80 für Amiga; trs80 für Mac-OS; Xtrs für Unix mit X11.

### A.1.3 Konsolen und Arcade Emulatoren

Spielhallencomputer, sogenannte Arcade-Systeme, sind eher interessant für technische Museen oder Computerspiele-Ausstellungen. Hier steht die konkrete Interaktion des Benutzers im Fokus. Dieser erwartet keine großen Herausforderungen ein Spiel zu starten. Gerade im Schnellstart ohne spezielle Computerkenntnisse lag der Reiz dieser Klasse von Systemen.

Es existiert ein recht generelles Arcade Machine Emulatorenverzeichnis "Arcade Emulation Repository Project" mit Centepede Emulator, Crazy Kong Emulator, Emu, Gyruss Emulator, Kong Emulator, Mr. Do Emulator, Pengo Arcade Emulator, Sparcade! für DOS; Asteroids Emulator für Power Mac; MacMoon, Space Invaders Emulator, Williams Digital Arcade für Mac-OS. Weiterhin zu nennen wären:<sup>4</sup> Cinematronics Emulator, Gauntlet Emulator, Gottlieb Emulator, MacMoon, Mr. Do Emulator für DOS, Namco's Museum of Games<sup>5</sup>, Pengo Arcade Emulator für DOS, Phoenix and Pleaides für Windows 95, Shinobi Emulator, Space Invaders Emulator für Mac-OS und die Williams Arcade Classics für DOS, Sony PlayStation, Windows 95.

**Atari 2600** - A26 für DOS; Activision Game Pack für Windows, Mac-OS; Atari 2600 Emulation Project für DOS, Unix mit X11, PCAE für DOS; Stella 96 für Unix, Linux mit X11, DOS, Windows 95, PowerMac; Virtual 2600/Virtual VCS für Unix mit X11, DOS; VCS2600 für DOS.

**Atari Jaguar** ColecoVision, ColEm für Unix mit X11, Mac-OS, PowerMac, DOS, Windows 95; Mission für DOS.

**GameBoy** - Fondle GameBoy Emulator für DOS; !GameBoy für Acorn; GBSIM, PCBOY für DOS; ToyBoy für Amiga; Virtual GameBoy für Unix mit X11, Windows, Amiga, Mac-OS, DOS, OS/2

**NES/Famicom** - iNES für Unix mit X11, PowerMac, Windows, Linux; LandyNES, NESA für DOS; PasoFami für Windows.

**SNES** - Emplant für Amiga; SPW, SNES 96 für Windows 95; SNES Professional, Virtual Magicom für DOS, XNES für Unix mit X11.

**Sega** - Sega Genesis; Emplant für Amiga; EmulatorX, GenEm, MegaDrive, für DOS; Kyoto für DOS, Linux, Mac-OS.

**Sega Master System/GameGear (SMS)** - Massage für DOS; MasterGear für Unix mit X11, DOS, Macintosh.

**Sony PlayStation (PSX)** - PSXMooSim für Amiga, Solaris.

<sup>4</sup>Diese Liste liefert nur einen Ausschnitt und hat keinen Anspruch auf Vollständigkeit. Ein Einstieg für weitere Recherchen wäre beispielsweise ist sicherlich das Google-Directory zu Emulatoren: <http://www.google.com/Top/Computers/Emulators>.

<sup>5</sup>Die Zusammenfassung populärer Spiele in Serien, siehe hierzu <http://www.mobygames.com/game-group/namco-museum-series>.



### A.1.4 Emulatorensuche zu sonstigen Ansätzen

Dieser Abschnitt stellt Emulationsansätze anderer Ebenen vor, die für die Hauptaussagen dieser Arbeit nicht weiter von Bedeutung sind. Neben den Hardwareemulatoren können jedoch weitere Ansätze interessant sein, um bestimmte digitale Objekte ablaufen zu lassen oder weitere Aspekte verschiedener Emulationsstrategien für die Langzeitarchivierung zu untersuchen.

Eine eigene Klasse bilden zudem Emulatoren als Bestandteile von Entwicklungsumgebungen, beispielsweise für Symbian OS basierte Mobiltelefone oder PDAs. Ebenso existieren Emulatoren für Sony PlayStation 3 oder andere Spielekonsolen.

**Betriebssystem-Emulator "Wine"** Ein Betriebssystememulator mit langer Entwicklungsgeschichte ist Wine ([Thorns 2007] Kapitel 4). Dieser Emulator stellt benötigte APIs für Windows-Programme auf Linux-X86-Systemen zur Verfügung. Eine noch weitergehende Windows-Nachprogrammierung wurde mit dem Projekt "ReactOS"<sup>6</sup> ins Leben gerufen. Diese Software bildet die Windows 32 bit-API derart nach, dass Windows-Treiber und Applikationen direkt genutzt werden können. Die Liste der ablauffähigen Applikationen ist inzwischen recht beachtlich. Umgekehrt stellt Cygwin ([Thorns 2007] Kapitel 5) Posix-Schnittstellen für Windows zur Verfügung.



**Abbildung A.1:** ScummVM nutzt die Kenntnis des Datenformats einer Reihe von Spieleserien und reimplementiert den Interpreter – sowohl für Grafik als auch Sound – für aktuelle Plattformen.

**Spiele-Interpreter ScummVM** Dieser Emulator setzt auf eine Abstraktionsschicht, die in einer Serie populärer Spiele des Herstellers Lucas Arts benutzt wurde. ScummVM ist eine Implementation des SCUMM (Script Creation Utility for Maniac Mansion) Interpreters, der für Spiele wie "Day of the Tentacle" oder die bekannte "Monkey Island"-Serie Verwendung

<sup>6</sup>Homepage - <http://www.reactos.org>.

fand. Für die Präsentation von Spielen wie: "Zak McCracken and the Alien Mindbenders", "Loom", "The Secret of Monkey Island", "Monkey Island 2: LeChuck's Revenge Indiana Jones and the Fate of Atlantis", "Sam & Max", "Simon the Sorcerer" ist der Interpreter ebenfalls geeignet (Abb. A.1).<sup>7</sup>

Die Entwicklung erfolgt durch eine aktive Fan-Gemeinde. So gibts inzwischen Portierungen für andere Spiele-Engines wie "Sierra AGI",<sup>8</sup> "Humongous Entertainment" und weitere. Der Vorteil dieses Ansatzes liegt darin, dass als Nutzungsumgebung aktuelle Plattformen wie Linux, Windows, Mac-OS X und diverse weitere Architekturen<sup>9</sup> in Frage kommen.

## A.2 Begleitende Experimente

Im Rahmen der Arbeitsgruppe zur funktionellen Langzeitarchivierung am Lehrstuhl für Kommunikationssysteme und im Zuge der Mitarbeit am PLANETS-Project<sup>10</sup> hauptsächlich in den Teilprojekten "Tools for Environments" und "Preservation Action Strategy Development" wurden etliche Experimente und Recherchen durchgeführt. Sie werden sukzessive im Rahmen der PLANETS-Kooperation [PLANETS 2008] dargestellt und sollen deshalb hier nur in einer kurzen Überblicksdarstellung berichtet werden.

Die Aktivitäten umfassten dabei unter anderem:

- die Recherche nach existierenden Emulatoren unterschiedlichster Herkunft, wie im vorangegangenen und Abschnitt 4.1 dargestellt.
- eine Reihe von Funktionstests und die Untersuchung der Eignung diverser Emulatoren für unterschiedliche Plattformen. In diesem Zuge wurde ebenfalls die Ineinanderschachtelung von Emulatoren untersucht. Das ist für einen geeigneten Weiterbetrieb alter Emulatoren, die nicht mehr für die jeweils aktuelle Referenzplattform verfügbar oder übersetzbar sind, relevant. Zudem wurden die Möglichkeiten der Parallelinstallation mehrerer verschiedener Virtualisierer und Emulatoren auf einer Referenzmaschine betrachtet.
- Parallelinstallation diverser Emulatoren in einem Host-System. Diese bietet Hinweise auf die Gestaltung geeigneter Referenzumgebungen. Für größere Benutzerumgebungen oder Remote-Dienste (Vgl. [Welte 2008]), wie sie in Bibliotheken oder Museen erwartet werden können, wird es interessant, inwieweit sich verschiedene Emulatoren (und Virtualisierungsumgebungen) auf einer Host-Maschine parallel betreiben lassen.

<sup>7</sup>Für einige Titel gilt eine Einschränkung auf bestimmte Ausgaben. Experimentell sind noch laut Projekt-Homepage folgende Titel spielbar: "Indiana Jones and the Last Crusade", "Full Throttle" und "The DIG". Auf der SAGA-Engine ist "I Have No Mouth, and I Must Scream" ablauffähig. Siehe Projekt-Homepage - <http://scummvm.sourceforge.net>.

<sup>8</sup>Mehr als zehn Spieletitel sind mit 90 prozentiger Unterstützung angegeben.

<sup>9</sup>Inklusive einer Reihe von Embedded-Architekturen, wie Palm OS, Symbian, GP2X, WindowsCE. Auf diese Weise lassen sich die Spiele auf tragbaren Geräten, wie Mobiltelefonen nutzen.

<sup>10</sup>Die Projektbeteiligung bestand zu einem wesentlichen Teil im Bereich "Preservation Action", siehe hierzu auch den Abschnitt 2.6.2.

- Querschnittsuntersuchungen der verschiedenen Virtualisierer und Emulatoren der X86-Architektur und praktische Vergleiche.<sup>11</sup> In diesem Zusammenhang wurden praktische Fragen erörtert, wie die Archivierung von Installationsdatenträgern und deren möglichst effiziente Handhabung für das Softwarearchiv. Dabei wird kurz auf die Problematik der Extraktion von Datenträgern und die Übertragung von Bitströmen von einem Datenträger auf einen anderen eingegangen.<sup>12</sup> Ebenfalls stand die Ermittlung des Treiberbedarfs in Abhängigkeit verschiedener X86-Emulatoren und -Virtualisierer im Fokus.
- das Kompilieren verschiedener Versionen eines Open-Source-Emulators aus seinem Quellcode und das Feststellen von Softwareabhängigkeiten.
- praktische Aspekte des Emulatorbetriebs: Migrationsexperimente des Mitführens verschiedener Gastsysteme durch fortgesetzte Versions-Updates am Beispiel der VMware Workstations von Version 3.2 bis 6.0 und zwischen verschiedenen Virtualisierern.
- Überlegungen zu möglichen Automatisierungen (VMware Konfigurationsdateien, QE-MU Monitor-Interface) und geeigneten Arten des Datenaustauschs.<sup>13</sup>

## A.3 Spezielle virtuelle Multi-Installationsdatenträger

Was für die Primärdaten gilt, dass sie unbedingt auch in ihrem Originalzustand archiviert sein sollten, sollte ebenso auf Sekundärobjekte angewendet werden. Dieses Vorgehen bewahrt weitgehend die Authentizität des ursprünglichen Datenträgers, erweist sich in der Anwendung aber als nicht sehr praktisch. Wer schon auf früheren Rechnergenerationen mit vielen Installationsdatenträgern hantiert hat, weiss Vereinfachungen zu schätzen. Vielfach ist es nicht möglich dabei auf älteren Systemen die neuen Techniken der Netzwerkinstallation von Software anzuwenden. Trotzdem lassen sich gewisse Vereinfachungen erreichen, die zudem einer Automatisierung bestimmter Prozesse Vorschub leisten können.

### A.3.1 Multi-Installations-ISO

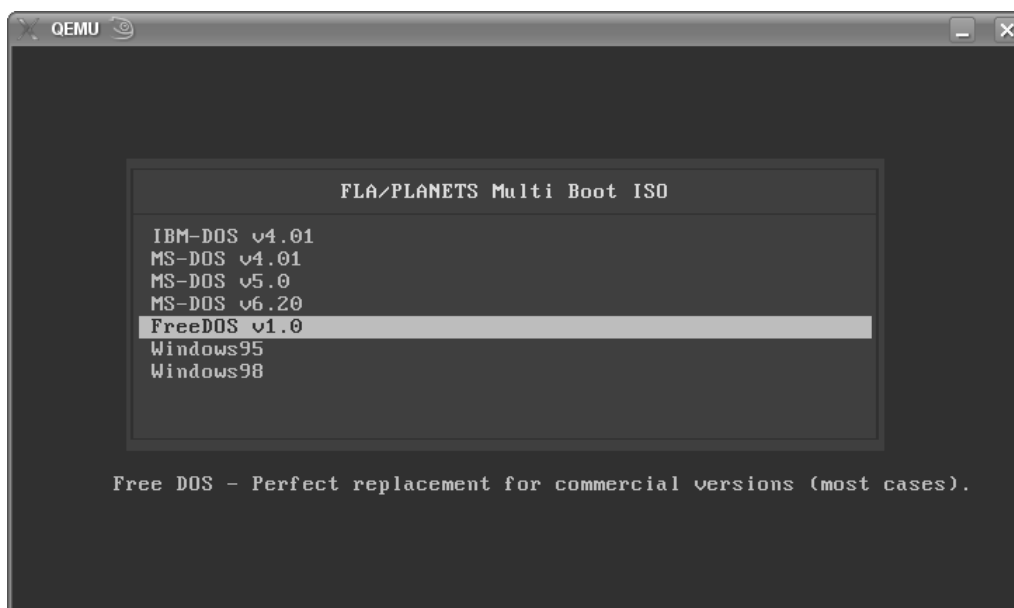
Mit der zunehmenden Größe von Betriebssysteminstallationen wurden Disketten unpraktisch. So passte das MS-DOS 5.0 noch auf vier 5,25" HD-Floppies, aber die Diskettenausgabe des Windows 95 erwies sich mit der größeren Zahl einzelner Datenträger als ziemlich unpraktisch. CD-ROMs als Installationsdatenträger übernahmen zunehmend die Aufgaben. Sehr schnell wurden Standards entworfen, um ebenfalls von CD-ROMs booten zu können. Auf dieser Grundlage bietet sich ein Zusammenfassen von Disketten auf optische Datenträger an. Die abstrakte Idee illustriert Abbildung 6.18, S. 178. Im Zuge der Experimente wurde ein sehr einfaches Generierungsskript erstellt, welches aus dem angelegten Beispielsoftwarearchiv einen bootfähigen Datenträger erzeugt. Voraussetzung hierzu besteht in der Verfügbarkeit der verwendeten Skriptsprache und dem Generierungsprogramm für das ISO-Image. Eine Unterstützung des ISO-Dateisystems auf dem erzeugenden System ist nicht erforderlich. Das gilt

---

<sup>11</sup>Vgl. hierzu auch Abschnitt 5.7.

<sup>12</sup>Vgl. dazu den Abschnitt 7.10.

<sup>13</sup>Vgl. hierzu die Abschnitte 5.6.1 und 6.8.



**Abbildung A.2:** Syslinux enthält mit "isolinux" ein ausführbares Programm zur Realisierung von Auswahlmenüs im BIOS-Mode. Das ebenfalls enthaltene "memdisk" erzeugt eine Ramdisk zur Floppy-Emulation für den Start von Boot-Disketten-Images verschiedener X86-Betriebssysteme.

ebenso für die Ausführbarkeit der eingebundenen Softwarekomponenten aus dem Syslinux-Paket.<sup>14</sup>

Um nicht für jedes installierbare X86-Betriebssystem einen eigenen Datenträger zu generieren, wird dem Nutzer nach dem Start ein Auswahlmenü präsentiert (Abb. A.2). Das hat zudem den Vorteil, dass gemeinsame Treiberpakete, die beispielsweise die Unterstützung verschiedener Windows Betriebssysteme für eine bestimmte Soundkarte enthalten, nicht mehrfach hinterlegt werden müssen. Für nicht-interaktive Installation kann der Default-Eintrag geändert oder die Zusammenstellung der Daten angepasst werden. Ebenso wäre es denkbar bei der Datenträgergenerierung die gewünschte Lokalisierung anzugeben, um automatisch die geeignete Tastaturbelegung auszuwählen.

Neben den Dateien für die verschiedenen Betriebssysteme enthält das ISO-Image alle notwendigen Treiber sowie eine Reihe oft benötigter Basiswerkzeuge, wie Dekompressoren. Grundsätzlich sollten auf der Installations-CD alle notwendigen Treiber und Pakete vorliegen, um in einem weiteren Schritt eine TCP/IP-Netzwerkverbindung einrichten zu können. Diese erlaubt eine deutlich höhere Flexibilität des Datentransports und ist nicht auf bestimmte Dateigrößen oder nur-lesbare Medien beschränkt.

### A.3.2 Installation aus einer Netzwerkquelle

Viele, auch sehr alte, Betriebssysteme stellen die notwendigen Werkzeuge und Treiber bereit oder sind entsprechend erweiterbar, um auf Computernetzwerke zugreifen zu können. Mit der Durchsetzung des TCP/IP-Standards als Technologie für die Verbindung von Rechnern untereinander und den Datenaustausch über vielfältige Protokolle kann eine einfache Schnittstelle

<sup>14</sup>Diese sollten ebenfalls im Softwarearchiv abgelegt werden. Homepage - <http://syslinux.zytor.com>.

für den Zugriff auf Installationsdateien geschaffen werden.

Hierzu sind beim Start der jeweiligen Nutzungsumgebung die Vorbereitung zur Einrichtung der virtuellen physikalischen und der darauf aufsetzenden TCP/IP-Schichten zu treffen. Für alle DOS und ältere Windows Betriebssysteme, wie 3.X oder 95 muss der Treiber für den TCP/IP-Stack separat installiert werden. Populär war für DOS das sogenannte NCSA-Telnet<sup>15</sup> oder Trumpet-WinSock<sup>16</sup> für Windows.

Darauf aufbauend lassen sich Protokolle für den Datentransfer, wie FTP, nutzen. Dieses ist sehr weit verbreitet und erlaubt auch zwischen sehr alten Nutzungs- und aktuellen Host-Umgebungen größere Datenmengen auszutauschen. Im Zuge des PLANETS Projects wurde ein Beispiel-FTP-Server aufgesetzt, um Softwarepakete bereitzustellen und Primärobjekte auszutauschen:

```
Connected to archive.ruf.uni-freiburg.de.
220 "Welcome to Long-Term Preservation FTP service (ALUF/PLANETS)."
```

Name (archive.ruf.uni-freiburg.de:user): planets

```
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||35899|)
150 Here comes the directory listing.
```

|            |   |     |     |                  |                |
|------------|---|-----|-----|------------------|----------------|
| drwxr-xr-x | 3 | ftp | ftp | 16 Nov 07 14:38  | homecomp-emu   |
| drwxr-xr-x | 3 | ftp | ftp | 49 Jan 17 12:47  | homecomp-games |
| drwxr-xr-x | 3 | ftp | ftp | 35 Jan 17 12:10  | homecomp-roms  |
| drwxr-xr-x | 7 | ftp | ftp | 114 Feb 01 14:50 | host-software  |
| drwxr-xr-x | 5 | ftp | ftp | 61 Jan 25 21:16  | incoming       |
| drwxr-xr-x | 2 | ftp | ftp | 22 Oct 30 14:33  | internal       |
| drwxr-xr-x | 2 | ftp | ftp | 89 Feb 01 14:53  | qemu-images    |
| drwxr-xr-x | 2 | ftp | ftp | 6 Jan 26 01:36   | vbox-images    |
| drwxr-xr-x | 7 | ftp | ftp | 86 Oct 30 15:05  | vmware-images  |
| drwxr-xr-x | 6 | ftp | ftp | 79 Jan 12 15:55  | x86-apps       |
| drwxr-xr-x | 5 | ftp | ftp | 109 Jan 12 15:14 | x86-games      |
| drwxr-xr-x | 6 | ftp | ftp | 68 Oct 26 05:53  | x86-os         |
| drwxr-xr-x | 9 | ftp | ftp | 88 Oct 08 15:36  | x86-os-driver  |
| drwxr-xr-x | 5 | ftp | ftp | 41 Dec 08 15:36  | x86-os-tools   |

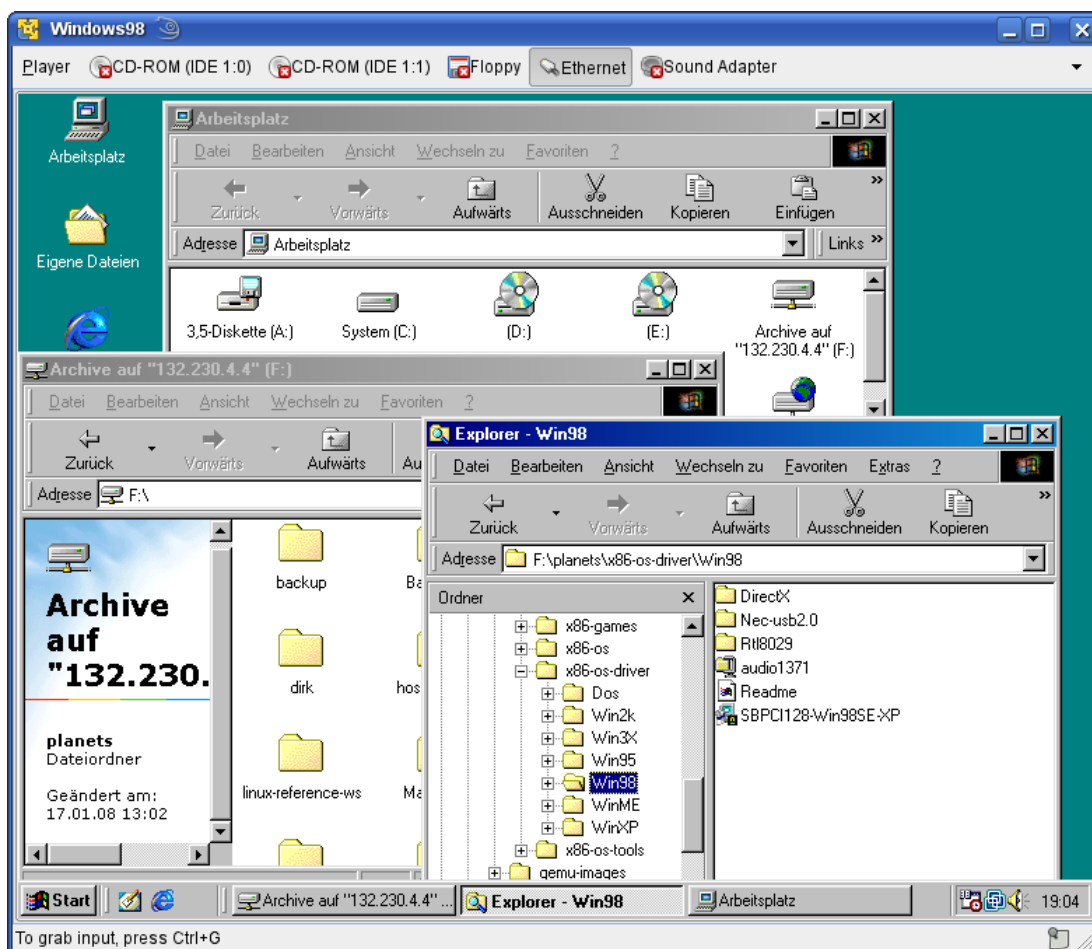
```
226 Directory send OK.
ftp>
```

Alternativ lassen sich Netzwerkdateisysteme, wie SMB/CIFS, für den Transfer von Objekten nutzen. Dieses Protokoll ist Bestandteil auch älterer Windows-Versionen und als Paket für Linux und Apple-Betriebssysteme erhältlich. Durch die ständige Weiterentwicklung

<sup>15</sup>Eine frühe Telnet- und FTP-Implementierung für PC und Mac, die von der University of Illinois am Urbana-Champaign National Center for Supercomputing Applications von 1986 bis 1995 entwickelt wurde.

<sup>16</sup>Eine Implementierung nach einer festgelegten Windows-API für TCP/IP-Anbindung.

der Spezifikation und die sukzessive Verbesserung von Sicherheitsfunktionen sinkt jedoch die Wahrscheinlichkeit der Kooperation sehr alter und neuer Implementierungen. In Experimenten



**Abbildung A.3:** Während die Einbindung von SMB-Shares unter Windows 98 und späteren Versionen recht problemlos gegen einen aktuellen Linux Samba-Server funktioniert, ist der Aufwand für ältere Microsoft Betriebssysteme höher.

gelang die Einbindung eines SMB-basierten Netzwerkdateisystems ab Windows 98 problemlos (Abb. A.3). Bei älteren Versionen traten etliche Schwierigkeiten auf, so dass hier der Zugriff per FTP eine bessere Wahl darstellt. Frühe Microsoft-Implementierungen des Protokolls für die Kommandozeile müssen zur Datenübertragung explizit auf den binären Transfermodus eingestellt werden. Andernfalls werden alle Nicht-ASCII-Objekte wegen der 7 bit-Übertragung beschädigt.

## A.4 Anwendung: Computerspiele-Museum

Computerspiele sind in den überwiegenden Fällen eine Herausforderung für die jeweilige Rechnerplattform, da sie an die Grenzen der Fähigkeiten der Hardware gehen. Das bedeutet im Umkehrschluss, dass ein im Emulator korrekt ablaufendes Spiel, ähnliche Erfolge für die anderen Applikationen der selben Nutzungsumgebung erwarten lässt.

Die durchgeführten Experimente sollten wegen des schieren Umfangs der bisher produzierten Spiele und des ähnlichen Verhaltens von Spieleklassen lediglich kurze Anhaltspunkte liefern, ob:

1. der Emulationsansatz generell und einzelne Emulatoren speziell dazu geeignet sind, einen langfristigen Zugriff auf Computerspiele sicherzustellen und
2. die Erfolgsbedingungen prüfen sowie mögliche Probleme und Herausforderungen identifizieren.

Das bedeutet, dass einige Computerspiele willkürlich für Tests ausgewählt wurden. Mit diesem Einsatzgebiet der Emulationsstrategie wird eine Nutzerperspektive eingeführt, die Erfahrungen für eine praktische Anwendung generiert.

#### A.4.1 Experimente und Setups

Ganz konkrete Fragestellungen ergaben sich nach der Vollständigkeit (im Sinne der Hardware-nachbildung) und Leistung einzelner Emulatoren sowie nach der Interaktion mit der jeweiligen Nutzungsumgebung. Zudem wurde untersucht, inwieweit das "Stacking" von Emulatoren sich darauf auswirken kann. Dieses spielt einerseits als Ansatz der Langzeiterhaltung, andererseits für die Paketierung und thematische Zusammenfassung.



**Abbildung A.4:** Anfang der 1990er Jahre wurden Spiele oft durch Abfragen geschützt, die anhand des mitgelieferten Handbuchs beantwortet werden sollten. Deshalb muss den Metadaten oder dem Objektpaket dieses beigelegt sein, um den Zugriff auf das eigentliche Abenteuer zu erlauben.

**Sammlung von C64 Spielen** Für die Klasse der Home-Computer kam eine Auswahl von Spielen für die wohl verbreiteste Maschine Commodore 64 zum Einsatz. Sie wurden im

Computerspiele-Museum<sup>17</sup> von den Originaldatenträgern ausgelesen. Um zu vermeiden, dass eine ganze Reihe von Primär- und Sekundärobjekten vom Archivnutzer behandelt werden muss, wird die Installation in eine virtuelle Maschine vorgenommen.

**Doom** war eines der ersten DOS-Spiele, welches in der flächigen Darstellung auf dem Bildschirm versuchte eine dreidimensionale Landschaft aus Spielerperspektive zu formen (Abb. 4.4, S. 95). Es hatte große Erfolge am Markt, erschien in mehrfach technisch verbesserten und erweiterten Versionen und schuf damit den Ausgangspunkt für die Entstehung eines ganzen Spielegenres. Die Grafikausgabe des Programms nutzt die halbe VGA-Auflösung. Es gibt Sounduntermalung und auch ziemlich neu – erste Netzwerkunterstützung für einen Mehrspielermodus.

**Monkey Island von Lucas Arts** zählt zu den charakteristischen Adventure Games, die zu Beginn der 1990er Jahre populär waren. Zu den hier vorgestellten Ansätzen (Abb. A.4) existieren weitere Alternativen, wie Abschnitt A.1.4 diskutiert (Abb. A.1, S. 229). Die Spiele wurden unter anderem für DOS erstellt und bieten Audiountermalung an.

**Myst von Broderbund** hatte für die damalige Zeit (1994) beeindruckend gerenderte Szenarien zu bieten, die jedoch noch unbewegt waren. Darüberhinaus waren eine Anzahl Multimedia-Clips enthalten, die nur durch eine Audioausgabe sinnvoll zu erfassen sind. Für die Darstellung war eine Festlegung der Farbtiefe auf 256 Farben zwingend erforderlich. Das Spiel erfordert zum Ablaufenlassen eine Windows-API.

**Command & Conquer 2** konnte im Einzelspielermodus oder über ein IPX/SPX später auch TCP/IP Netzwerk gegeneinander gespielt werden. Es wurde Mitte der 1990er Jahre aufgelegt und setzte eine Windows-API mit DirectX voraus.

#### A.4.2 Erkenntnisse

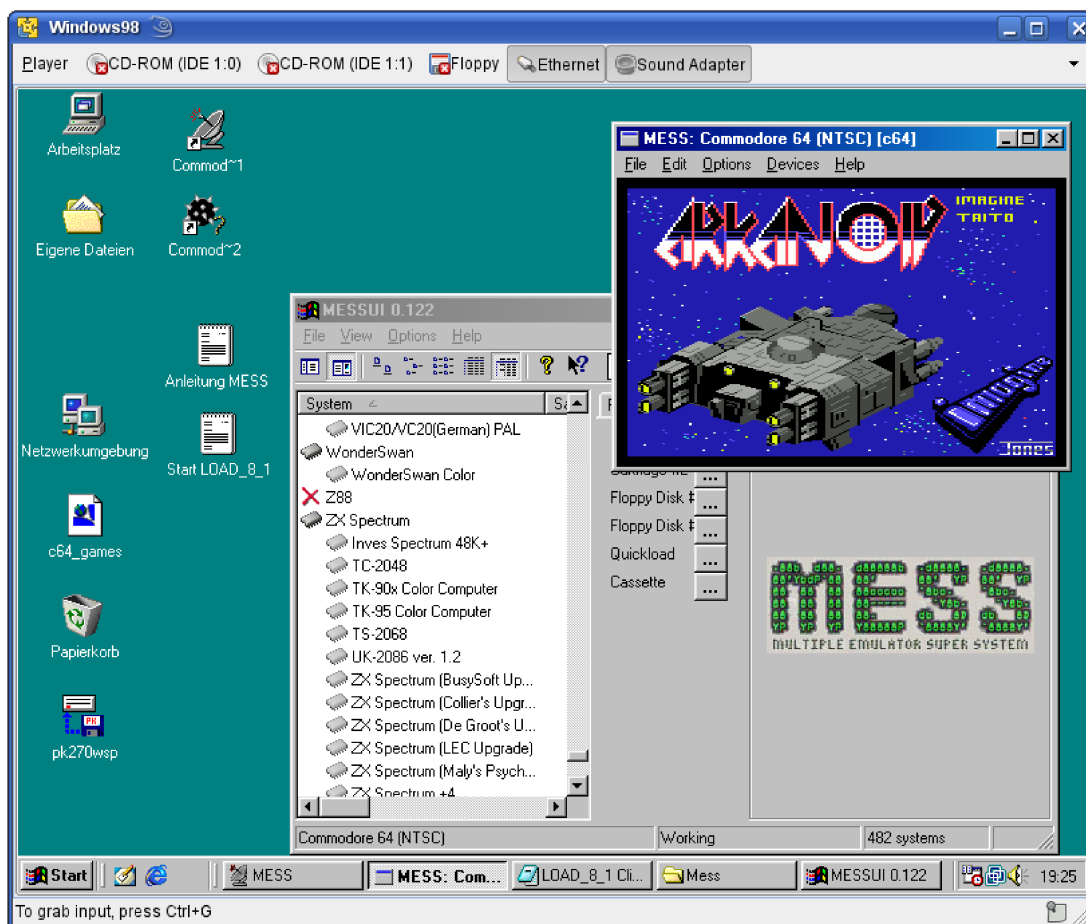
Für die Zugänglichmachung der C64-Spiele bietet sich der modulare Emulator MESS<sup>18</sup> an. Dieser wird von der Entwicklergemeinde für eine Reihe denkbarer Referenzumgebungen angeboten, ist jedoch für die Windows-API am weitesten vorangeschritten. MESS bietet ein intuitives grafisches Benutzer-Interface, an dem sich alle wesentlichen Einstellungen vornehmen lassen. Ebenso dient es dem Start, Stop der Emulation und der Bildgrößeneinstellung. Alternativ offeriert MESS die Steuerung via Kommandozeile, so dass sich bequem Shortcuts zu einzelnen Spielen auf dem Desktop der jeweiligen Nutzeroberfläche anlegen lassen.

Neben dem Spielepaket musste ein ROM-Image eines C64 für den Emulator erreichbar abgelegt werden. Spiele für den C64 können im Cartridge-, Disketten- oder Magnetbandformat abgelegt sein. Diese lassen sich beim Laden nicht gegeneinander austauschen und haben unterschiedliche Auswirkungen für den Start des jeweiligen Spiels. Während Cartridge-Images automatisch mit der C64-Emulation starten, müssen Programme vom Disketten-Image manuell aufgerufen werden. Wegen der größeren Anzahl der Primär- und Sekundärobjekte – beson-

<sup>17</sup>Vgl. [CSM 2008].

<sup>18</sup>Vgl. Abschnitt 5.1.1 und die Homepage [MESS 2008].





**Abbildung A.5:** Der modulare Emulator MESS bietet sich an in einer vorkonfigurierten Nutzungsumgebung wie einem Windows 98 im VMware Player abzulaufen. In einer solchen Umgebung lassen sich gut Metadaten wie Anleitungen paketieren und Shortcuts zum Start einzelner Spiele anlegen.

ders wenn man weitere Spiele auch anderer Home-Computer-Plattformen anbieten möchte – bietet sich eine Paketierung bestehend aus einem Windows in einer Virtualisierungsumgebung oder einem Emulator wie QEMU an (Abb. A.5). Im Test gab es keinerlei Probleme mit Benutzerinteraktion und der Darstellung der Primärobjekte. Sowohl der äußere als auch der innere Emulator beherrschen einen Vollbildmodus, was zusammengekommen einen guten Gesamteindruck auf einer Referenz-Workstation erlaubt. Ebenso wurde der begleitende Sound des Spiels in ausreichender Qualität durchgereicht.

Obwohl Spiele wie Command & Conquer 2 durchaus unter Windows XP noch installiert werden können und ablauffähig sind, kann es trotzdem sinnvoll sein, sie in eine Emulations- oder Virtualisierungsumgebung zu paketieren. Auf diese Weise kann eine Verteilung an Wissenschaftler oder Ausstellungen erfolgen, ohne dass die Empfänger über ein umfassendes Expertenwissen verfügen müssen. So lässt sich sicherstellen, dass alle notwendigen Softwarekomponenten bereitstehen, falls vorhanden der Lizenzcode eingetragen wurde und das Handbuch in elektronisch lesbarer Form bereitliegt. Zudem können weitgehende Tests im Vorhinein erfolgen. Alle Experimente erfolgten auf verschiedenen Windows XP/Vista und Linux-Referenz-Workstations.

Notwendige Expertise und Wissen können so an wenigen Stellen gebündelt sein, wenn man sich auf die Austauschenebene der Virtualisierung begibt. Zudem bieten sich Überlegungen zu Automatisierung und Skripting von Startvorgängen an. Hier bietet QEMU mit dem Monitor-Interface gute Ansatzpunkte.

Für eine Reihe älterer DOS-Spiele, wie hier im Beispiel Atlantis<sup>19</sup> und Doom (Abb.A.4 und 4.4, S. 95), bieten sich für eine Ausführung in der DOSBOX an. Es wird keine gesonderte DOS-Installation und damit keine Lizenz benötigt, vorteilhaft ist zudem die fehlende Startverzögerung durch die sonst übliche Initialisierung des Betriebssystems. Die notwendigen Schnittstellen für die erwartete Hardware zur Audioausgabe standen zur Verfügung. Den Vorteilen des schnellen Starts und geringen Overheads der Emulationsumgebung stehen die Beschränkungen auf wenige Referenzumgebungen und eine begrenzte Menge an Software speziell Spielen gegenüber.

Der Einsatz eines Emulators oder Virtualisierers der X86-Plattform bietet den generelleren Ansatz, da fast alle Betriebssysteme dieser Architektur ablauffähig sind. Der Overhead und die Startverzögerung für den Zugriff auf die Nutzungsumgebungen fallen jedoch höher aus. Dieses lässt sich jedoch durch die Möglichkeit des Einfrierens bestimmter Zustände umgehen. Nutzungsumgebungen können an einem bestimmten Punkt in der Ausführung angehalten werden und zu einem beliebigen späteren Zeitpunkt wieder von diesem Punkt an weiterlaufen. Zusätzlich können innerhalb neuerer Nutzungsumgebungen Metadaten wie Anleitungen, Zusatzinformationen, Bilder der Packungen der installierten Software, für den Archivnutzer bequem erreichbar hinterlegt werden. Für größere Sammlungen sind verschiedene Arten des Paketierens vorstellbar. ISO-Images neuerer Spiele können extern als optisches Laufwerk mittels Emulator eingebunden werden oder innerhalb des Dateisystems der Nutzungsumgebung abgelegt sein. Der zweite Fall reduziert die Menge der zu behandelnden Dateien, erfordert jedoch Spezialwerkzeuge, wie die Daemon-Tools (Abschnitt 7.10.5).

In den Tests zeigten sich zudem die Unterschiede der Hardwareemulation. So verlangte das Spiel Command & Conquer 2 eine Funktion der Grafikkhardware, die QEMU bieten konnte, VMware Workstation oder Player hingegen nicht.

## A.5 Datenträger

Elektronische Daten müssen über das Ausschalten eines Computers weg dauerhaft gespeichert werden können. Hier meint jedoch die Dauerhaftigkeit erst einmal nicht besonders lange Zeiträume, sondern den meist recht kurzen Zeitabschnitt bis zum Wiedereinschalten der Maschine. Für diesen Zweck wurden im Laufe der Zeit etliche Arten von Datenträgern entwickelt. Datenträger sind im Zuge dieser Betrachtungen insofern von Interesse, als dass sie bestimmte Sekundärobjekte enthalten können. Hierzu zählen einerseits die Speichermedien für die Softwareinstallation. Andererseits können es komplette Festplatten sein, die von ihrer physikalischen Repräsentation in ihre virtuelle Entsprechung umgewandelt<sup>20</sup> werden.

Digitale Informationen können auf jedem beliebigen Medium gespeichert werden, welches

<sup>19</sup>Eine zusätzliche Alternative besteht in der Verwendung von ScummVM, vgl. auch Abb. A.1, S. 229. Interessanterweise entfällt die Abfrage, wie sie Abb. A.4 zeigt.

<sup>20</sup>Vgl. Überlegungen in Abschnitt 6.6.1.

in der Lage ist Bits aufzuzeichnen.<sup>21</sup> Verschiedene Medientypen speichern die Äquivalente der "0" und "1" in unterschiedlicher Art und Weise abhängig von den Eigenschaften des Mediums und verschiedener Festlegungen. Das Aufzeichnen und spätere Wiederauslesen oder Ändern dieser Bits erfordert spezielle Hardware, die ihrerseits wiederum gesteuert werden muss.

Ein wichtiges Klassifikationskriterium von Speichermedien ist die Art, wie Informationen auf ihnen repräsentiert werden. Eine wichtige Medienklasse bilden magnetische Speichersubstanzen, die bei Festplatten, Disketten, Magnetbändern, und Magnetkarten eingesetzt werden. Weiterhin spielen optische Speichermedien, die mittels Laser die Daten in spezielle Substanzschichten einbrennen, derzeit von Bedeutung. Hierzu zählen CD und DVD mit Varianten, sowie als neuere Entwicklung die Blu-ray Disk. Holografische Speicher könnten die optischen Speichermedien der Zukunft sein. Während alle bis dahin genannten Datenträger mit mechanischen Komponenten arbeiteten, werden diese bei halbleiterbasierten Speichermedien, wie Flash Disks vermieden.

Zudem lassen sich Speichermedien nach der Art ihrer Beschreibbarkeit klassifizieren. Man unterscheidet hierbei zwischen nur lesbaren, einmalig beschreibbaren und wiederbeschreibbaren Datenträgern.

Datenträger können zudem nach der Art ihrer Nutzung unterschieden werden. So liegt beispielsweise bei Festplatten eine starre mechanische Kopplung zwischen dem Datenträger und Lese-Schreib-System vor. Der Datenträger alleine kann nicht vom restlichen System getrennt werden. Die Kapselung hat den Vorteil, dass sie durch Reinheit des Funktionsraumes und die mechanische Verlässlichkeit sehr hohe Packungsdichten erlaubt. Zudem ist der Datenträger gut vor Umwelteinflüssen geschützt. Damit entscheidet neben der funktionellen Integrität die Verfügbarkeit der Schnittstelle der Festplatte über den Zugriff auf die gespeicherten Daten.

Wechseldatenträger haben als hervorstechendste Eigenschaft die Möglichkeit den Datenträger vom Laufwerk zu trennen. Das stellt zum einen an den Datenträger höhere Stabilitätsansprüche gegenüber Umwelteinflüssen und zum anderen Anforderungen an eine gewisse mechanische Robustheit. Die Lesbarkeit der gespeicherten Daten ist mit der physikalischen Haltbarkeit der Aufzeichnung und der Verfügbarkeit passender Laufwerke verknüpft. So können sich beispielsweise Schnittstellen in der Zwischenzeit weiterentwickeln, ohne dass die Medien geändert werden müssen. Einen Überblick für Nichttechniker findet sich in [Rohde-Enslin 2004].

### A.5.1 Computerdisketten

Disketten gehören zu den ältesten magnetischen Datenträgern mit blockorientiertem wahlfreiem Zugriff. Die erste Variante<sup>22</sup> erschien 1971. Fünf Jahre später wurde das sich breit durchsetzende 5,25" große Diskettenlaufwerk vorgestellt, 1981 folgte Sony mit dem 3,5" Format mit einem im Gegensatz zu den Vorgängern starren Gehäuse.

---

<sup>21</sup>Siehe auch [Rothenberg 1999], S. 2. Zudem gibt die Tabelle S. 3 einen Überblick zur erwarteten Lebensdauer verschiedener Medien.

<sup>22</sup>IBM brachte die Alan Shugart zugeschriebene Erfindung in einer 200 mm Durchmesser, 8" 80 KByte fassenden Variante auf den Markt.

**Physische Eigenschaften** Disketten unterscheiden sich nicht nur in der äußeren Form, sondern lassen sich in ein- und zweiseitig beschriebene Disketten<sup>23</sup> unterteilen. Weiterhin kamen verschiedene Aufzeichnungsdichten vor, die sich aus der eingesetzten Magnetschicht ergaben.<sup>24</sup> Frühe Disketten waren hard- spätere nur noch soft-sektoriert.

Hard-sektorierte Datenträger verfügen auf der Kunststoffscheibe über ein Indexloch pro Sektor, soft-sektorierte nur eines insgesamt. Die Indexlöcher definieren den Startpunkt einer Spur, sofern es vom Laufwerk nicht generell ignoriert wird.

Neben diesen Information findet man noch weitere Angaben, wie die Anzahl der Magnet Spuren,<sup>25</sup> die Zahl der Sektoren.<sup>26</sup> Vielfach wurde noch die Blockgröße angegeben, die früher 128, 256 oder 512 Byte betragen konnte, heute aber meistens mit 512 Byte festgelegt ist. Daneben findet man Angaben zur Spurdichte<sup>27</sup> Diskettenlaufwerke verwendeten unterschiedliche Aufzeichnungsverfahren, ursprünglich anfänglich Frequency Modulation (FM), später meistens MFM. Systeme wie Commodore oder Apple setzten zusätzlich auf das GCR-Verfahren. Für weitergehende Informationen kann man [Messmer 1997] S. 765 ff. heranziehen.

**Neuere Diskettenstandards** Im Laufe der Zeit entwickelten sich die Speicherkapazitäten von Disketten und Festplatten immer weiter auseinander. Dem versuchte man mit sogenannten ZIP-Laufwerken mit anfänglich 100 MByte, später 250 MByte und am Ende 750 MByte Kapazität zu begegnen. Zur Erhöhung der Datentransferraten konnten ZIP-Laufwerke am IDE- oder SCSI-Bus betrieben werden. Damit waren sowohl die Schnittstellen als auch die Medientypen zu den bisherigen Diskettenstandards inkompatibel.

Ein Kompromiss zwischen klassischer Diskette und ZIP stellt der LS120 Standard dar. Diese Laufwerke hielten das Format der 3,5" Diskette ein. Sie waren so in der Lage spezielle hochkapazitive und traditionelle Disketten zu lesen und zu schreiben. Durch Verwendung der IDE-Schnittstelle stieg die Datentransfargeschwindigkeit deutlich an. Trotz ihrer besseren Kompatibilität konnte sich dieses Diskettenformat nicht durchsetzen. Für höhere Kapazitäten sorgten zum gleichen Zeitraum immer preiswerter verfügbare schreibbare und wiederbeschreibbare CD-ROMs.

## A.5.2 Magnetfestplatten

Festplatten sind nichtflüchtige Massenspeichermedien mit wahlfreiem, blockorientiertem Zugriff. Dabei ist ein theoretisch beliebig häufiges Schreiben und Lesen der Daten möglich. Das Medium besteht aus einem starren Material mit einer magnetisierbaren Oberfläche. Im Gegensatz zur Diskette realisiert die Festplatte eine höhere Datendichte auf dem Medium.

<sup>23</sup>"Single Sided", "Double Sided" Eigenschaften sind oft durch die Abkürzungen "SS" und "DS" angegeben. Dieses spielte bei älteren Rechnerplattformen eine Rolle.

<sup>24</sup>Bezugspunkt ist die "Single Density" (SD) der ersten Disketten, Double Density (DD), High Density (HD) und später Extended Density (ED), welches sich jedoch aufgrund des komplexeren Aufzeichnungsverfahrens und höherer Datenträgerpreise nicht durchgesetzt hat.

<sup>25</sup>Tracks – auf frühen Disketten meist 35, später 40 oder 80.

<sup>26</sup>Blöcke pro Spur, klassischerweise 18.

<sup>27</sup>Diese beträgt üblicherweise 96 tpi (tracks per inch), Hersteller wie Commodore verwendeten bei größeren Laufwerken abweichende Werte wie 100 tpi, um mehr Daten auf einer Scheibe unterbringen zu können. Aus diesen Abweichungen resultierten Probleme, wie die Unlesbarkeit solcher Datenträger in Fremdlaufwerken.

Einige Festplatten sind als wechselbare Datenträger ausgelegt, wie beispielsweise die Implementationen von Syquest oder Iomega.<sup>28</sup> Der wesentliche Unterschied der Festplatte zur CD-ROM oder DVD-RW besteht in der Wahlfreiheit des Datenzugriffes, der Positionierungsgeschwindigkeit und der Organisation der Datenblöcke.

Die Speicherung der Daten erfolgt mittels magnetischen Verfahrens auf einem oder mehreren rotierenden Datenträgern. Hierfür schweben die Leseköpfe über dem Medium und registrieren unterschiedlich gepolte Magnetfelder. Das Schreiben der Daten erfolgt durch Veränderung der Magnetisierung. Die Speicherung ist ein analoges Verfahren. Man benötigt einen Analog-Digital-Wandler zur Extraktion der digitalen Information.

Die Adressierung des Datenträgers erfolgt über die Sektoren. Zylinder-, Kopf und Sektornummer bestimmen die Position eines bestimmten Datums. Die übliche Sektorlänge beträgt 512, 1024, 4096 Byte. Die maximale Zahl aus Sektoren, Zylindern und Köpfen ergibt die Gesamtkapazität des Datenträgers. Jedoch gibt es Unterschiede zwischen der Bruttokapazität (unformatierter Datenträger) und dem Nettospeichervermögen (Low-Level formatierte Kapazität), da die Formatierung erst die Synchronisations-, CRC-Prüf- und Korrekturfelder einfügt.

Anders als Wechsellaufwerke sind Festplatten gekapselt, weshalb der Datenträger nicht vom Laufwerk getrennt werden kann. Bei einem Austausch wird das komplette Gerät samt Datenträger obsolet.

Nach einer anfänglichen Vielfalt von Schnittstellen sowohl für Wechsellaufwerke als auch Festplatten haben sich zwei Standards durchgesetzt. Die ATA- bzw. SCSI-Schnittstellen werden seit mehr als zehn Jahren verwendet. Jedoch gab es immer wieder Upgrades der Spezifikation, um mit den technologischen Anforderungen Schritt zu halten. Deshalb sind die Schnittstellen nur in einem gewissen Grade abwärtskompatibel. So gibt es allein für die SCSI-Schnittstelle eine ganze Reihe verschiedener mechanischer und elektrischer Beschreibungen. Bei der ATA-Schnittstelle erfolgt derzeit eine Umstellung des 40-poligen parallelen Interfaces auf ein 4-poliges serielles Interface. Jedoch werden für eine ganze Weile beide Schnittstellentypen nebeneinander existieren.

Für die Erstellung von Festplatten-Images – aus Sicht der Betriebssysteme ist nur das Konzept "Blockdevice" von Interesse – spielen die diversen technischen Daten nur eine untergeordnete Rolle. Während die meisten Virtualisierer und Emulatoren von IDE-Geräten ausgehen, kennt VMware sowohl IDE als auch SCSI-Container, die sich im inneren Aufbau unterscheiden. Besonders zu Berücksichtigen sind jedoch Partition Table und Master Boot Record, da diese für das initiale Laden des jeweiligen Betriebssystem-Kernels zu kennen sind.

Weitere Angaben zu Festplatten, der Entwicklung ihrer physischen und elektronischen Schnittstellen finden sich in [Messmer 1997] S. 847 ff.

### A.5.3 Optische Datenträger

Optische Speichermedien legen ihre Daten im Gegensatz zu magnetischen Speichersystemen, wie Festplatte oder Diskette, in digitaler Form ab. Sie besitzen vergleichbar mit der

<sup>28</sup>Syquest-Wechselmedien sind inzwischen vom Markt verschwunden, stellten aber lange Zeit den Quasi-standard zum Transport größerer Datenmengen. Diese Aufgabe übernahmen später die vorgenannten ZIP-"Disketten", welche jedoch mit Kapazitäten von 100 - 250 MByte auch nicht mehr dem Stand der Technik entsprechen und von CD-R oder CD-RW als preiswertere Medien verstärkt verdrängt wurden.

inzwischen fast historischen Vinyl-Schallplatte nur eine einzige durchgehende Spur die spiralförmig von innen nach außen verläuft. Deshalb fallen die mittleren Zugriffszeiten, trotz hoher Drehgeschwindigkeiten und hohen linearen Transferraten, deutlich ungünstiger als bei einer Festplatte aus. Eine Ausnahme bilden magneto-optische Medien und die DVD-RAM.<sup>29</sup>

Entlang der Spur tastet ein Laserstrahl das optische Medium ab, um Daten zu lesen. Die Datenträger bestehen im wesentlichen aus einem Trägermaterial (üblicherweise Kunststoff) und einer darauf aufgetragenen Reflexionsschicht. Darüber liegt eine Schutzschicht, um Beschädigungen auf der Reflexionsschicht zu vermeiden. Die Reflexionsschicht verfügt über zwei Zustände, die den Laserstrahl je nach Zustand unterschiedlich zurückwirft. Ein Sensor, wie eine Fotodiode, registriert den reflektierten Strahl und wandelt ihn in einen digitalen Datenstrom um. Ein Überblick zu CD-ROM und WORM liefert beispielsweise [Messmer 1997] S. 936 ff.

**Compact Disks** Sie kann bis 700 MByte Daten aufnehmen oder bis zu 80 Minuten Musik im Audio-Standard<sup>30</sup> speichern. Kombinationen von Audio und Daten auf einem Datenträger sind möglich. Als Massenware liegen die Preise für CD's so niedrig, dass fast jede Software auf einer Installations-CD geliefert wird. Mit der Verfügbarkeit preiswerter CD-Schreiber (bzw. -Brenner) spielte das Medium auch im Bereich des Backups eine zunehmende Rolle.

**Die Digital Versatile Disc (DVD)** Die DVD hat die Nachfolge der CD als optischer Datenträger im Endanwenderbereich angetreten. Wie bei vielen anderen Entwicklungen auch, gelang es den Herstellern zu Beginn nicht, sich auf einen einheitlichen Standard für dieses Medium zu einigen. Die Unsicherheit für Produzenten, Lizenznehmer und Konsumenten hat sicherlich zur Verzögerung ihrer Einführung beigetragen.

DVDs und CDs sehen sich äußerlich zum Verwechseln ähnlich, da sie identische Dimensionen besitzen. Ein wesentlicher Unterschied liegt neben der höheren Spurdichte bei der DVD in der Mediendicke. DVD bringt auf den 1,2 mm, die eine CD an Höhe hat, zwei Scheiben der halben Dicke unter. Diese werden aufeinander geklebt. Meistens ist nur eine Scheibe als Datenträger benutzt und die zweite ein einfacher Plastikdummy, jedoch gibt es auch doppelseitig beschriebene DVDs, die bei Bedarf zum Auslesen der anderen Seite im Laufwerk gewendet werden müssen.

Die DVD lässt sich als Ton-, Bild- und Datenträger einsetzen. Sie hat die gleichen Abmessungen wie eine CD, kann jedoch ein vielfaches an Daten speichern. Dazu wurden die Pits auf eine Größe von 0,4 Mikrometer reduziert und der Abstand der Spur auf 0,74 Mikrometer verringert. Um diese kleinen Pits genau abtasten zu können, wurde die Wellenlänge des Abtastlasers verringert. Man erreicht dadurch eine bis zum Faktor 25 gesteigerte Kapazität gegenüber einer CD. Auf der DVD kann auf zwei Ebenen übereinander und auf jeder Seite gespeichert werden. Auf einer Seite mit einfacher Speicherschicht erzielt man so ein Volumen von 4,7 GByte.

<sup>29</sup>Sie ist zwar im Zuge des DVD-Standards mitdefiniert worden, wird aber eher selten eingesetzt. Nicht alle Laufwerke können DVD-RAM lesen, wenige der üblichen Brenner diese schreiben.

<sup>30</sup>Die ersten Daten-CDs besaßen eine Kapazität von 650 MByte; inzwischen gibts Datenträger mit bis zu 800 MByte Kapazität. Das entspricht einer Audiospielzeit zwischen 74 und 80 Minuten. Sony hat versucht eine High-Density-CD mit doppelter Kapazität zu etablieren. Diese hat sich jedoch wegen fehlender Laufwerksunterstützung anderer Hersteller nicht durchgesetzt.

Zweiseitig und doppelschichtig können insgesamt 17 GByte untergebracht werden. Die zweischichtige Speicherung gelingt durch die Verwendung von halbdurchlässigem Material für die erste Schicht. Die zweite ist undurchlässig. Der Laser wird zum Lesen auf eine der beiden Schichten fokussiert. DVD-Laufwerke sind typischerweise abwärtskompatibel, das heißt, dass sie außer DVD's auch alle Arten von CD's lesen. Da im Laufe der Zeit die erreichten Kapazitäten nicht mehr genügten, wurde die High Density (HD) DVD definiert.

Mit etwas Verzögerung haben sich (wieder)beschreibbare Medien am Markt etabliert. Die Formate unterscheiden sich nicht sehr stark, da die Technik optisch Daten zu speichern und zu lesen recht einheitlich umgesetzt wird. Erhältlich sind derzeit DVD-R, -RW und DVD+R, +RW sowie DVD-RAM.

Alle einmal beschreibbaren DVD-Medien also DVD-R und DVD+R stammen meistens aus denselben Pressmaschinen, arbeiten mit denselben Farbstoffen und demselben Phase-Change-Verfahren. Dieses gilt auch für fast alle RW-Medien. Die Unterschiede ergeben sich aus der unterschiedlichen Anlage der Spiralen, denen der Laser beim Schreiben und Lesen folgt. DVD-RAM-Medien fallen hier heraus, da die Datenblöcke in konzentrischen Kreisen, wie auf einer Festplatte angeordnet sind.

DVD-RAM-Medien wurden von vornherein als Nur-Datenspeicher konzipiert. Die Rohlinge wurden ursprünglich zum besseren Schutz im Caddy geliefert. Inzwischen wird diese Maßnahme zur Verlängerung der Lebensdauer wegen verbesserter Materialien nicht mehr für nötig erachtet. Theoretisch sollen die Medien bis zu 100.000 mal wiederbeschreibbar sein. DVD-RAM unterscheiden sich durch vorgestanzte Rillen und sind nicht in jedem DVD-Laufwerk lesbar.

Wegen des Einsatzes als Speicher von Videomaterial wurden DRM-Techniken, wie das Content Scrambling System als auch der Regionalcode eingeführt. Sie wurden relativ schnell gebrochen, so dass die Schutzwirkung fragwürdig ist. Deshalb gibt es jedoch Probleme solcherart geschützte Medien innerhalb von Virtualisierungsumgebungen anzusehen. Als Kopierschutz der HD DVD wurde das Advanced Access Content System (AACS) vorgesehen, welches auch bei der Blu-ray Disk zum Einsatz kommt.

**Blu-ray Disks** Sie zählt zu den designierten Nachfolgern der DVD. Dabei bezieht sich das Kunstwort "blu-ray" auf einen blau-violetten Laser, eine höhere Packungsdichte, einen höheren Anspruch an Mechanik und Laser. Deshalb erfolgte ein Wechsel vom langwelligen roten Laser auf kürzerwelligen blauen Laser der wegen der feineren Oberflächenstrukturen im Gegensatz zu den für CD und DVD eingesetzten roten Laser verwendet wird.

Das physische Format einer Blu-ray Disc ist mit einem Durchmesser von 120 mm identisch zu CD und DVD. Eine einlagige Scheibe nimmt bis zu 27 GByte auf, die sich mit zwei Lagen auf bis zu 54 GByte Daten verdoppeln lassen. Ein wichtiger Bestandteil der Blu-ray-Spezifikation besteht im Kopier- und Abspielschutz AACS aus dem Bereich des Digital Rights Management. Das Auslesen geschützter Daten kann deshalb ein Problem werden und damit die enthaltenen digitalen Objekte gefährden.

Während dieser Datenträgertyp ein gewisses Kapazitätsverhältnis zwischen aktuellen Festplatten und Wechselmedien wiederherstellt, kann man davon auszugehen, dass auch diese Datenträger kapazitätsmäßig überholt werden. Deutlich problematischer für eine längerfristige Archivierung und Nutzung sind die eingebauten DRM Komponenten – sie können Inhalte

stärker als Alterung des Mediums gefährden.<sup>31</sup>

**Magneto-Optische Datenträger** Obwohl verwandt zu CD-ROM und DVD haben sich WORM's (Write Once Read Many) nicht durchsetzen können. Sie arbeiten nach einem ähnliches Verfahren wie CDs und verwenden zwei verschiedene Laserstrahlen zum Schreiben und Lesen. Wegen ihrer höheren garantierten Datensicherheit werden sie in bestimmten Bereichen, wie beispielsweise Versicherungen noch eingesetzt.

Es gibt Mischformen optischer und magnetischer Speicherung: sogenannte magneto-optische Laufwerke. Diese erreichen aufgrund optischer Spurführung und Steuerung eine höhere Datendichte als rein magnetische Wechsellaufwerke. Jedoch hat auch dieses Verfahren sich aufgrund des relativ hohen Preises und deshalb eher seltenen Einsatzes nicht wirklich durchsetzen können.

#### A.5.4 Weitere Datenträger

Magnetbänder zählen zu den ältesten Speichermedien. Sie bestehen seit längerer Zeit aus einer langen, schmalen Kunststoffolie, welche mit einem magnetisierbaren Material beschichtet wurde. Es existieren eine Vielzahl verschiedener Formate und dementsprechend viele Schreib- und Lesegeräte. Bis Anfang der 1980er Jahre zählten Magnetbänder zu den Standardinstallations- und Speichermedien. Für Home-Computer kam als preiswerte Lösung die sogenannte Datasette zum Einsatz, Laufwerke, die Daten auf handelsüblichen Audiokassetten speichern konnten. Wegen der Aufbewahrung in engen Wicklungen, ließen sich ältere Medien oft nicht mehr sauber trennen. Zudem kam mit der immer höheren Packungsdichte das Problem der schleichenden Entmagnetisierung hinzu, da immer weniger Partikel zur Speicherung eines Bits zur Verfügung standen.

In jüngerer Zeit kam mit den Flash-Speichern<sup>32</sup> ein neuartiger Datenträgertyp hinzu, der anders als die bisher behandelten Datenträger die Information nicht als Zeichen auf einem Medium (Vgl. [Thibodeau 2002]), sondern in elektronischen Schaltungen nichtflüchtig festhält. Diesen Speichertyp, von dem Einheiten bis zu Kapazitäten von mehreren Gigabyte angeboten werden, findet sich in den verschiedensten elektronischen Geräten, wie Kameras, Mobiltelefonen, MP3-Playern.

---

<sup>31</sup>Vgl. hierzu Abschnitt 7.11.

<sup>32</sup>Die technisch genaue Bezeichnung lautet Flash-EEPROM.



## B Glossar

Die hier aufgelisteten Begriffe sind in dieser Arbeit von Bedeutung. Findet sich im Hauptteil eine ausführliche Darstellung, so erfolgt hier nur eine kurze Zusammenfassung mit dem Verweis auf den entsprechenden Abschnitt. Die Begriffe wurden im Haupttext beim ersten Auftreten durch Kursivsetzung gekennzeichnet.

**Ablaufumgebung** siehe *Nutzungsumgebung*.

**AIP** bezeichnet das *Archival Information Package* als eine Zusammenfassung des in das Langzeitarchiv aufzunehmenden Primärobjekts mit seinen Metadaten. In den Informationspaketen werden inhaltliche Informationen und Preservation Description Information (PDI) zusammengefasst. Erstere, die Content Information, stammt aus den übernommenen und eventuell aufbereiteten Ursprungs- und Metadaten. Zweitere erklärt welche Techniken und Verfahren auf die Inhaltsinformation angewandt wurden. Hierzu zählen Maßnahmen der Sicherung, eindeutigen Identifikation und Einordnung in ihren jeweiligen Kontext. Zudem sollten Hinweise für den zukünftigen Zugriff angebracht werden.

**Applikation** Dieses Wort wird üblicherweise für Computerprogramme, Anwendungsprogramme, auch *Software* genannt, gebraucht. Sie besteht aus ausführbarem Code, der für eine bestimmte Rechnerplattform aus dem Quellcode übersetzt (compiliert) wurde. Deshalb ist eine Applikation zur Ausführung an eine bestimmte Prozessorarchitektur gebunden.

**Archiv** meint traditionell eine Institution oder Organisationseinheit, in der Dokumente in weitestem Sinne des Wortes, erfasst, erschlossen, erhalten, ausgewertet und zugänglich gemacht werden. Diese Archivalien werden nach bestimmten, je nach Organisation festgelegten, Kriterien eingestellt. In der heutigen Zeit wird der Begriff um die elektronische Archivierung erweitert. Die Begriffe Aufbewahrung von Information und Archivierung gehen dabei nahtlos ineinander über. Traditionelle Archive werden in Abschnitt 2.1 ausführlicher erläutert. Für moderne Archive sei speziell auf das *Digitale Langzeitarchiv* verwiesen. In dieser Arbeit wird der Begriff im Sinne der postmodernen Kulturwissenschaft gebraucht und als Oberbegriff für Informations- und Wissensspeicher wie Museen und Bibliotheken eingesetzt. Vielfach verwendet man für diese Zusammenfassung auch den Begriff *Gedächtnisorganisation*. Eine kurze Überblicksdarstellung liefert [Coy 2006] mit dem Abschnitt "Die Digitale Bibliothek" oder umfassender [Endres und Fellner 2000].

**Archivalien** sind die Objekte von Archiven. In *digitalen Langzeitarchiven* handelt es sich dabei um Dateien bestimmter Dateiformate, die auf geeigneten Datenträgern gespeichert sind.

**Betrachter** oder Dateibetrachter meint ein Programm (Applikation), welches dazu dient digitale Objekte zugänglich zu machen. Üblicherweise haben Dateibetrachter gegenüber den erstellenden Applikationen, den Editoren, die Einschränkung, dass sie Dateien nicht verändern (können). Das spielt für die Langzeitarchivierung nur eine untergeordnete Rolle: So wäre beispielsweise ein altes AmiPro ein Programm zur Textverarbeitung. Im Sinne des Langzeitzugriffs auf Daten spielt jedoch nur die Fähigkeit des Programms, den Inhalt von Textdateien eines bestimmten Formates anzuzeigen, eine Rolle.

Das Dateibetrachten kann je nach Datenformat und Programm durchaus sehr verschiedenes bedeuten: Ein Programm zum Ansehen von PDF-Dokumenten besitzt eine grafische Schnittstelle, in der seitenweise der Inhalt bestehend aus Text oder Bild angezeigt wird. Dabei stellt das Programm je nach Umsetzung Hilfsmittel zum Suchen im Dokument, Hyperlinks, Vergrößerung und Verkleinerung der Ansicht, Blättern oder Ausdrucken zur Verfügung. Die grafische Schnittstelle ihrerseits greift zum Anzeigen auf Bibliotheken des grafischen Benutzer-Interfaces und des *Betriebssystems* zurück. Dieses gilt ebenfalls für die Wiedergabe auf Papier (Ausdruck).

**Betriebssystem** bezeichnet Software, welche die Betriebsmittel eines Computers wie Speicher, Ein- und Ausgabegeräte verwaltet und die Ausführung von Programmen steuert. Die englische Bezeichnung ist Operating System (OS). Für eine ausführliche Darstellung sei auf [Stallings 1998] verwiesen, zudem bezieht sich Abschnitt 3.2.3 kurz auf Betriebssysteme unter dem Fokus eines *digitalen Langzeitarchivs*.

**Computerplattform** siehe *Rechnerplattform*.

**Containerdatei** wird besonders im Zusammenhang mit Virtualisierern oder Emulatoren verwendet. Die Bezeichnung stammt daher, dass sie nicht lediglich wie Verzeichnisse<sup>1</sup> Dateien zusammenfasst, sondern die logischen Strukturen abbildet, die sich auf realen blockorientierten Datenträgern, wie Festplatten finden.

**Darstellungspfad** siehe *View-Path*.

**Dateibetrachter** siehe *Betrachter*.

**Datenformat** beschreibt eine Spezifikation der Datenverarbeitung, welche definiert, wie Daten beim Laden, Speichern oder Verarbeiten programmtechnisch zu interpretieren sind. Datenformate sind nicht identisch mit Dateiformaten, da letztere sich aus verschiedenen Datenformaten zusammensetzen können. Zudem gibt es Datenformate für Datenströme, die nicht in Dateiform vorliegen müssen. Siehe außerdem Abschnitt 2.3.2.

---

<sup>1</sup>Vgl. hierzu Abschnitt 2.3.1 zu Dateien, Verzeichnissen und Dateisystemen.

**Datentyp** bezeichnet in der Sprache der Informatik die Zusammenfassung von Objektmen-gen mit den darauf definierten Operationen.

**Digitales Langzeitarchiv** bezeichnet formal die Organisationseiheit von technischen Sys-temen und Personal, deren Aufgabe in der Langzeitverfügbarkeit und dem -erhalt von *digitalen Objekten* liegt. Langzeitarchive können sich je nach den Bedürfnissen der (beauftragenden) Zielgruppe unterscheiden. Je nach Ausrichtung des Archivs spielt Vertrauenswürdigkeit der Integrität und Langzeiterhaltung der Daten [nestor 2006] eine zentrale Rolle. Die Zeiträume können dabei von gesetzlich vorgeschriebenen wenigen Jahrzehnten bis auf unbegrenzte Zeit reichen.

**Digitales Objekt** ist der Hauptgegenstand der Betrachtungen dieser Arbeit. Es liegt übli-cherweise in Form einer Datei als abgegrenzte Menge digitaler Daten auf Datenträgern vor. Bei Daten handelt es sich um maschinenles- und bearbeitbare Repräsentationen von Infor-mationen. Daher sind Kontext dieser Arbeit statische digitale Objekte durch den Transport von Inhalten definiert. Dynamische digitale Objekte besitzen als Eigenschaft die durch sie erreichbare Funktionalität, die üblicherweise als Computerprogramm bezeichnet wird. Zwi-schen diesen beiden Ausprägungen finden sich Mischformen. Der Abschnitt 2.3 setzt sich ausführlich mit digitalen Objekten auseinander.

Je nach Art des digitalen Objekts spricht man von Öffnen oder Betrachten einer Datei oder ihrer Ausführung. Ersteres trifft auf statische Inhalte, wie Bilder, Texte etc. zu. Zweites auf ausführbare Programme oder interaktive Komponenten, Abschnitt 3.2. Zudem wird eine Unterscheidung zwischen *Primär-* und *Sekundärobjekten* benötigt, Kapitel 7.

**Digitalisierung** zählt zu den Langzeitarchivierungsstrategien (siehe Abschnitt 3.1) analo-ger Objekte.

**DIP** bezeichnet das Dissemination Information Package als der das Archiv verlassende AIP (Digest). In den meisten Fällen kann es notwendig sein, das auf die Archivierung spezialisierte AIP in eine für den Nutzer handhabbare Form zu bringen. Die Umformung der AIPs in DIPs kann bedeuten, dass Änderungen an den Datenformaten vorgenommen werden, aber auch die Anpassung der beinhalteten digitalen Informationen an komfortable Nutzung. An diesem Punkt können Emulationsstrategien ansetzen: Ein digitales Objekt, welches auf aktuellen Rechnerplattformen nicht mehr nutzbar ist, wird durch Verfolgung eines *View-Paths* in einer Nutzungsumgebung so dargestellt oder ausgeführt, dass ein zukünftiger Benutzer damit etwas anfangen kann.

**Disk-Image** sind spezielle Dateien, welche die komplette Struktur und den vollständigen Inhalt von Speichermedien enthalten. Images können von Disketten, CD-ROMs, DVDs, Fest-platten und weiteren Medien, wie auch Hauptspeicher angefertigt werden. Das hat beispiels-weise den Vorteil viele physische Einzeldatenträger wie Disketten oder CD-ROMs auf einem einzigen Datenträger, wie einer Festplatte in einem Dateisystem zu versammeln. Disk-Images von Festplatten enthalten dann beispielsweise Strukturinformationen, wie Partitionierung. Da-durch bleiben Eigenschaften wie die Boot-Fähigkeit erhalten. Der Umfang eines Disk-Images

ist mindestens so groß, wie alle enthaltenen Dateien, da ein Disk-Image agnostisch gegenüber dem enthalten Dateisystem und Dateiinhalten ist und leere Dateisysteme nicht verkleinert.

**Dokument** ist im Sinne dieser Arbeit und im Sprachgebrauch der meisten Initiativen zur Langzeitarchivierung eine Untermenge *digitaler Objekte*, meistens statischen textbasierten Inhalts. Der Objekttyp kann dabei sehr unterschiedlich und von verschiedenen Datenformaten sein.

**Editor** meint Programme (Applikationen) mit denen *digitale Objekte* primär erstellt werden. Hierbei handelt es sich wie bei den Betrachtern auch, um *Software*, die auf eine bestimmte Ablaufumgebung für korrekte Funktion angewiesen ist.

**Emulation** ist eines der zentralen Themen dieser Arbeit. Sie bezeichnet ein erfolgversprechendes Konzepte zur Langzeitarchivierung digitaler Objekte. Im Fall einer Hardwareemulation oder in Virtualisierungslösungen läuft mehr als ein Betriebssystem auf einer gegebenen Hardware. Deshalb trifft man folgende Unterscheidung: Bei einer Host-Plattform bzw. dem Host-System handelt es sich um das System bestehend aus Hardware und Betriebssystem, auf der der Emulator als Applikation ausgeführt wird. Bei Virtualisierung spricht man meistens von der "Domain 0" kurz Dom 0.

Der Emulator als Applikation hat üblicherweise gewisse Anforderungen an seine Umgebung, wie spezielle Treiber oder Kernel-Module. Er benimmt sich jedoch überwiegend wie eine ganz normale Applikation eines gegebenen Betriebssystems.

Die im Emulator nachgebildete Umgebung bezeichnet man als Gast- oder aus Sicht der Langzeitarchivierung als Zielsystem. Dabei kommt es auf die Gastplattform an, ob lediglich eine bestimmte Hardware inklusive Firmware, wie bei den Home-Computern, oder komplette Umgebungen bestehend aus Hard- und Software ablaufen. Überblick und Einordnung verschafft zudem Abschnitt 3.5.

**Emulator** bezeichnet spezielle *Software*. Hierbei handelt es sich um *Applikationen*, die in der Lage sind *Ablaufumgebungen*, wie *Hardwareplattformen*, in derart geeigneter Weise in *Software* nachzubilden, so dass Software für diese Ablaufumgebung weitgehend so arbeitet, wie auf der ursprünglichen Plattform. Emulatoren bilden damit die Schnittstelle 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.

Durch diese Bedingungen grenzen Emulatoren Host- und *Gastsysteme* weitgehend voneinander ab. Gast- und Host teilen sich üblicherweise nicht gemeinsame Arbeits- oder Festspeicherbereiche. Das bedeutet, dass keine trivialen oder für den durchschnittlichen Anwender gewohnt komfortablen Datenaustauschmöglichkeiten zur Verfügung stehen. Ausnahmen bilden einige moderne Virtualisierer, die für bestimmte Gast- und Host-Systeme sogenannte "Shared Folders" (VMware) oder systemübergreifendes "Cut-and-Paste" anbieten.

Das generelle Konzept wird als *Emulation* bezeichnet.

**Erstellungsumgebung** beschreibt die zum Erstellungszeitraum des digitalen Objekts gültige und notwendige *Software*. Vergleiche hierzu [Rohde-Enslin 2004], S. 4, 5 und vergleiche zudem *Nutzungsumgebung*.

**Gastsystem** wird im Zusammenhang mit *Emulatoren* gebraucht. Es bezeichnet die virtuelle Hardware, die ein Emulator oder Virtualisierungswerkzeug in *Software* nachbildet. Siehe darüber hinaus *Host-System*.

**Gedächtnisorganisation** Dieser Begriff fasst üblicherweise Archive, Museen, Nationalbibliotheken oder Denkmalschützer zusammen. Diese kommen als Betreiber von digitalen, vertrauenswürdigen Langzeitarchiven in Frage. Sie unterliegen bestimmten gesetzlichen Regelungen, wie beispielsweise die Deutsche Nationalbibliothek.<sup>2</sup>

**Host-System** wird im Zusammenhang mit *Emulatoren* gebraucht. Es bezeichnet die konkret gegebene Softwareumgebung in der ein bestimmter Emulator als *Applikation* ausgeführt wird. *Referenzumgebungen* sind abstraktere Host-Systeme, in den durchaus sehr verschiedene Emulatoren und Viewer installiert sein können. Die im Emulator ablaufende Software beziehungsweise *Nutzungsumgebung* wird als Gast- oder *Zielsystem* bezeichnet.

**Ingest** bezeichnet die Aufnahme oder den Archiveingang von Archivalien. Mit der Aufnahme von digitalen Objekten sind bestimmte Prozesse verknüpft, wie die Beschreibung durch Metadaten oder die Feststellung der *Nutzungsumgebung*. Diese Prozesse werden im *OAIS* teilweise beschrieben und im Zuge dieser Arbeit näher spezifiziert.

**Metadaten** oder Metainformationen. Zu den primären *digitalen Objekten* oder auch Primärdaten (wissenschaftlicher Forschung) kommen in ihrem Lebenszyklus, wie der Produktion, Aufnahme ins Archiv oder Extraktion aus dem Archiv, weitere Daten hinzu. Diese dienen beispielsweise der Organisation von Datenbeständen. Sie helfen beim Auffinden, bei der Identifizierung oder auch Rekonstruktion der Primärdaten. Oft wird der Nachweis der Authentizität und Integrität oder auch die Zugriffskontrolle auf ein Objekt durch Metadaten realisiert.

Sie können als Bestandteil der logischen Einheit digitales Objekt betrachtet und sowohl gemeinsam, als auch getrennt vom Primärobjekt, verwaltet werden.

**Migration** bezeichnet im Sinne dieser Arbeit eines der zentralen Konzepte zur Langzeitarchivierung *digitaler Objekte*. Besonderer Augenmerk liegt auf der Verfügbarkeit der Objekte über längere Zeiträume hinweg mit den jeweils aktuellen Werkzeugen.

Mit der horizontalen Migration ist die Mitführung eines Objekts innerhalb des Entwicklungspfades seiner Erstellungsumgebung, *Editors* oder allgemein *Nutzungsumgebung* gemeint. Das bedeutet beispielsweise ein Objekt welches mit einer frühen Version von Microsoft Excel erstellt wurde, schrittweise in das Format der jeweils aktuellen Version zu überführen.

Vertikale Migration meint hingegen ein Objekt, welches mit einer bestimmten Applikation ursprünglich erstellt wurde, mit einer anderen Applikation mit vergleichbaren Eigenschaften

---

<sup>2</sup>Siehe hierzu [DNB 2006].

zu öffnen und dann im Format der neuen Applikation zu speichern. So kann beispielsweise ein Textdokument, welches mit Microsoft Word 95 erstellt wurde, in OpenOffice eingelesen und in dessen Format erneut abgespeichert werden.

Ausführlichere Anmerkungen liefert auch Abschnitt 3.4, ein Vergleich zur *Emulation* als Archivierungsstrategie bietet 3.6.

**Nutzungsumgebung** oder auch Ablaufumgebung meint die Rechnerplattform, ein Software Environment, welches auf einer bestimmten Rechnerarchitektur ausgeführt wird, in der auf ein digitales Objekt zugegriffen werden soll. Der Zugriff erfolgt nach dem OAIS-Modell üblicherweise seitens des Archivbenutzers. Die Nutzungsumgebung wird deshalb in den wenigsten Fällen mit der *Entstehungsumgebung* dieses Objekts identisch sein. Eine gute Einführung in diese Problematik gibt [Rohde-Enslin 2004], S. 4, 5.

**OAIS** Das Open Archival Information System ist ein vielzitiertes Referenzmodell für digitale Langzeitarchive. Eine ausführliche Darstellung bietet Abschnitt 3.1.1.

**Primärobjekt** Da zur Darstellung oder zum Ablaufenlassen eines *digitalen Objekts* oft weitere *Software* benötigt werden kann, die ebenso im Archiv gespeichert wird, bezeichnet in dieser Arbeit Primärobjekt das Objekt, das ursprünglich langzeitarchiviert werden sollte. Alternative Bezeichnungen sind Ziel- oder Target-Objekt.

So können die digitalen Objekte des Softwarearchivs bestehend aus Emulatoren, Betriebssystemen, *Applikationen*, Treibern und Hilfsprogrammen als Intermediär- oder *Sekundärobjekte* bezeichnet werden, da sie ohne das Primärobjekt nie ins Archiv eingestellt worden wären. Das Primärobjekt unterscheidet sich im Sinne dieser Arbeit von den Primärdaten [Sereiens und Hilf 2006] insofern, als dass mit Primärdaten üblicherweise Rohdaten wissenschaftlicher Forschungen oder Erhebungen gemeint sind.

**Rechnerarchitektur** Eine Hardwareplattform bezeichnet einen bestimmten Computertyp oder eine Familie von Computertypen. Sie ist hauptsächlich durch eine bestimmte Computer- bzw. Prozessorarchitektur bestimmt. Einen umfassenden Überblick zur Thematik findet sich in [Obershelp und Vossen 1998] im Abschnitt II oder bietet der Online-Auftritt des Computer-History-Museums [CHM 2008].

**Rechnerplattform** bezeichnet in Verbindung mit Computern eine Kombination aus Hardware und *Software*. Im allgemeinen Sprachgebrauch ist damit die Kombination von einem *Betriebssystem* mit einer bestimmten Hardware oder Rechnerarchitektur gemeint. Eine Plattform definiert somit ein System, auf dem ein Programm oder Applikation (als Teilmenge von Software) ablaufen kann.

**Referenzumgebung** oder auch Bezugsumgebung stellt einen fixierten Endpunkt für *View-Paths* dar. Aus den Darstellungspfaden folgen notwendige in das Softwarearchiv einzustellende Komponenten. Diese sind typischerweise sogenannte *sekundäre digitale Objekte*, da sie zur

Betrachtung oder zum Ablaufenlassen der *Primärobjekte* erforderlich sind. Eine Referenzumgebung kann im Sinne dieser Arbeit üblicherweise mehrere Host-Plattformen für verschiedene *Emulatoren* in einer Einheit zusammenfassen.

**Registry** oder Register bezeichnet spezielle Informationsdatenbanken beispielsweise zu *Daten-* oder *Dateiformaten*.

**Sekundärobjekt** meint ein oder mehrere dynamische digitale Objekte, die zur Betrachtung eines Zielobjekts notwendig sind. Ein Archivobjektbetrachter benötigt mindestens einen *Viewer*, der ebenso wie das für den *Viewer* benötigte Betriebssystem Bestandteil eines Softwarearchivs sein muss, da andernfalls ein geeigneter Zugriff auf das Zielobjekt, die Erstellung eines *View-Paths* fehlschlägt. Siehe zudem Kapitel 7.

**SIP** bezeichnet das Submission Information Package. Die angelieferten Primärobjekte werden mit zur Langzeiterhaltung notwendigen Metadaten angereichert, was im Zuge der Bestimmung des Objekttyps erfolgen kann. Hierbei kommen beispielsweise *Format-Registries* zum Einsatz. Siehe dazu zudem die Abschnitte 6.1 zur Archivaufnahme und 3.3 zu Metadaten.

**Software** Im Gegensatz zu den räumlich faßbaren und abgegrenzten Bestandteilen eines Datenverarbeitungssystems, die unter den Begriff *Hardware* zusammengefasst werden, bezeichnet man die Komponenten, die ausschließlich einen Informationsgehalt darstellen und nicht physikalisch greifbar sind, als Software.

Unter Software versteht man somit Programme oder Sammlungen von Programmen und (Funktions-)Bibliotheken, die zur Lösung von bestimmten Problemen erstellt wurden. Software besteht aus maschinenlesbarem Code, der Anweisungen für die Abarbeitung im Prozessor beinhaltet. Software wird daher immer aus dem Quellcode (üblicherweise eine höhere Programmiersprache) speziell für eine Rechnerarchitektur übersetzt (kompiliert). Der übersetzte Binärcode kann deshalb nur auf dem passenden Prozessortyp ablaufen und ist auf anderen Architekturen komplett sinnlos.

Eine Software besteht üblicherweise aus einer oder mehreren Dateien und bildet damit ein *dynamisches digitales Objekt*. Diese Objekte dienen beispielsweise als Werkzeuge der Erstellung und Bearbeitung statischer *statischer digitaler Objekte* und sind damit als *Sekundärobjekte* in der Langzeitarchivierung interessant. Als Computerspiel kann Software auch ein *Primärobjekt* sein.

**Target-Objekt** siehe *Zielobjekt*.

**Treiber** ist die direkte Übersetzung des englischen Begriffs *Driver*. Dieser bezeichnet *Software*, welche die Schnittstelle zwischen abstrakten APIs üblicherweise des *Betriebssystems* auf der einen Seite und einer konkreten Implementierung zum Ansprechen einer Hardware auf der anderen Seite, bildet. Treiber werden von den Herstellern der Hardwarekomponente, wie beispielsweise Grafikkarte, Audio- oder Netzwerkadapter für die verschiedenen Betriebssysteme erstellt.

**Viewer** synonym gebraucht für *Betrachter*.

**View-Path** oder Darstellungspfad stellt die notwendige *Software* und Zwischenschritte zusammen, um ein *Primärobjekt* auf einer aktuellen Rechnerplattform rendern zu können. Ziel ist dabei eine für das Objekt weitgehend vollständige *Nutzungsumgebung* wiederherzustellen. Mit dieser Problematik befasst sich Kapitel 6. Eine kompakte Einführung in dieses Konzept bietet zudem [van Diessen und Steenbakkers 2002].

**Virtualisierung** gibt es als Konzept in der Computertechnik schon lange. So trägt das IBM Mainframe-Betriebssystem "VM" – Virtual Machine den Begriff bereits im Namen. Das erste Mal zum Einsatz kam diese Technik auf dem S/360 Model CP67 vor 40 Jahren. In diesem System sorgte eine damals noch in Hardware umgesetzte Page-Translation-Tabelle für die Virtualisierung des Hauptspeichers, ebenso gab es virtuelle IO-Operationen und Interrupts.

Mit Virtualisierung werden Methoden zur Aufteilung der zentralen Hardwareressourcen einer Maschine, Arbeitsspeicher, Prozessor, Festspeicher und IO, wie Netzwerk, bezeichnet. Ziel ist es dabei, den Benutzern, also den Betriebssystemen eine Abstraktionsschicht zur Verfügung zu stellen, die ihn von der realen Hardware isoliert. Hierzu wird üblicherweise eine logische Softwareschicht eingeführt, um die physischen Gegebenheiten der Hardware zu verbergen. Den Nutzern der virtuellen Umgebung wird vorgespielt, dass sie Ressourcen exklusiv nutzen. Der Vorteil von Virtualisierung liegt in verschiedenen meist kommerziellen Aspekten begründet:

- Multiplexing der physikalischen Hardware hat das Ziel, mehr als ein Betriebssystem auf einer gegebenen Hardware ablaufen zu lassen.
- Isolation unterschiedlicher Applikationen voneinander
- Es können gleichzeitig komplett unterschiedliche Betriebssysteme
- Einfacheres Management der Server durch erleichterte Maschinenmigration über Hardwaregrenzen hinweg auch im laufenden Betrieb, das schnellere Setup von Servern, beispielsweise durch das Klonen von bestehenden Instanzen

Virtualisierung durch die Bereitstellung geeigneter Betriebssystemschnittstellen bietet neben Linux auch das aktuelle SUN OpenSolaris 10. Jedoch sind beide Ansätze zu spezifisch, um für eine Langzeitarchivierung von Interesse zu sein.



# Literaturverzeichnis

- [Adams und Agesen 2006] ADAMS, Keith ; AGESEN, Ole: A comparison of software and hardware techniques for x86 virtualization. In: ASPLOS-XII: Proceedings of the 12th international conference on Architectural support for programming languages and operating systems. New York, NY, USA : ACM, 2006. – ISBN 1-59593-451-0, S. 2-13
- [Adobe 2001] ADOBE: PDF Reference, Adobe Portable Document Format Version 1.4. Third Edition. Boston San Francisco Toronto Tokyo : Addison-Wesley, 2001. – ISBN 0-201-75839-3
- [AEP 2008] AEP: aep emulation page. WWW-Dokument, <http://www.aep-emu.de>. 2008. – [Online; letzter Aufruf 10.01.2008]
- [APA 2008] APA, Alliance for Permanent Access: PARSE.Insight - Permanent Access to the Records of Science. WWW-Dokument, <http://www.alliancepermanentaccess.eu>. 2008. – [Online; letzter Aufruf 16.01.2008]
- [Barham u. a. 2003] BARHAM, Paul ; DRAGOVIC, Boris ; FRASER, Keir ; HAND, Steven ; HARRIS, Tim ; HO, Alex ; NEUGEBAUER, Rolf ; PRATT, Ian ; WARFIELD, Andrew: Xen and the art of virtualization. In: SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles. New York, NY, USA : ACM, 2003. – ISBN 1-58113-757-5, S. 164-177
- [Baumgärtel 2002] BAUMGÄRTEL, Tillmann: Computerspiele: Und weg waren sie. In: Wochenzeitung - Die Zeit 52 (2002), S. 60-61
- [Beger 2005] BEGER, Gabriele: Langzeitarchivierung und Recht. In: Kulturelles Gedächtnis im 21. Jahrhundert, Tagungsband des internationalen Symposiums 23. April 2005. Universitätsbibliothek, Straße am Forum 2, D-76131 Karlsruhe : Universitätsverlag, 2005, S. 75-86
- [Bellard 2008] BELLARD, Fabrice: QEMU - open source processor emulator. WWW-Dokument, <http://fabrice.bellard.free.fr/qemu>. 2008. – [Online; letzter Aufruf 11.01.2008]
- [BMBF 2002] BMBF, Bundesministerium für Bildung und Forschung, Referat Öffentlichkeitsarbeit: Information Vernetzen - Wissen Aktivieren. WWW-Dokument, [http://www.bmbf.de/pub/information\\_vernetzen\\_wissen\\_aktivieren.pdf](http://www.bmbf.de/pub/information_vernetzen_wissen_aktivieren.pdf). 2002 (Strategisches Positionspapier). – [Online; letzter Aufruf 10.01.2008]

- [Borghoff u. a. 2005] BORGHOFF, Uwe M. [u. a.] ; KOMPETENZNETZWERK LANGZEIT-ARCHIVIERUNG (Hrsg.): Vergleich bestehender Archivierungssysteme. WWW-Dokument, [http://www.langzeitarchivierung.de/downloads/mat/nestor\\_mat\\_03.pdf](http://www.langzeitarchivierung.de/downloads/mat/nestor_mat_03.pdf). 2005 (nestor – materialien 3). – [Online; letzter Aufruf 01.02.2008]
- [Borghoff u. a. 2003] BORGHOFF, Uwe M. ; RÖDING, Peter ; SCHEFFCZYK, Jan ; SCHMITZ, Lothar: Langzeitarchivierung: Methoden zur Erhaltung digitaler Dokumente. Heidelberg : dpunkt.verlag, 2003. – ISBN 3–89864–245–3
- [Burnet und Supnik 1996] BURNET, Max ; SUPNIK, Bob: Preserving Computing's Past: Restoration and Simulation. In: Digital Technical Journal 8 (1996), Nr. 3, S. 23–38
- [Bárány 2006] BÁRÁNY, Balázs: Informationsverlust durch Digitalisierung. Wien : VDM Verlag Dr. Müller, 2006. – ISBN 978–3–86550–282–7
- [CCSDS 2002] CCSDS, Council of the Consultative Committee for Space Data Systems: Reference Model for an Open Archival Information System (OAIS). WWW-Dokument, <http://public.ccsds.org/publications/archive/650x0b1.pdf>. 2002. – [Online; letzter Aufruf 10.01.2008]
- [CHM 2008] CHM: Computer History Museum. WWW-Dokument, <http://www.computerhistory.org>. 2008. – [Online; letzter Aufruf 10.01.2008]
- [Choi und Rasmussen 2006] CHOI, Youngok ; RASMUSSEN, Edie: What do digital librarians do. In: JCDL '06: Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries. New York, NY, USA : ACM, 2006. – ISBN 1–59593–354–9, S. 187–188
- [CHSP 2007] CHSP: The Computer History Simulation Project. WWW-Dokument, <http://simh.trailing-edge.com>. 2007. – [Online; letzter Aufruf 12.01.2008]
- [CMUvA 2008] CMUvA: Computer Museum Universiteit van Amsterdam. WWW-Dokument, <http://www.science.uva.nl/museum>. 2008. – [Online; letzter Aufruf 10.01.2008]
- [Cooper und Garcia 2001] COOPER, Brian ; GARCIA, Hector: Creating trading networks of digital archives. In: JCDL '01: Proceedings of the 1st ACM/IEEE-CS joint conference on Digital libraries. New York, NY, USA : ACM Press, 2001. – ISBN 1–58113–345–6, S. 353–362
- [Cooper und Garcia-Molina 2002] COOPER, Brian F. ; GARCIA-MOLINA, Hector: Peer-to-peer data trading to preserve information. In: ACM Trans. Inf. Syst. 20 (2002), Nr. 2, S. 133–170. – ISSN 1046–8188
- [Coy 2006] COY, Wolfgang ; KOMPETENZNETZWERK LANGZEITARCHIVIERUNG (Hrsg.): Perspektiven der Langzeitarchivierung multimedialer Objekte. WWW-Dokument, <http://edoc.hu-berlin.de/series/nestor-materialien/5/PDF/5.pdf>. 2006 (nestor – materialien 5). – [Online; letzter Aufruf 01.02.2008]

- [CSM 2008] CSM, Das Computerspiele Museum im Förderverein für Jugend und Sozialarbeit e. V.: Computerspiele Museum. WWW-Dokument, <http://www.computerspielmuseum.de>. 2008. – [Online; letzter Aufruf 24.01.2008]
- [DBT 2007] DBT, Deutscher Bundestag: Unterrichtung durch die Bundesbeauftragte für die Unterlagen des Staatssicherheitsdienstes der ehemaligen Deutschen Demokratischen Republik. WWW-Dokument, <http://dip.bundestag.de/btd/16/058/1605800.pdf>. 2007 (Bundes-Drucksache 16/5800). – [Online; letzter Aufruf 11.01.2008]
- [van Diessen 2002] DIESSEN, Raymond van: Preservation Requirements in a Deposit System. PO Box 90407, 2509 LK The Hague, The Netherlands : IBM Netherlands, Amsterdam, 2002. – ISBN 90-6259-156-6
- [van Diessen und Lorie 2005] DIESSEN, Raymond van ; LORIE, Raymond A.: Long-Term Preservation of Complex Processes. In: Proceedings at the IS&T Archiving Conference, Washington, USA, 2005, 2005
- [van Diessen und van Rijnsoever 2002] DIESSEN, Raymond van ; RIJNSOEVER, Ben J.: Managing Media Migration in a Deposit System. PO Box 90407, 2509 LK The Hague, The Netherlands : IBM Netherlands, Amsterdam, 2002. – ISBN 90-6259-158-2
- [van Diessen und Steenbakkens 2002] DIESSEN, Raymond van ; STEENBAKKERS, Johan F.: The Long-Term Preservation Study of the DNEP project - an overview of the results. PO Box 90407, 2509 LK The Hague, The Netherlands : IBM Netherlands, Amsterdam, 2002. – ISBN 90-6259-154-X
- [van Diessen und van der Werf-Davelaar 2002] DIESSEN, Raymond van ; WERF-DAVELAAR, Titia van d.: Authenticity in a Digital Environment. PO Box 90407, 2509 LK The Hague, The Netherlands : IBM Netherlands, Amsterdam, 2002. – ISBN 90-6259-155-8
- [DIOSCURI 2008] DIOSCURI: DIOSCURI Java X86 Emulator. WWW-Dokument, <http://dioscURI.sourceforge.net>. 2008. – [Online; letzter Aufruf 10.01.2008]
- [DLR 2004] DLR, Projektträger Neue Medien in der Bildung + Fachinformation: digital library forum. WWW-Dokument, <http://www.dl-forum.de>. 2004. – [Online; letzter Aufruf 10.01.2008]
- [DNB 2006] DNB, Deutsche Nationalbibliothek: Gesetz über die Deutsche Nationalbibliothek (DNBG). WWW-Dokument, <http://www.d-nb.de/wir/pdf/dnbg.pdf>. 2006. – [Online; letzter Aufruf 10.01.2008]
- [DNB 2008] DNB, Deutsche Nationalbibliothek: dnb, Deutsche Nationalbibliothek. WWW-Dokument, <http://www.d-nb.de>. 2008. – [Online; letzter Aufruf 10.01.2008]
- [Dobratz und Tappenbeck 2002] DOBRATZ, Susanne ; TAPPENBECK, Inka: Thesen zur Zukunft der digitalen Langzeitarchivierung in Deutschland. In: Bibliothek 26 (2002), Nr. 3, S. 257-261

- [Dooijes 2000] DOOIJES, E.H.: Old computers, now and in the future. WWW-Dokument, [http://www.science.uva.nl/museum/pdfs/oldcomputers\\_dec2000.pdf](http://www.science.uva.nl/museum/pdfs/oldcomputers_dec2000.pdf). 2000. – [Online; letzter Aufruf 10.01.2008]
- [DPC 2001] DPC: Digital Preservation Coalition. WWW-Dokument, <http://www.dpconline.org>. 2001. – [Online; letzter Aufruf 10.01.2008]
- [EMUU 2008] EMUU: Emulators Unlimited. WWW-Dokument, <http://www.emuunlim.com>. 2008. – [Online; letzter Aufruf 10.01.2008]
- [Endres und Fellner 2000] ENDRES, Albert ; FELLNER, Dieter W.: Digitale Bibliotheken. Heidelberg : dpunkt.verlag, 2000. – ISBN 3-932588-77-0
- [Factor u. a. 2007] FACTOR, Michael ; NAOR, Dalit ; RABINOVICI-COHEN, Simona ; RAMATI, Leeat ; RESHEF, Petra ; SATRAN, Julian: The need for preservation aware storage: a position paper. In: SIGOPS Oper. Syst. Rev. 41 (2007), Nr. 1, S. 19–23. – ISSN 0163-5980
- [Gabriel und Ribeiro 2001] GABRIEL, David ; RIBEIRO, Cristina: A Metadata Model for Multimedia Databases. WWW-Dokument, <http://www.phil.uni-sb.de/werner/ichim.pdf>. 2001. – [Online; letzter Aufruf 10.01.2008]
- [Gladney und Lorie 2005] GLADNEY, H.M. ; LORIE, Raymond A.: Trustworthy 100-year digital objects: durable encoding for when it's too late to ask. In: ACM Trans. Inf. Syst. 23 (2005), Nr. 3, S. 299–324. – ISSN 1046-8188
- [Goebel und Scheller 2004] GOEBEL ; SCHELLER ; KOMPETENZNETZWERK LANGZEIT-ARCHIVIERUNG (Hrsg.): Digitale Langzeitarchivierung und Recht. WWW-Dokument, <http://nbn-resolving.de/urn/resolver.pl?urn=urn:nbn:de:0008-20040916022>. 2004 (nestor – materialien 1). – [Online; letzter Aufruf 11.01.2008]
- [Granger 2000] GRANGER, Stewart: Emulation as a Digital Preservation Strategy. In: D-Lib Magazine 6 (2000), Nr. 10. – ISSN 1082-9873
- [Hank u. a. 2007] HANK, Carolyn ; CHOEMPAYONG, Songphan ; SHEBLE, Laura: Blogger perceptions on digital preservation. In: JCDL '07: Proceedings of the 2007 conference on Digital libraries. New York, NY, USA : ACM, 2007. – ISBN 978-1-59593-644-8, S. 477–477
- [Hedstrom und Lampe 2001] HEDSTROM, Margaret ; LAMPE, Clifford: Emulation vs. Migration: Do Users Care? In: DigiNews 5 (2001), Nr. 6, S. 2. – ISSN 1093-5371
- [Hedstrom und Montgomery 1998] HEDSTROM, Margaret ; MONTGOMERY, Sheon ; RESEARCH LIBRARIES GROUP (Hrsg.): Digital Preservation Needs and Requirements in RLG Member Institutions. WWW-Dokument, <http://www.oclc.org/programs/ourwork/past/digpresneeds/digpres.pdf>. 1998. – [Online; letzter Aufruf 07.02.2008]

- [Hedstrom u. a. 2003] HEDSTROM, Margaret ; ROSS, Seamus ; OTHERS ; NSF-DELOS (Hrsg.): Invest to Save - Report and Recommendations of the NSF-DELOS Working Group on Digital Archiving and Preservation. WWW-Dokument, <http://delos-noe.iei.pi.cnr.it/activities/internationalforum/Joint-WGs/digitalarchiving/Digitalarchiving.pdf>. 2003. – [Online; letzter Aufruf 10.01.2008]
- [Heinze 2005] HEINZE, Michael ; INSTITUT FÜR ZUKUNFTSSTUDIEN UND TECHNOLOGIEBEWERTUNG BERLIN, IZT (Hrsg.): Fallstudie: Digitale Langzeitarchivierung. WWW-Dokument, [http://www.izt.de/pdfs/Studie\\_Digitale\\_Langzeitarchivierung.pdf](http://www.izt.de/pdfs/Studie_Digitale_Langzeitarchivierung.pdf). 2005. – [Online; letzter Aufruf 10.01.2008]
- [van der Hoeven 2007] HOEVEN, Jeffrey van d.: Dioscuri: emulator for digital preservation. In: D-Lib Magazine 13 (2007), Nr. 11/12. – ISSN 1082-9873
- [van der Hoeven u. a. 2007] HOEVEN, Jeffrey van d. ; LOHMAN, Bram ; VERDEGEM, Remco: Emulation for Digital Preservation in Practice: The Results. In: The International Journal of Digital Curation 2 (2007), Nr. 2, S. 123–132. – ISSN 1746-8256
- [van der Hoeven u. a. 2005] HOEVEN, J.R. van d. ; DIESSEN, R.J. van ; MEER, K. van d.: Development of a Universal Virtual Computer (UVC) for long-term preservation of digital objects. In: Journal of Information Science 31 (2005), Nr. 3, S. 196–208. – ISSN 0165-5515
- [van der Hoeven und van Wijngaarden 2005] HOEVEN, J.R. van d. ; WIJNGAARDEN, H.N. van: Modular emulation as a long-term preservation strategy for digital objects. In: International Web Archiving Workshop 2005 (IWAW'05), 2005. – [Online; letzter Aufruf 10.01.2008]
- [Holdsworth und Sergeant 2000] HOLDSWORTH, David ; SERGEANT, Derek M.: A Blueprint for Representation Information in the OAIS Model. In: NASA Goddard Conference on Mass Storage Systems and Technologies, 2000
- [Holdsworth und Wheatley 2001] HOLDSWORTH, David ; WHEATLEY, Paul: Emulation, Preservation and Abstraction. In: RLG DigiNews 5 (2001), Nr. 4
- [Holdsworth und Wheatley 2004] HOLDSWORTH, David ; WHEATLEY, Paul: Long-Term Stewardship of Globally-Distributed Representation Information. In: 12th NASA Goddard, 21st IEEE Conference on Mass Storage Systems and Technologies April 13-16. College Park, Maryland, USA, 2004, S. 101–114
- [Huth 2004] HUTH, Karsten: Probleme und Lösungsansätze zur Archivierung von Computerprogrammen – am Beispiel der Software des Atari VCS 2600 und des C64. WWW-Dokument, [http://www.digitalgamearchive.org/data/news/Softw\\_Preserv\\_huth.pdf](http://www.digitalgamearchive.org/data/news/Softw_Preserv_huth.pdf). 2004. – [Online; letzter Aufruf 16.02.2008]

- [HZV 2007] HZV, Heise Zeitschriften Verlag: Heise Online - News. WWW-Dokument, <http://www.heise.de>. 2007. – [Online; letzter Aufruf 10.01.2008]
- [innotek AG 2008] INNOTEK AG: Innotek. WWW-Dokument, <http://www.innotek.de>. 2008. – [Online; letzter Aufruf 20.05.2008]
- [JaJa 2006] JAJA, Joseph: Robust technologies for automated ingestion and long-term preservation of digital information. In: dg.o '06: Proceedings of the 2006 international conference on Digital government research. New York, NY, USA : ACM, 2006, S. 285–286
- [JPCP 2007] JPCP, Subdepartment of Particle Physics: The JPC Project - Computer Virtualization in Java. WWW-Dokument, <http://www-jpc.physics.ox.ac.uk>. 2007. – [Online; letzter Aufruf 10.01.2008]
- [KB 2007] KB, Koninklijke Bibliotheek: The archiving system for electronic publications: The e-Depot. WWW-Dokument, <http://www.kb.nl/dnp/e-depot/dm/dm-en.html>. 2007. – [Online; letzter Aufruf 10.02.2008]
- [kopal 2008] KOPAL: kopal - Co-operative Development of a Long-Term Digital Information Archive. WWW-Dokument, <http://kopal.langzeitarchivierung.de>. 2008. – [Online; letzter Aufruf 23.01.2008]
- [Kwon u. a. 2006] KWON, Hyuckbin ; PARDO, Theresa A. ; BURKE, G. B.: Building a state government digital preservation community: lessons on interorganizational collaboration. In: dg.o '06: Proceedings of the 2006 international conference on Digital government research. New York, NY, USA : ACM, 2006, S. 277–284
- [Lehmann 1996] LEHMANN, Klaus-Dieter: Das kurze Gedächtnis digitaler Bibliotheken. In: Zeitschrift für Bibliothekswesen und Bibliografie 3 (1996), S. 209–226
- [Levy 1998] LEVY, David M.: Heroic measures: reflections on the possibility and purpose of digital preservation. In: DL '98: Proceedings of the third ACM conference on Digital libraries. New York, NY, USA : ACM Press, 1998. – ISBN 0–89791–965–3, S. 152–161
- [LoC 2008] LoC, Library of Congress: PREMIS - Preservation Data Maintenance Activity. WWW-Dokument, <http://www.loc.gov/standards/premis>. 2008. – [Online; letzter Aufruf 25.01.2008]
- [Lorie 2002] LORIE, Raymond: The UVC: a Method for Preserving Digital Documents - Proof of Concept. PO Box 90407, 2509 LK The Hague, The Netherlands : IBM Netherlands, Amsterdam, 2002. – ISBN 90–6259–157–4
- [Lorie 2001] LORIE, Raymond A.: Long term preservation of digital information. In: JCDL '01: Proceedings of the 1st ACM/IEEE-CS joint conference on Digital libraries. New York, NY, USA : ACM Press, 2001. – ISBN 1–58113–345–6, S. 346–352

- [Lorie 2002] LORIE, Raymond A.: A methodology and system for preserving digital data. In: JCDL '02: Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries. New York, NY, USA : ACM, 2002. – ISBN 1-58113-513-0, S. 312–319
- [Maniatis u. a. 2003] MANIATIS, Petros ; ROSENTHAL, David S. H. ; ROUSSOPOULOS, Mema ; BAKER, Mary ; GIULI, TJ ; MULIADI, Yanto: Preserving peer replicas by rate-limited sampled voting. In: SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles. New York, NY, USA : ACM, 2003. – ISBN 1-58113-757-5, S. 44–59
- [McCallum 2005] MCCALLUM, Sally H.: MARCXML Sampler. WWW-Dokument, <http://www.ifla.org/IV/ifla71/papers/175e-McCallum.pdf>. 2005. – [Online; letzter Aufruf 25.01.2008]
- [Mellor u. a. 2002] MELLOR, Phil ; WHEATLEY, Paul ; SERGEANT, Derek ; ECDL (Hrsg.): Migration on Request: A practical technique for digital preservation. WWW-Dokument, <http://www.si.umich.edu/CAMILEON/reports/migreq.pdf>. 2002. – [Online; letzter Aufruf 11.01.2008]
- [MESS 2008] MESS: Multi Emulator Super System. WWW-Dokument, <http://www.mess.org>. 2008. – [Online; letzter Aufruf 20.02.2008]
- [Messmer 1997] MESSMER, Hans-Peter: PC-Hardwarebuch - Aufbau, Funktionsweise, Programmierung. 4. Auflage. Bonn : Addison Wesley, 1997. – ISBN 3-8273-1086-5
- [Mittler 2005] MITTLER, Elmar: Langzeitarchivierung digitaler Objekte in Bibliotheken. In: Kulturelles Gedächtnis im 21. Jahrhundert, Tagungsband des internationalen Symposiums 23. April 2005. Universitätsbibliothek, Straße am Forum 2, D-76131 Karlsruhe : Universitätsverlag, 2005, S. 87–98
- [Muir 2001] MUIR, Adrienne: Legal deposit of digital publications: a review of research and development activity. In: JCDL '01: Proceedings of the 1st ACM/IEEE-CS joint conference on Digital libraries. New York, NY, USA : ACM, 2001. – ISBN 1-58113-345-6, S. 165–173
- [nestor 2006] NESTOR ; KOMPETENZNETZWERK LANGZEITARCHIVIERUNG (Hrsg.): Kriterienkatalog vertrauenswürdige digitale Langzeitarchive. WWW-Dokument, <http://edoc.hu-berlin.de/series/nestor-materialien/2006-8/PDF/8.pdf>. 2006 (nestor – materialien 8). – [Online; letzter Aufruf 10.01.2008]
- [nestor 2008] NESTOR: Kompetenznetzwerk Langzeitarchivierung. WWW-Dokument, <http://www.langzeitarchivierung.de>. 2008. – [Online; letzter Aufruf 23.01.2008]
- [Obershelp und Vossen 1998] OBERSHHELP, Walter ; VOSSEN, Gottfried: Rechneraufbau und Rechnerstrukturen. 7. überarb. Auflage. München, Wien : R. Oldenbourg Verlag, 1998. – ISBN 3-486-24288-1

- [Oltmans u. a. 2004] OLTMANS, Erik ; DIESSEN, Raymond van ; WIJNGAARDEN, Hilde van: Preservation functionality in a digital archive. In: JCDL '04: Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries. New York, NY, USA : ACM Press, 2004. – ISBN 1-58113-832-6, S. 279–286
- [PADI 2008] PADI: Preserving Access to Digital Information. WWW-Dokument, <http://www.nla.gov.au/padi>. 2008. – [Online; letzter Aufruf 10.01.2008]
- [PLANETS 2008] PLANETS: PLANETS - Digital Preservation Research and Technology. WWW-Dokument, <http://www.planets-project.eu>. 2008. – [Online; letzter Aufruf 25.01.2008]
- [Pomerantz u. a. 2007] POMERANTZ, Jeffrey ; OH, Sanghee ; WILDEMUTH, Barbara M. ; YANG, Seungwon ; FOX, Edward A.: Digital library education in computer science programs. In: JCDL '07: Proceedings of the 2007 conference on Digital libraries. New York, NY, USA : ACM, 2007. – ISBN 978-1-59593-644-8, S. 177–178
- [QUMRANET 2007] QUMRANET: Kernel Based Virtual Machine. WWW-Dokument, <http://kvm.qumranet.com/kvmwiki>. 2007. – [Online; letzter Aufruf 10.02.2008]
- [von Ranke-Graves 1984] RANKE-GRAVES, Robert von: Griechische Mythologie - Quellen und Deutung. 39.-42. Tausend, Neuauflage in einem Band Juli 1993. Reinbek bei Hamburg : Rowohlt Taschenbuch Verlag, 1984. – ISBN 3-499-55404-6
- [Reichherzer und Brown 2006] REICHHERZER, Thomas ; BROWN, Geoffrey: Quantifying software requirements for supporting archived office documents using emulation. In: JCDL '06: Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries. New York, NY, USA : ACM Press, 2006. – ISBN 1-59593-354-9, S. 86–94
- [Rohde-Enslin 2004] ROHDE-ENSLIN, Stefan: Nicht von Dauer - Kleiner Ratgeber für die Bewahrung digitaler Daten in Museen. Berlin : Institut für Museumskunde, 2004. – ISBN 1860-4641
- [Rothenberg 1995] ROTHENBERG, Jeff: Ensuring the Longevity of Digital Information. In: Scientific American 272 (1995), Nr. 1, S. 42–47
- [Rothenberg 1996] ROTHENBERG, Jeff: Metadata to Support Data Quality and Longevity. In: Proceedings of the 1st IEEE Metadata Conference, NOAA Complex, Silver Spring, Md., 16-18 April, 1996
- [Rothenberg 1999] ROTHENBERG, Jeff: Ensuring the Longevity of Digital Information. WWW-Dokument, <http://www.clir.org/pubs/archives/ensuring.pdf>. 1999. – [Online; letzter Aufruf 27.01.2008]
- [Rothenberg 2000] ROTHENBERG, Jeff: An Experiment in Using Emulation to Preserve Digital Publications. WWW-Dokument, <http://nedlib.kb.nl/results/emulationpreservationreport.pdf>. 2000. – [Online; letzter Aufruf 14.01.2008]



- [Rothenberg 2002] ROTHENBERG, Jeff: Avoiding Technological Quicksand: Finding a Viable Technical Foundation for Digital Preservation. In: The State of Digital Preservation: An International Perspective, Conference Proceedings, Documentation Abstracts, Inc., Institutes for Information Science, Washington, D.C., April 24-25. 1755 Massachusetts Avenue, NW, Suite 500, 2002, S. 1–32
- [Rödig u. a. 2003] RÖDIG, Peter ; BORGHOFF, Uwe M. ; SCHEFFCZYK, Jan ; SCHMITZ, Lothar: Preservation of digital publications: an OAIS extension and implementation. In: DocEng '03: Proceedings of the 2003 ACM symposium on Document engineering. New York, NY, USA : ACM Press, 2003. – ISBN 1–58113–724–9, S. 131–139
- [Sereiens und Hilf 2003] SEREIENS, Thomas ; HILF, Eberhard R.: Digital Preservation Testbed White Paper - Emulation: context and current status. WWW-Dokument, [http://www.digitaleduurzaamheid.nl/bibliotheek/docs/White\\_paper\\_emulationUK.pdf](http://www.digitaleduurzaamheid.nl/bibliotheek/docs/White_paper_emulationUK.pdf). 2003 (Digital Preservation Testbed Project). – [Online; letzter Aufruf 10.01.2008]
- [Sereiens und Hilf 2006] SEREIENS, Thomas ; HILF, Eberhard R. ; KOMPETENZ-NETZWERK LANGZEITARCHIVIERUNG (Hrsg.): Studie zum Stand vorhandener Forschungsdaten und Rohdaten aus wissenschaftlicher Tätigkeit: Erfordernisse und Eignung zur Archivierung bzw. Zurverfügungstellung in Deutschland (Primärdaten). 2006 (nestor – materialien 6). – [Online; letzter Aufruf 10.02.2008]
- [Spilker 2007] SPILKER, Heiko: Datenspeicher: Das Gedächtnis der Kultur. In: Technology Review 9 (2007), Nr. 0, S. 94–95. – ISSN 1613–0138
- [Stallings 1998] STALLINGS, William: Operating Systems - Internals and Design Principles. 3rd Edition. Upper Saddle River, New Jersey 07458, USA : Prentice Hall, 1998. – ISBN 0–13–887407–7
- [Steenbakkens 2000] STEENBAKKERS, Johan F. ; KONINKLIJKE BIBLIOTHEEK (Hrsg.): Setting up a Deposit System for Electronic Publications. The NEDLIB Guidelines.. Bd. 5. PO Box 90407, 2509 LK The Hague, The Netherlands : NEDLIB Consortium, 2000. – ISBN 906259149–3
- [Strodl u. a. 2007] STRODL, Stephan ; BECKER, Christoph ; NEUMAYER, Robert ; RAUBER, Andreas: How to choose a digital preservation strategy: evaluating a preservation planning procedure. In: JCDL '07: Proceedings of the 2007 conference on Digital libraries. New York, NY, USA : ACM Press, 2007. – ISBN 978–1–59593–644–8, S. 29–38
- [von Suchodoletz und Welte 2006] SUCHODOLETZ, Dirk von ; WELTE, Randolph: Funktionale Langzeitarchivierung der X86 Ära. In: Praxis der Informationsverarbeitung und Kommunikation (PiK) 4 (2006), S. 221–227. ISBN 0930–5157
- [Tansley u. a. 2003] TANSLEY, Robert ; BASS, Mick ; STUVE, David ; BRANSCHOFISKY, Margret ; CHUDNOV, Daniel ; MCCLELLAN, Greg ; SMITH, MacKenzie: The DSpace institutional digital repository system: current functionality. In: JCDL '03: Proceedings

- of the 3rd ACM/IEEE-CS joint conference on Digital libraries. Washington, DC, USA : IEEE Computer Society, 2003. – ISBN 0-7695-1939-3, S. 87–97
- [Thibodeau 2002] THIBODEAU, Kenneth: Overview of Technological Approaches to Digital Preservation and Challenges in Coming Years. In: The State of Digital Preservation: An International Perspective, Conference Proceedings, Documentation Abstracts, Inc., Institutes for Information Science, Washington, D.C., April 24-25. 1755 Massachusetts Avenue, NW, Suite 500, 2002, S. 4–31
- [Thorns 2007] THORNS, Fabian: Das Virtualisierungs-Buch. Workstation Server Rechenzentrum. Zavelsteiner Straße 20, 71034 Böblingen : Computer & Literatur, 2007. – ISBN 978-3936546-43-9
- [TNA 2007] TNA, The National Archives: The technical registry PRONOM. WWW-Dokument, <http://www.nationalarchives.gov.uk/pronom>. 2007. – [Online; letzter Aufruf 10.01.2008]
- [Tsai u. a. 2006] TSAI, Janice Y. ; CRANOR, Lorrie F. ; CRAVER, Scott: Vicarious infringement creates a privacy ceiling. In: DRM '06: Proceedings of the ACM workshop on Digital rights management. New York, NY, USA : ACM, 2006. – ISBN 1-59593-555-X, S. 9–18
- [Verdegem und van der Hoeven 2006] VERDEGEM, Remco ; HOEVEN, Jeffrey van d.: Emulation: To be or not to be. In: IS&T Conference on Archiving 2006, Ottawa, Canada, May 23-26, 2006, S. 55–60
- [VMware Inc. 2008] VMWARE INC.: Virtualization, Virtual Machine & Virtual Server Consolidation - VMware. WWW-Dokument, <http://www.vmware.com>. 2008. – [Online; letzter Aufruf 12.01.2008]
- [Vögler 2007] VÖGLER, Max: Geschichte im Netz: Praxis, Chancen, Visionen. In: Beiträge der Tagung .hist 2006, 2007. – ISBN 1612-5940, S. 208–215
- [Ward 2003] WARD, Jewel: A quantitative analysis of unqualified dublin core metadata element set usage within data providers registered with the open archives initiative. In: JCDL '03: Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital libraries. Washington, DC, USA : IEEE Computer Society, 2003. – ISBN 0-7695-1939-3, S. 315–317
- [Waugh u. a. 2000] WAUGH, Andrew ; WILKINSON, Ross ; HILLS, Brendan ; DELL'ORO, Jon: Preserving digital information forever. In: DL '00: Proceedings of the fifth ACM conference on Digital libraries. New York, NY, USA : ACM Press, 2000. – ISBN 1-58113-231-X, S. 175–184
- [Welte 2008] WELTE, Randolph: Funktionale Langzeitarchivierung digitaler Objekte – Entwicklung eines Demonstrators zur Internet-Nutzung emulierter Ablaufumgebungen. 2008. – Arbeitsgruppe am Lehrstuhl für Kommunikationssysteme, Universität Freiburg

- [van der Werf 2000] WERF, Titia van d. ; KONINKLIJKE BIBLIOTHEEK (Hrsg.): The Deposit System for Electronic Publications: A Process Model. Bd. 6. PO Box 90407, 2509 LK The Hague, The Netherlands : NEDLIB Consortium, 2000. – ISBN 906259150–7
- [van der Werf 2002] WERF, Titia van d.: Experience of the National Library of the Netherlands. In: The State of Digital Preservation: An International Perspective, Conference Proceedings, Documentation Abstracts, Inc., Institutes for Information Science, Washington, D.C., April 24-25. 1755 Massachusetts Avenue, NW, Suite 500, 2002, S. 54–64
- [Wheatley 2001] WHEATLEY, Paul ; ARIADNE (Hrsg.): Migration - a CAMiLEON discussion paper. WWW-Dokument, <http://www.ariadne.ac.uk/issue29/camileon>. 2001. – [Online; letzter Aufruf 10.01.2008]

# Index

- AACS, 243
- ABC
  - 80/800/802, 111
  - 80, 226
- Abdeckung, 53
- Abhängigkeit, 143
- Ablage, 185
- Ablaufumgebung, 153, 245
- Abschnitt, 9
- Abschreiben, 4
- Abspielgerät, 24
- Abstraktionsschicht, 252
- Access, 42
- Acorn
  - Atom, 227
  - Electron, 111
- Actionscript, 69
- Adobe, 50
- Advanced Access Content System, 243
- Adventure Vision, 111
- AIFF, 149
- AIP, 42, 47, 167, 245
- Akte, 12, 95
- Alltagsgegenstand, 11
- Alltagskultur, 16, 94
- Altair, 226
- Alternative, 53, 123, 207, 236
- AMD, 80, 84, 114
- Amiga, 83, 86, 91, 111, 227
- Amstrad, 111, 227
- Anforderung, 169, 240, 248
  - Hardware, 99
  - Host-System, 99
  - Nutzungsumgebung, 101
- Anhang, 225
- Animation, 17, 53, 215
- Anleitung, 92, 187
- Anwendung, 48, 49, 68, 187, 219, 245
- API, 70, 71, 191
  - Microsoft, 71
- Apple, 83, 88, 134, 173
  - I, 111
  - II, 84, 111
  - III, 111
  - Macintosh, 89, 227
  - Macintosh Plus, 89
  - PowerPC, 89
- Appletalk, 46
- Applikation, 48, 49, 153, 182, 184, 187, 245
  - Nachbildung, 67
  - Netz-, 220
- Aquarius, 111
- ARAnyM, 112
- Arbeitsgerät, 12, 28
- Arbeitsplatz, 17, 220
- Arbeitsspeicher, 101, 252
- Arbeitsumgebung, 64, 169
- Arcade, 81, 228
- Arcadia 2001, 111
- Archäologie, 11, 19, 94, 215
  - Software-, 153
- Architektur, 71, 250
  - Unabhängigkeit, 131
- Archiv, 30, 93, 95, 171, 245, 249
  - Aufgabenstellung, 12
  - Aufnahme, 249
  - Benutzer, 151
  - Beschreibung, 51
  - Bestand, 213
  - Definition, 12, 245
  - digitales, 5, 41
  - Hintergrund-, 201
  - Inhalt, 51
  - Management, 5, 42, 96, 169, 199

- Online-, 202
- Software-, 181, 184
- traditionelles, 12
- Typ, 171
- Archival Information Package, 245
- Archivalie, 12, 245
- Archivar, 9, 215
- Archivierung
  - Langzeit-, 245
  - Ziel, 77
- Archivmanagement, 147, 216
- ARM, 190
- Artefakt, 11
- ASCII, 20, 26, 57, 149, 187
- Atari, 86
  - 400, 111
  - 800, 111
  - 2600, 168, 228
  - 5200, 111
  - 7800, 111
  - Falcon, 112
  - Jaguar, 111, 228
  - ST, 112, 227
  - TT, 112, 227
- Atom, 111
- Audience, 55
- audience, 53
- Audio, 25, 45, 53, 187, 191, 213, 251
- Aufbau, 9
- Aufbewahrung, 12, 202, 245
- Aufwand, 46, 157
- Aufzeichnung, 238
  - Dichte, 239
  - Verfahren, 239
- Ausgabe
  - 2D, 173
  - 3D, 123, 133, 173
- Ausgabesystem, 165
- Ausgangspunkt, 153
- Ausstellung, 5, 94, 110, 163
- Auswahlkriterium, 77, 91
  - nichttechnisch, 96
  - technisch, 99
- Authentizität, 35, 45, 55, 62, 93, 249
- AVI, 187
- BASIC, 190
- Basilisk, 89
- Bauwerk, 11
- BBC, 227
  - Micro, 111
- Bedeutung, 11, 212, 245
- Bedienung, 160
- Begriff, 65, 245
- Benutzer, 147
  - Interaktion, 143, 165
  - Interface, 94, 99, 125, 132, 186
- BeOS, 102
- Berechnung, 15
- Bericht, 95
- Beschreibung, 92
- Betrachtung, 24, 246
- Betriebssystem, 26, 48, 49, 70, 99, 102, 103, 153, 160, 172, 182, 190, 207, 246, 252
  - ausgelaufen, 193
  - BeOS, 102, 193
  - BSD, 195
  - kommerziell, 193
  - Linux, 102, 195
  - Open-Source, 195
  - OS/2, 102, 193
  - Windows, 102
- Bewahrung, 4, 24
- Bewertung, 69, 217
- Bibliothek, 32, 93, 103, 172, 249
  - dynamische, 197
  - GTK, 120
  - National-, 32
  - Programm, 49
  - Software-, 184
  - Universitäts-, 32
- Bibliothekar, 9
- Bild, 53
- Bildschirm, 24, 101, 165, 187
  - Auflösung, 165
- Binary Translation, 114
- BIOS, 77, 85, 101, 119, 139, 191
- Bitstrom, 3, 18, 19, 23, 203
- Blockgerät, 176
- Blu-ray, 210, 243

- BMBF, 30
- Bochs, 84, 135
  - BIOS, 133, 137
- Boot-Fähigkeit, 207
- BSD, 132, 157
- Buch, 11, 19, 25, 45
- Buchhaltung, 51
- Bytestream, 149
  
- C64, 29, 66, 86, 113, 168
- CAMILEON, 35
- CASPAR, 32, 34
- Castor, 135
- CD-ROM, 27, 112, 134, 174, 203, 205, 231, 242, 247
  - virtuelle, 177, 208, 212
- CEDAR, 44
- CGA, 101, 140
- CHIP8, 227
- Cirrus Logic, 133
- Civilization, 68
- CLOOP, 133
- CMS, 51
- CoCo 2, 227
- Codec, 187
- Coleco
  - Adam, 227
  - Vision, 111
- Colour Genie, 111, 227
- Commodore
  - 64/128, 86, 111, 113, 212, 219, 227, 235
  - x0xx, 111
- Compact Disk, 242
- Compis, 111
- Component Library, 107
- Computer, 11, 14
  - Geschichte, 28, 80, 225
  - Museum, 13
  - Netzwerk, 174
  - Nutzung, 47
  - Plattform, 246
- Computer History Simulator, 80
- Computerkunst, 16, 50, 215
- Computerspiel, 14, 49, 68, 82, 139, 189, 215, 235
  - Museum, 16, 234
- Concept, 111
- Container, 168, 176
  - Datei, 142, 209, 246
  - Format, 119, 124
- Content Information, 245
- Content Management System, 35, 51
- contributor, 55
- coverage, 53
- COW, 133
- CPC, 227
- CPU, 71, 99
- creator, 55
- Cultural Heritage, 215
- Cygwin, 70, 229
- Cyrix, 84
  
- Daemon-Tools, 209, 236
- DAI Personal Computer, 111
- Darstellung, 217
  - 2D, 165
  - 3D, 165
  - Pfad, 147, 246, 250, 252
- Data Link Layer, 175
- dataset, 53
- Datei, 19, 57, 247
  - Art, 20
  - Betrachter, 246
  - Endung, 149
  - Name, 210
- Dateisystem, 23, 177, 209
- Daten
  - Ablage, 21
  - Austausch, 100, 124, 174, 184, 248
  - Export, 100
  - Format, 46, 53, 116
  - Import, 100
  - klassische, 47
  - Roh-, 250
  - technische, 53
  - Transport, 169, 174
- Datenbank, 23, 35, 51, 167
- Datenformat, 5, 46, 53, 246
  - Migration, 60
  - proprietäres, 21

- Datenkapsel, 168
- Datennetz, 220
- Datenträger, 18, 23, 25, 27, 174, 176, 202, 212, 216, 238
  - analoger, 47
  - Blu-ray, 243
  - Diskussion, 46
  - DVD, 242
  - Festplatte, 240
  - HD DVD, 242
  - Installations-, 203, 231
  - magneto-optischer, 244
  - Migration, 203
  - Obsoleszenz, 46
  - optischer, 241
  - Umkopieren, 203
  - virtueller, 5, 116, 206
- Datentyp, 19, 246
- Datenverarbeitung, 17
- Datum, 155
  - Änderungs-, 55
  - Erstellungs-, 55
  - Veröffentlichungs-, 55
- DCC, 35
- DCMI, 53
  - Type Vocabulary, 53
- dd, 167, 197, 205, 207
- DEC, 190, 226
  - Alpha, 134
- Definition, 18
- Dekodierung, 187
- DELOS, 32
- description, 53
- Deutsche Nationalbibliothek, 30, 93, 249
- DFG, 30, 94
- DIAS, 5, 30, 44, 72, 93, 153
- Dienst, 173
  - web-basierter, 163
- Digest, 147, 247
- DigiCULT, 32
- Digital Equipment Corporation, 226
- digital library forum, 30
- Digital Millenium Act, 213
- Digital Preservation Coalition, 35
- Digital Rights Management, 211, 218
- Digitale Revolution, 13
- Digitalisierung, 11, 45, 163, 247
- DIN EN ISO 3166, 53
- Dioscuri, 84, 135, 140, 173, 211
- DIP, 42, 147, 216, 217, 247
- DirectX, 71, 123, 192
- Disk-Image, 247
- Diskette, 205–207, 239
  - Image, 176, 207, 236
  - Installations-, 205
  - ZIP/LS120, 240
- Diskettenlaufwerk, 124, 239
- Dissemination Information Package, 247
- DMG, 133
- DNB, 30, 93, 94, 213, 249
- DOI, 53
- Dokument, 19, 53, 95, 248
  - Erstellung, 189
- Domain, 248
- Doom, 96, 236
- DOS, 102, 138, 207, 228, 236
- DOSBOX, 84, 96, 139, 236
- DOSEMU, 84, 130
- DPC, 35
- DPE, 35
- Drag&Drop, 126, 179
- Dragon 32/64, 227
- DRM, 163, 211, 243
- DROID, 149
- Druckerpresse, 13, 24
- Druckerzeugnis, 11
- DSpace, 44
- Dublin Core, 30, 53, 90, 202
- DVD, 174, 205, 242, 247
  - virtuelle, 177, 208
- E-Learning, 28
- Ebene, 182, 201, 212
- Echtheit, 55
- ECPA, 32
- Editor, 187, 248
- EDSAC, 226
- EFI, 88, 191
- EGA, 101, 140
- Eingabegerät, 134, 197

- Einheit, 155
- Einordnung, 55, 212
- Email, 14
- Emulation, 3, 35, 41, 46, 248
  - Ansatz, 72, 107
  - Applikation, 64
  - Ebene, 72
  - geschachtelte, 105, 169
  - Hardware-, 71
  - Stacked, 105, 235
  - unvollständige, 66
  - Vergleich, 72
  - Vollständigkeit, 102
- Emulator, 182, 184, 195, 216, 225, 248
  - ARAnyM, 112
  - Auswahl, 77
  - Betriebssystem, 229
  - Bochs, 135
  - Commodore-, 113
  - Dioscuri, 135
  - DOSBOX, 139
  - Großrechner, 80
  - Hybrid, 84
  - JPC, 137
  - Langzeitverfügbarkeit, 103
  - MESS, 110
  - Migration, 169
  - modularer, 110
  - PowerPC, 89
  - QEMU, 131
  - Recherche, 83
  - ScummVM, 229
  - Spezifikation, 107
  - verfügbarer, 79
  - X86, 80, 131
  - ZX Spectrum, 113
- Endpunkt, 159
- Entscheidungsbaum, 154
- Entstehungsumgebung, 3, 248
- Enzyklopädie, 28
- Epoche, 16, 49, 94
- Erbe
  - kulturelles, 32
- Erfolgsbedingung, 5, 181, 215
- Erhaltung, 11
- ERPANET, 32
- Ersatzteil, 24, 25
- Ersteller, 92
- Erstellungsumgebung, 249
- Ethernet, 101, 112
- European Commission on Preservation and Access, 32
- EVM, 106, 107
- Experiment, 167, 230
- Extensible Firmware Interface, 191
- Fahrplan, 51
- Famicom, 111
- Farbtiefe, 165
- FAUmachine, 84
- Fehler, 98
- Feiertag, 155
- Fernseher, 25, 82
- Fernzugriff, 163
- Festplatte, 23, 191, 240, 247
  - virtuelle, 124, 168, 178, 209
- File Transfer Protocol, 175
- Film, 19, 45, 53
- Filmindustrie, 17
- Firmware, 77, 112, 134, 193
- Flash, 50, 69, 189, 244
- Flexibilität, 143
- Floppy Image, 176
- Format, 53, 93, 172
- Formatierung, 119, 178
- Forschung, 15, 172
  - Einrichtung, 94
  - Ergebnis, 94
- Fotografie, 25, 53, 173
- Fragestellung, 5, 9, 11
- Freelords, 68
- FTP, 175, 232
- Funktionsmodell, 28
- FunVision, 111
- Galaksija, 111
- Game Console, 87, 96, 228
- GameBoy, 82, 88, 111, 228
- GameCube, 82, 113
- Gastsystem, 248, 249



- GDFR, 149
- Gedächtnisorganisation, 3, 30, 42, 93, 249
- Geisteswissenschaft, 4
- General Public License, 211
- genisoimage, 197
- Geschäftsvorgang, 95
- Gesetzliche Regelung, 93, 249
- GIF, 149
- Glossar, 245
- Google, 12, 173, 228
- GPL, 126, 211
- Grafik, 197, 251
  - Adapter, 101, 126, 143
  - Ausgabe, 173, 219, 236
  - Tablett, 101
- Großrechner, 89, 164, 220
- Grundmenge, 53
- Gutenberg, 24
- GwDG, 30
  
- Höhlenzeichnung, 11
- Haltbarkeit, 24, 61, 173, 238
- Handbuch, 187, 236
- Hardware, 94, 202
  - Auswahl, 101
  - Museum, 28
  - Treiber, 102, 190
- Hardwaremuseum, 212
- HDMI, 213
- Herausforderung, 11, 213, 218
- Hercules, 89, 101, 140
- Hersteller, 79, 92, 225
- Hewlett-Packard, 226
- Hilfsmittel, 16
- Hilfsprogramm, 184, 196, 207
- Home-Computer, 16, 82, 109, 110, 154, 190, 225
- Host-Plattform
  - DOS, 226
  - Linux, 84, 226
  - Unix, 226
  - Windows, 84, 226
- Host-System, 100, 248, 249
- HP, 226
  - 2114, 2115, 2116, 2100, 21MX, 226
  - HP-41, 226
  - HP-48, 226
- HTML, 26, 55, 60, 149
- Hypervisor, 114
  
- IBM, 30, 226
  - IBM 1130, 226
  - IBM 1401, 226
  - IBM 7090/7094, 226
  - IBM System 3, 226
  - PC 08/16, 111
  - PC 10/27/82, 111
  - PC AT, 111
  - PC AT 386, 111
  - PC XT, 111
  - S/360, 252
  - S/370, 89
  - S/390, 89, 134, 190
- IBM-PC, 84
- Ideenaustausch, 11
- identifizier, 53
- Immutable Disk, 128
- Indigo2, 111
- Indy, 111
- Informatik, 9, 14, 30
- Information, 3, 11, 13, 215, 245
  - Portal, 30
  - Technik, 3
  - Träger, 4
  - Transport, 11
  - Verlust, 45
- Ingest, 42, 147, 166, 249
- Inhalt, 53, 207, 243, 246
- Initiative, 53, 218, 248
  - außereuropäische, 35
  - europaweite, 32
  - Forschungs-, 30
  - Langzeitarchivierung, 30
- Innotek, 127
- Installation
  - CD, 208, 231
  - Diskette, 207
- Institution, 55
- Integrität, 249
- Intel, 80, 84, 114, 149

- AC97, 124, 129
- BX Chipset, 120
- PentiumPro, 80
- TX Chipset, 119
- Intellivision, 111
- interactive resource, 53
- Internet, 220
  - Auftritt, 51
- IPv4, 175
- IPX, 175
- IPX/SPX, 46, 189
- IRIS Indigo, 111
- IronBabel, 113
- ISO
  - 639, 53
  - 9660, 177, 208
  - 14721, 42
- ISO-Image, 100, 124, 168, 177, 197, 231
- Itanium, 80
- Java, 50, 67, 69, 84, 135, 149
  - Virtual Machine, 67
- Java PC Emulator, 85, 137
- Java Runtime Environment, 136
- Java Virtual Machine, 113, 135
- JHOVE, 30, 149
- Joystick, 101
- JPC, 85, 137
- JPEG, 149, 187
- JRE, 136
- Jupiter Ace, 111
- JVM, 113, 135
- Kapazität, 205, 240, 244
- Kapitel
  - Aufteilung, 9
- Karte, 12
- Katalog, 163
- KC 85/3,4, 111
- Kernel, 117
  - Modul, 118, 132, 175
- Klassifikation, 147
- Klassifizierung, 55
- Kloster, 4
- Kodierung, 53, 187
- Kohl, 17
- Kompatibilität, 80, 140, 240
- Komponente, 182, 193, 213
- Kompression, 45, 198
  - verlustbehaftete, 45
  - verlustfreie, 45
- Konsole
  - entfernte, 123
  - Spiele, 111, 229
  - Spiele-, 81
- Kontakt, 92
- Kontext, 26, 147
- Konvertierung, 26, 58, 132
- Kopal, 5, 30, 57, 200
- Kopie, 4
  - analoge, 24
  - digitale, 24
- Kopierschutz, 206, 218, 243
- Korrespondenz, 14
- Kosten, 99, 115, 128, 157
  - Reduktion, 17
- Kultur, 4, 16
  - Geschichte, 12
- Kunst, 16, 172
- KVM, 132
- Langlebigkeit, 26, 208
- Language, 53
- Langzeitarchiv, 47, 218, 247
- Langzeitarchivierung, 5, 41, 213
  - digitale, 91
  - Strategie, 42, 247
- Laser, 111, 241
- Laserlock, 209
- Lebenszyklus, 55
- Lesbarkeit, 25
- Library of Congress, 11, 35, 57
- Lifecycle, 169
- Linux, 102, 139, 160, 190, 228, 229
  - Kernel, 70
- Linux VServer, 70
- Lizenz, 55, 96, 112, 128, 163, 187
  - Kosten, 115, 124
  - Prüfung, 189
  - Schlüssel, 213

- Lochstreifen, 89, 164, 165, 179
- LOCKSS, 200
- Lokalisierung, 165, 186
- lomount, 133, 178
- LS120, 239
- Lucas Arts, 189, 229
  
- Mönch, 4
- Maß, 155, 186
- Mac-OS, 157
  - classic, 64
  - X, 64, 86
- Macintosh, 131, 227
- MADRAS, 35
- Magnetband, 23, 164, 179, 244
- Mainboard, 101, 124, 209
- Mainframe, 14, 84, 89, 225, 252
- Malerei, 25
- MAME, 88, 197, 225
- MARC, 30, 55
- Maschine
  - Sprache, 186
  - virtuelle, 162
- Massenware, 14
- Maus, 101, 123, 132
- Mean Time between Failures, 61
- Medium, 27, 53
  - analoges, 45, 47
  - Archiv, 173
  - digitales, 238
- MESS, 110, 225, 236
- Metadata Object Description Standard, 57
- Metadaten, 20, 41, 46, 51, 90, 149, 167, 185, 187, 207, 210, 213, 249
  - Format, 57
- Metainformation, 51, 249
- Metrik, 155, 187, 217
- METS, 57
- Microfiche, 61
- Microsoft, 85, 125, 173, 213
- Migration, 3, 41, 46, 58, 140, 249
  - Archivierungsstrategie, 73
  - Aufwand, 63
  - Begriff, 58
  - Datenträger, 61
  - Emulator-, 104
  - horizontale, 59
  - Probleme, 62
  - Vergleich, 72
  - vertikale, 59, 187
  - Vorteile, 73
- Mikroelektronik, 14
- Miniaturisierung, 16, 82
- MIPS, 113, 134, 190
- MIT, 44
- MITS Altair, 226
- mkisofs, 197
- Mobiltelefon, 80, 81, 229
- MODS, 57
- Monitor-Interface, 132, 236
- Motif, 119
- Motivation, 9, 70
- Motorola
  - 68000, 83
  - 68040, 112
- moving image, 53
- MP3, 96
- MPG, 94
- MRBIOS, 126
- MSX, 227
- MTBF, 61
- Multimedia, 28, 50, 213
- Museum, 5, 30, 94, 110, 163, 172, 187, 249
  - Computer-, 28
  - Computerspiele-, 234
- Myst, 213
  
- Nachhaltigkeit, 3, 79, 218
- Nachwelt, 11, 19
- Name, 92, 99
  - Datei-, 23
- NARA, 44
- Nationalarchiv, 93
  - Britisches, 149
  - Niederländisches, 84
  - Schwedisches, 32
- NDAD, 44
- NE2000, 134, 195
- Nebenbedingung, 5, 167
- NES, 88, 228

- Nestor, 5, 30, 213
- NetBIOS, 101, 175
- Network Layer, 175
- Network Mode, 133
- Netzwerk, 19, 131, 174, 191, 220, 232, 252
  - Protokoll, 175
- Netzwerkkarte, 129, 143, 251
  - DEC-Tulip, 126
  - NE2000, 133, 193
  - PCNet32, 193
  - Realtek, 133
- Nintendo, 88, 111, 165
- NUMA, 114
- Nutzbarkeit, 5, 98, 131, 212
- Nutzergruppe, 171
- Nutzungsumgebung, 250
  - vorbereitete, 166, 185
- OAIS, 5, 34, 42, 147, 216, 250
- Objekt
  - abstraktes, 18
  - analoges, 25, 247
  - Aufnahme, 42, 147
  - Ausgabe, 42, 147
  - Beschreibung, 53
  - digitales, 3, 18, 247, 248
  - dynamisches, 47
  - interaktives, 48
  - konzeptionelles, 19
  - logisches, 19
  - nicht-archiviertes, 153
  - physikalisches, 18
  - Primär-, 250
  - primäres, 187
  - Sekundär-, 181, 250, 251
  - sekundäres, 187
  - statisches, 28, 47
  - Titel, 53
  - Typ, 51, 53, 99, 147, 149, 171, 248
  - Zugriff, 151
- Obsoleszenz
  - Hardware, 66
- OCA, 11
- Office, 21, 59, 67
- OGG, 187
- Online Ressource, 35
- Open Content Alliance, 11
- Open Source, 96, 131, 216, 226
- OpenGL, 101, 123, 133, 192
- OpenOffice, 157, 249
- OpenVZ, 70
- Operating System, 246
- Opteron, 80
- Organisation, 95, 173
- Oric, 227
- Originalzustand, 3
- OS, 246
- OS/2, 102
- Pac-Man, 68
- Pacifica, 114
- Packungsdichte, 24
- PADI, 35
- Palm OS, 229
- PANDORA, 44
- Papier, 4, 45, 61
- Papyrus, 4, 27
- Parallels
  - Workstation, 124
- PARSE, 94
- Partitionierung, 119, 247
- PC-Architektur, 80
- PCNet32, 193
- PDA, 81, 165
- PDF, 30, 46, 149, 157, 168, 198
- PDI, 245
- PDP, 226
- PearPC, 89
- Pentium, 119, 124, 135
- Performance, 98, 114, 131, 164
- Peripherie, 49, 71, 101, 143, 216
- Perl, 67, 189
- person, 55
- Personal Computer, 14, 84
- Perspektive, 173
- Pfad
  - Datei-, 21
  - Delimiter, 165
- Pflichtabgabe, 213
- Philips P2000, 227

- Philosoph, 4
- physical object, 53
- Physik, 15
- Plan, 12
- PLANETS, 32, 84, 109, 135
- Plattform
  - Computer-, 246
  - Spiel-, 82
- Plex86, 85
  - BIOS, 137
- PLM, 153
- PNG, 187
- Politik, 35
- Pollux, 135
- Portabilität, 98
- Pose, 64
- Postscript, 168, 198
- PowerPC, 113, 131, 134, 172, 190
- Präferenz, 155
- PREMIS, 30, 57, 202
- Preservation Description Information, 245
- Preservation Layer Model, 153
- Preservation Manager, 153, 217
- Preservation Planning, 42, 169, 216
- Primärobjekt, 9, 147, 153, 187, 212, 250
  - kombiniertes, 166
- Privatperson, 96, 173
- Problemstellung, 210
- Produkt, 71
  - Geschichte, 96
  - Palette, 115
  - Reife, 98
  - Zyklus, 96
- Programm, 26, 172, 182, 187
- Programmablauf, 132
- Programmiersprache, 98
- Projekt, 96
  - Bochs, 135
  - CASPAR, 32
  - Dioscuri, 135
  - Langzeitarchivierung, 30
  - PLANETS, 32
  - QEMU, 131
- PRONOM, 102, 149, 173, 211
- Prozessor, 71, 99, 252
- Publikation, 15
- publisher, 55
- PUID, 149
- PXE, 121, 133
- Pyramide, 4, 11
- Python, 67, 189
- QCOW(2), 133
- QEMU, 85, 131, 143, 197
  - qemu-img, 132, 142, 167, 197
- Qualitätsverlust, 24, 63, 211
- Quellcode, 96, 98, 195, 226, 245, 251
- Quelle, 55, 172, 208
- Rückkopplung, 155
- Rahmenbedingung, 93, 218
- Randbedingung, 3, 25, 71, 169
- RAW, 133
- rawrite, 197, 207
- ReactOS, 70, 229
- Rechenanlage, 17
- Recherche, 79, 163, 225
- Rechnerarchitektur, 49, 250
- Rechnerplattform, 250
  - Arcade, 87, 228
  - Game-Computer, 87, 226, 228
  - Großrechner, 80, 226
  - Home-Computer, 82, 86, 226
  - Intel PC, 84
  - Klasse, 80
  - Mainframe, 80, 226
  - Spiel-, 82
  - Unix Workstation, 80
  - X86, 84
- Rechnung, 96
- Recht, 55
- Rechtssystem, 211
- Redundanz, 8, 200, 217
- reference, 53
- Referenzumgebung, 147, 154, 159, 162, 195, 215, 250
- Reflexion, 241
- Regeneration
  - Datenträger, 61
- Relation, 169

- Objekt-, 55
- Revolution, 27
  - industrielle, 11
  - informationstechnische, 24
- rights, 55
- RMPS, 209
- ROM, 190
- Rothenberg, 27
- RSS, 55
- Ruby, 67
- Runtime Environment, 189, 195
  
- S3 Trio32/64, 126
- Safedisc, 209
- Sammelauftrag, 93
- Sammlung, 53
- SAP, 51
- Scanner, 45
- Schöpfung, 211
- Schallplatte, 3, 241
- Schicht, 182
- Schnittstelle, 92, 103, 238
- Schröder, 17
- Schrift, 4
- Schriftart, 184
- Schutz, 96
- Schutzfrist, 211
- Schutzvorrichtung, 168, 213
- Scientific Data Systems, 226
- SCSI, 101, 191
- ScummVM, 229
- SDL, 128, 132, 139, 197
- SDS 940, 226
- Second Life, 17, 165, 220
- Securom, 209
- Seefahrt, 11
- SegaMaster, 228
- Sekundärobjekt, 9, 187, 250, 251
- service, 53
- Shared Folder, 129, 179
- Shared Library, 197
- Sicherheit, 99
- signature, 55
- Silverlight, 50
- Sinclair
  - 1000, 227
  - Z88, 227
  - ZX, 227
  - ZX 81, 165
  - ZX Spectrum, 185, 227
  - ZX-Spectrum, 165
  - ZX81, 227
- SIP, 42, 147, 213, 251
- Skriptsprache, 189
- Skulptur, 25
- SMB, 232
- SNES, 88, 111, 228
- Software, 59, 251
  - Bibliothek, 98, 99, 197
  - kommerziell, 96
  - Komponente, 184
  - modulare, 187
  - Open Source, 96
  - Variante, 186
- Softwarearchiv, 5, 112, 217
- Solaris, 157
- Solars 10, 252
- Sondertaste, 164
- Sony PlayStation, 87, 111, 228, 229
- Sound, 191, 197
- Soundblaster, 126, 133, 140
- Sparc, 134, 190
- Speicher, 99, 101
  - Medium, 241
  - Typ, 244
- Speichersystem, 24
- Speicherung, 24, 185
- Spiel
  - Arcade-, 16
  - Civilization, 68
  - Computer-, 16, 68, 81
  - Doom, 96
  - Freelords, 68
  - Konsole, 64
  - Online-, 189
  - Pac-Man, 68
  - Sammlung, 213
  - Video-, 81
  - Warlords, 68
- Sprache, 53, 92

- natürliche, 184
- Schrift-, 11
- Spur, 241
- Stabilität, 98
- Standard, 41, 42, 51, 173, 175
  - Datenformat, 21
- Standardisierung, 46
- Steuererklärung, 96
- still image, 53
- Strategie, 3, 42, 181, 215
- Struktur, 9
- subject, 53
- Submission Information Package, 251
- SUN Microsystems, 83, 127, 134, 136
- SVM, 114
- SWsoft, 85, 124
- Symbian, 229
- Systemuhr, 101
- Tabellenkalkulation, 21, 67
- Tandy
  - TRS-80, 111
- Tastatur, 101, 124, 138, 165, 231
  - Eingabe, 132
  - Mehrfachbelegung, 165
  - Sprache, 165
- Taste
  - Modifier, 165
- TCP/IP, 46, 101, 133, 189, 232
- Technikfolgenabschätzung, 30
- Technologie, 47
- Teilchenbeschleuniger, 15
- Telefon, 13
  - Mobil-, 219, 229
- Teleskop, 15
- Text, 53
- Textverarbeitung, 21, 59, 246, 249
- Theater, 4
- TI-81, 227
- TIFF, 30, 149, 187
- Titel, 53
- title, 53
- Tontafel, 27
- Transport Layer, 175
- Treiber, 101, 143, 181, 184, 190, 213
  - Beschaffung, 193
  - Quellen, 193
- TRS-80 Models I-IV, 227
- Turing Machine, 226
- UAE, 227
- Umgebung, 159
  - Ablauf, 245
  - Entstehungs-, 249
- UML, 65
- UNESCO, 11
- Uniform Resource Identifier, 92
- UNIVAC, 80
- Universal Virtual Machine, 106
- Unterhaltung, 50, 68, 87, 228
- Unternehmen, 14, 95, 173
- Unvollständigkeit, 141
- URI, 19, 23, 92
- URN, 53
- UTF8, 60, 149, 187
- UVM, 106, 107
- Vanderpool Technology, 114
- Variante, 23, 64, 100, 186
- VAX, 226
- VBoxManage, 128
- VBoxSDL, 128
- VBoxVRDP, 128
- Vergleich, 140, 215, 230
- Verifikation, 35, 63
- Verkehrsträger, 51
- Verlag, 55, 92
- Verteilung, 200, 236
  - Software-, 117
- Vervielfältigung, 11, 45
- Verwaltung, 17, 96
- Verzeichnis, 21, 246
- VESA, 134, 173
- VGA, 101, 140, 191
- VHS, 25
- Video, 12, 20, 24, 25, 30, 45, 53, 96, 173, 187, 213, 251
- View-Path, 91, 147, 187, 215, 217, 252
  - Aggregation, 156
  - Anzahl, 157

- Kosten, 157
- Zusammenfassung, 156
- Viewer, 147, 157, 160, 182, 187
- Virtual Machine, 252
  - Monitor, 114
- Virtual PC, 86, 131, 140
  - 2004, 125
  - 2007, 125
  - PC-Architektur, 86
  - PPC, 85
- Virtual Server 2005, 125
- VirtualBox, 86, 127, 197
- Virtualisierer, 167
  - DOSEMU, 130
  - kommerzieller, 115, 124
  - Open Source, 126
  - Parallels Workstation, 124
  - QEMU, 126
  - Virtual PC, 125
  - Virtual Server, 125
  - VirtualBox, 127
  - X86, 114
  - XEN, 126
- Virtualisierung, 65, 143, 162, 206, 248
  - Begriff, 65, 252
  - X86, 114
- VMDK, 128, 133
- VMM, 114
- VMware, 86, 115, 140
  - Converter, 143, 167
  - Fusion, 123
  - GSX-Server, 123
  - Host-Plattform, 118
  - Player, 86, 123
  - Server, 86, 123
  - Workstation, 86, 115, 142
  - Workstation 3.2, 119
  - Workstation 4.0, 120
  - Workstation 4.5, 121
  - Workstation 5.0, 121
  - Workstation 5.5, 122
  - Workstation 6.0, 122, 142
- vmware-cmd, 123
- vmware-mount.pl, 119, 133, 178, 197
- vmware-vdiskmanager, 119
- VNC, 131
- Von Neumann, 107
- Vorgang, 95
- VPC, 133
- VTx, 114
- VVFAT, 133
- Währung, 155, 186
- Warlords, 68
- WAV, 149
- Web-Browser, 50, 125, 132, 138, 159
- Wechseldatenträger, 100
- Welt, 11
  - virtuell, 17
- Wiederherstellung, 153
- Windows, 64, 102, 160, 190, 228, 229
  - 3.11, 205
- Wine Project, 70, 229
- Wirtschaft, 17, 35, 173
- Wissen, 211, 213
- Wissenschaft, 15, 172
- Wissensumsatz, 3, 11
- Workstation, 14, 160
- World Digital Library, 11
- World of Warcraft, 17, 220
- WWW, 50, 53, 220
- X86-PC, 80, 91, 109, 191
- XEN, 65, 114
- XML, 46, 57, 59, 60, 149
- Zeit, 186
- Zeitablauf, 169
- Zeitschrift, 11
- Zeitspanne, 55
- Zeitung, 11
- Zielgruppe, 55
- Zielsystem, 248
- ZIP, 239
- Zusammenfassung, 215
  - thematische, 166
- Zuverlässigkeit, 98