Informationsverlust durch die Digitalisierung

Diplomarbeit

zur Erlangung des Magistergrades der
Philosophie
an der Fakultät für Sozialwissenschaften
der Universität Wien

eingereicht von

Balázs Bárány

Wien, November 2004

Vorwort

Die Information – dieses Etwas, weder Materie noch Energie, und so einflußreich in unserer Welt – fasziniert mich schon länger. Der Wunsch, mich mit den unterschiedlichen Aspekten der Information zu beschäftigen, hat auch die Zusammenstellung meines Studiums inklusive Fächerkombination motiviert.

In meiner ersten längeren beruflichen Tätigkeit, während der Entwicklung einer medizinischen Software, sah ich mich erstmals mit der Geschwindigkeit der Entwicklung der Computerindustrie konfrontiert. Sie geht so schnell voran, daß es schon große Mühe kostet, mit ihr nur Schritt zu halten. Daten, Programme und Methoden müssen ständig erneuert werden, weil ihr Umfeld sich ständig ändert.

Meine Studienzeit fiel in die Jahre des Internet-Booms und des darauffolgenden Platzens der „Seifenblase“. Trends und Websites, alle als „die Zukunft der Kommunikation“ ausgerufen, kamen und gingen; bestehen blieb nur der Alltag mit seinen „alltäglichen“ Aufgaben, wie z. B. der Sicherung des Zugangs zu Informationen, auch wenn diese lediglich drei bis fünf Jahre alt oder noch jünger waren. In dieser Zeit wurde mir zum ersten Mal bewußt, wie schwach eines der Fundamente unserer heutigen Gesellschaft und Kultur ist. Das Wissen um diese Problematik brachte mich dann dazu, erste Nachforschungen zum Thema anzustellen und letztendlich zu beschließen, „Informationsverlust durch die Digitalisierung“ als Diplomarbeitsthema zu wählen.

Während der Recherchen lernte ich interessante und kompetente Leute kennen, die mir nützliche Hinweise lieferten. Andere hörten von mir zum ersten Mal über die Problematik und änderten daraufhin teilweise ihren Umgang mit digitalen Daten.

Ich möchte mich an dieser Stelle bei allen Menschen bedanken, die durch Tun oder Unterlassen direkt oder indirekt das Zustandekommen der Arbeit ermöglichten; insbesondere denjenigen, die sich die große Mühe machten, sich durch den ganzen Text oder Teile davon durchzuarbeiten und hilfreiche Kommentare abzugeben.

Inhaltsverzeichnis

1 Einleitung

Ein immer größerer Teil unseres Wissens und unserer Kultur ist digital. Das bedeutet, daß die geistigen Schöpfungen häufig gleich im Computer entstehen, meist am Computer weiterbearbeitet und wiederum über Computer(netzwerke) an andere Menschen weitergegeben werden.

Diese Informationen sind sehr fragil; die meisten von uns haben schon einmal unabsichtlich ein wichtiges Dokument gelöscht oder eine Diskette nicht mehr lesen können.

Diese Fragilität hängt mit vielen, im Weiteren zu beschreibenden Eigenschaften der digitalen Technologie (bzw. ihrer heutigen Ausprägung) zusammen, und ist sehr schwer zu vermeiden. Einzelne Autoren sprechen schon von „digitalem Alzheimer“ [Siet02] oder einem „digitalen dunklen Mittelalter“ [Embe02].

Die wissenschaftliche Öffentlichkeit wurde erstmals 1995 mit dem Artikel „Ensuring the Longevity of Digital Documents“ von Jeff Rothenberg im Scientific American [Roth95a] auf das Problem aufmerksam gemacht; Rothenberg präsentiert dort auch einen Lösungsvorschlag. Seitdem werden in den damit befaßten Kreisen die unterschiedlichen Methoden der digitalen Langzeitverfügbarkeit diskutiert.

In der Öffentlichkeit und der Industrie ist kaum ein Bewußtsein für die Problematik vorhanden. Das behindert die Verbreitung besserer Technologien und Handlungsweisen, mit denen der Informationsverlust verlangsamt oder vermieden werden könnte.

1.1 Über diese Arbeit

Kapitel 1 dient als Einleitung. Es enthält genaue Definitionen der Begriffe, um die es in der Arbeit geht, und beschreibt überblicksmäßig die Problematik sowie die öffentliche und wissenschaftliche Sicht des Problems. Anschließend führt es einige Beispiele an, in denen wichtige digital gespeicherte Informationen verlorengingen.

In Kapitel 2 werden die Forschungsfragen und Hypothesen vorgestellt.

Kapitel 3 beschreibt die Elemente und Methoden der digitalen Speicherung von Information, u. a. Datenträger, Dateiformate und Klassen von Dateiformaten sowie Software. Kapitel 4 gibt dann für die beschriebenen Themen ihre für die langfristige Speicherung relevanten Aspekte an.

In Kapitel 5 werden die Lösungsansätze für das Problem und ihre rechtlichen Rahmenbedingungen vorgestellt. Es wird auch beschrieben, unter welchen Voraussetzungen die Lösungsansätze anwendbar sind und welche Probleme es mit ihnen gibt, die ihre Anwendung in der Praxis verhindern können.

Kapitel 6 enthält die Schlußfolgerungen in Form von Antworten auf die Forschungsfragen und die Bestätigung der Hypothesen.

Im Anhang befinden sich noch in Kapitel 7 die Beschreibungen verschiedener Experimente, die u. a. die Eignung der vorgestellten Methoden für einige Arten von Daten zeigen oder widerlegen sollen.

1.2 Begriffsbestimmungen

In dieser Arbeit geht es um Digitalisierung, Information und ihren Verlust. Leider bezeichnen „Information“ und auch „Digitalisierung“ in unterschiedlichen Zusammenhängen unterschiedliche Dinge, weswegen sie (und der Vollständigkeit halber auch „Informationsverlust“) definiert werden müssen.

1.2.1 Information

Laut Duden Fremdwörterbuch (Bd. 5, 7. Auflage, Mannheim 2001) bedeutet Information „Nachricht, Mitteilung, Hinweis; Auskunft; Belehrung, Aufklärung;“, oder in der Informatik auch den „Gehalt einer Nachricht, die aus Zeichen eines Kodes zusammengesetzt ist“.

Eine eigene Wissenschaft, die Informationstheorie, beschäftigt sich mit der Information. Einer der Begründer der Informationstheorie, Claude Shannon, beschreibt Information als Auswahlmöglichkeit aus verschiedenen Elementen einer Grundmenge [Shan93, S. 214], die nicht immer eine Bedeutung hätten bzw. deren Bedeutung für die Informationstheorie überhaupt irrelevant sei.

Meyers Großes Universallexikon (Mannheim, 1983) gibt als weitere Bedeutung auch an: „Bez. für Daten, bes. wenn diese eine log., in sich abgeschlossene Einheit bilden“.

Das sind verschiedene Sichten auf Information, die einander teilweise widersprechen: Eine umgangssprachlich als Information bezeichnete „Nachricht“, „Mitteilung“, oder „Auskunft“ ist ohne ihre Bedeutung („meaning“ bei Shannon) nicht sinnvoll.

In englischen Lexika ist „information“ als eigenständiges Wort häufig gar nicht zu finden, oder nur als Fachausdruck der US-Justiz („Information in the United States is a formal written accusation of crime prepared and presented to the court...“ – Encyclopædia Britannica, 1967). Meist wird jedoch „information theory“ im Sinne von Shannon aufgeführt.

Die verschiedenen angeführten Bedeutungen können zu einer Definition zusammengefügt werden, um genau das zu beschreiben, was „Information“ in dieser Arbeit bezeichnen soll:

Information: „Festgehaltene Daten, die wichtig sind, d. h. deren Verlust nicht wünschenswert ist.“

Das sagt noch nichts über die Art der Speicherung der Daten (etwa analog oder digital) aus, und setzt voraus, daß die Daten für mindestens einen Menschen eine Bedeutung haben (sonst wären sie ja nicht wichtig).

Der Informationsbegriff aus Shannons Informationstheorie ist in dieser Definition bewußt nicht enthalten, da es in dieser Arbeit mehr um den sozialen als um den technischen Aspekt der Information geht.

1.2.2 Informationsverlust

Die normale Bedeutung von „Verlust“ ist ziemlich eindeutig: Wenn etwas, was vorher existiert hat, nicht mehr existiert, sprechen wir von Verlust. Ein „de-facto-Verlust“ kann aber auch eintreten, wenn etwas zwar noch existiert, aber nicht mehr mit vernünftigem Aufwand zugänglich ist. Beim Verlust digitaler Information dürfte das sogar der häufigere Fall sein.

Informationsverlust tritt ein, wenn gespeicherte Informationen überhaupt nicht mehr lesbar und interpretierbar sind, oder wenn ihr Auslesen und Interpretieren teurer oder aufwendiger wäre als der angenommene Wert der Information oder der Aufwand für ihre Wiederbeschaffung aus anderer Quelle (falls möglich).

In der Technik wird auch von Informationsverlust gesprochen, wenn bei technischen Verfahren ein Teil der Information in Shannonschem Sinne unwiederbringlich verloren geht. Um diese Bedeutung von der nicht technischen Definition dieser Arbeit abzugrenzen, schlage ich die Verwendung des Wortes „Reduktion“ vor, die in der Technik ebenfalls für solche Vorgänge verwendet wird:

Informationsreduktion tritt ein, wenn ein technisches Verfahren eine Vorlage abbildet, und diese Abbildung nicht mehr mit dem Original identisch ist.

1.2.3 Digitalisierung

Die eigentliche Bedeutung der Digitalisierung[1] ist „Umwandlung der analogen Darstellung des Wertes einer physikalischen Größe in eine digitale Darstellung“ (Lexikon der Informatik und Datenverarbeitung, Oldenbourg München Wien 1997).

In häufigem Gebrauch sind jedoch andere, erweiterte Bedeutungen: Das Lexikon des Verlagswesens (Oldenbourg, München Wien 1997), versteht unter „Digitalisieren“: „Eine Vorlage in digitale Daten umwandeln, meist per Scanner, bisweilen auch manuell...“.

[1] Engl. *digitization*. Achtung, „*digitalization*“ bedeutet etwas komplett anderes, nämlich „Administration of digitalis to a patient with heart-diesease, in amounts sufficient to produce full therapeutic effect“ in der Medizin (Chambers Science and Technology Dictionary, Chambers Cambridge 1988).

Häufig ist diese Bedeutung gemeint, wenn im Zusammenhang mit Medieninhalten und Datenträgern (etwa Bücher, Filme etc.) von Digitalisierung gesprochen wird.

In der Umgangssprache wird Digitalisierung noch weiter gefaßt verwendet, wie z. B. in: „Bei der Digitalisierung der öffentlichen Verwaltung und der Sozialversicherungsträger sowie der Einbindung der Bürger besteht Handlungsbedarf.“ [Lind03, S. 1]. Diese Formulierung steht auf der Titelseite einer „Wochenzeitung für IT“ [Informationstechnologie], kann daher als der Öffentlichkeit verständlich angenommen werden. Hier meint Digitalisierung die Umstellung aller Geschäftsprozesse und sonstigen Arbeitsvorgänge auf elektronische Datenverarbeitung. Eine ähnliche Bedeutung möchte ich in dieser Arbeit verwenden.

Digitalisierung bedeutet in dieser Arbeit demnach die Tendenz, geschäftliche, wissenschaftliche und private Aktivitäten, die mit Daten zu tun haben, an Computern durchzuführen, und ihre Ergebnisse digital auf Datenträgern zu speichern.

1.3 Die Problematik im öffentlichen Bewußtsein

In Gesprächen stelle ich immer wieder fest, wie wenig die Problematik vielen Menschen bewußt ist. Selbst InformatikerInnen und andere TechnikerInnen merken häufig, daß sie zwar meine Argumente nachvollziehen können, aber das Problem vorher nicht bedacht haben.

Es scheint so zu sein, daß wir auf Grund unserer Erfahrungen mit Informationsträgern der physischen Welt (Bücher, Fotos usw.) die Erwartung haben, daß neue Technologien, die an die Stelle der alten treten, einfach in jedem Aspekt „besser“ sind und keine neuen Probleme aufwerfen.

BibliothekarInnen sind auch häufig verblüfft, wenn sie von den Problemen hören. Sie wissen zwar, daß Bücher nur eine begrenzte Lebensdauer haben (siehe Kap. 4.1.2 auf Seite 54), aber daß die Digitalisierung nur mit Einschränkungen eine langfristige Sicherung des Zugangs bedeuten kann, widerspricht den früheren, optimistischen Sichtweisen und auch der massiven Werbung der mit Digitalisierung befaßten Firmen, die ihre Dienstleistungen u. A. mit diesem Argument verkaufen wollen.

In den Massenmedien taucht die Problematik kaum auf. Wenn von verlorenen Daten zu lesen ist, dann meist im Kontext von Katastrophen und anderen unerwarteten Ereignissen, sehr selten als normaler Vorgang.

Ich habe eine Anzeige der Firma Philips in einer Fernsehzeitschrift gefunden, die eine angebliche Lösung für das Problem der Lebensdauer von Videoaufnahmen bewirbt.

Die Werbung zeigt eine Videokassette und eine DVD+RW-Scheibe nebeneinander. Der Slogan ist: „Vergängliche Aufnahmen ... halten jetzt ewig“.

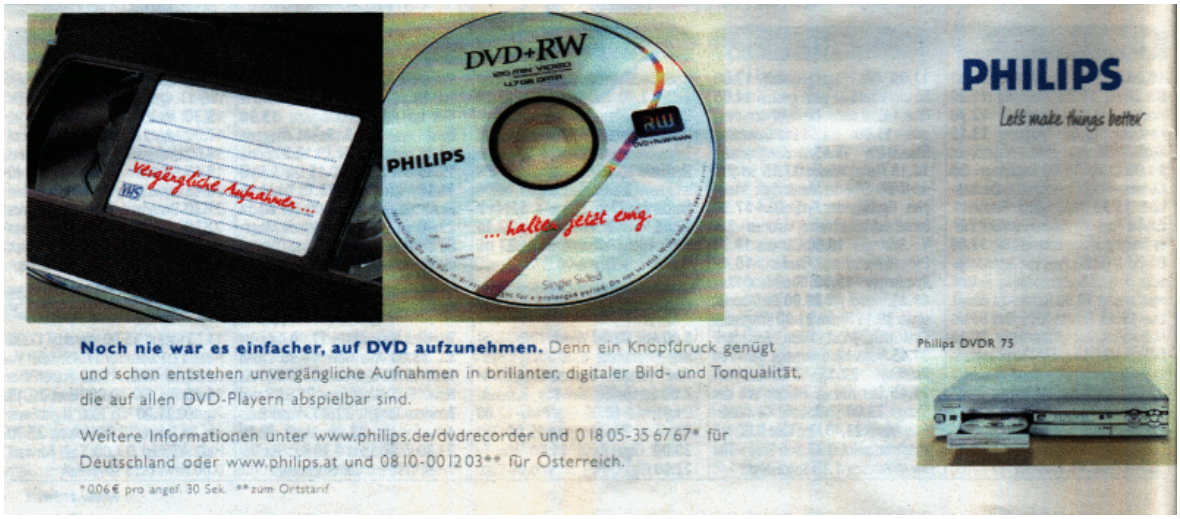


Abbildung 1: Werbung für Philips DVD-Recorder (aus „tele“ 41/2003, 9. 10. 2003)

Die Aussage „ewig“ gehört eindeutig ins Reich der Fantasie. Die Haltbarkeit von wiederbeschreibbaren DVD-Medien wurde noch nicht genügend erforscht, um eine Aussage auch nur über 5 Jahre zu treffen. Es ist jedoch bekannt, daß wiederbeschreibbare Medien (CD-RW, DVD+/-RW) kürzer haltbar sind und früher unlesbar werden als einmal beschreibbare (CD-R, DVD+/-R), siehe Kap. 4.1.8 auf Seite 60. Es kann also durchaus passieren, daß die Aufnahme auf der VHS-Videokassette (vielleicht in schlechterer Qualität, aber noch ansehbar) die auf der DVD+RW überlebt.

Ich denke nicht, daß hier bewußt versucht wird zu manipulieren. Wahrscheinlich denken die Leute, die die Anzeige gestaltet haben, tatsächlich, daß DVD+RWs (und digitale Daten allgemein) länger haltbar sind als VHS-Kassetten, und vor allem halten sie diese Aussage für so unumstritten, daß sie ohne besondere Argumentation in einer Anzeige stehen kann.

Selbst Computerfirmen verstehen unter „Langzeitverfügbarkeit“ nicht mehr als einige wenige Jahrzehnte. Das Produkt „Tivoli Storage Manager“ der Firma IBM, ein Dokumentenarchivierungssysstem, beherrschte bis zur Version 5.2 nur die Speicherung über 27 Jahre. In der Version 5.2 wurde diese Frist auf immerhin 82 Jahre ausgedehnt[2].

1.4 Überblick über die Literatur zum Thema

Als Standardwerke gelten Jeff Rothenbergs SCIENTIFIC AMERICAN-Artikel [Roth95a] aus 1995 und sein Bericht „Using Emulation to Preserve Digital Documents“ [Roth00] aus 2000. Diese zwei Texte werden in praktisch allen neueren Veröffentlichungen zitiert, in jenen von vor 2000 nur der Aufsatz aus SCIENTIFIC AMERICAN.

[2]IBM Tivoli Storage Manager - Product enhancements in V5.2.3 http://www.ibm.com/software/tivoli/products/storage-mgr/enhancements-v5.2.html

Der Großteil der Publikationen übernimmt Rothenbergs Argumentation in Bezug auf Migration und Emulation und stellt die Emulation näher vor oder geht auf Einzelaspekte ein. Werke, die sich kritisch mit Rothenbergs Aussagen beschäftigen, sind schwerer zu finden, und es werden auch keine neuen Wege der Sicherung der Langzeitverfügbarkeit vorgeschlagen.

Neben den allgemein-wissenschaftlichen Publikationen wie SCIENTIFIC AMERICAN oder BILD DER WISSENSCHAFT ist Literatur zur Langzeitverfügbarkeit vor allem in bibliotheks- und archivwissenschaftlichen Zeitschriften, seltener auch in Computerzeitschriften zu finden. Die AutorInnen kommen häufig erkennbar von der einen (BibliothekarIn, ArchivarIn) oder der anderen (InformatikerIn) Seite, was sich auf die Perspektive und auch auf die Kenntnisse über die Aspekte der „fremden“ Wissenschaft auswirkt; manchmal werden notwendige Fragen gar nicht gestellt.

1.5 Überblick: Langzeitverfügbarkeit

Unsere Gesellschaft produziert Information in ständig zunehmenden Mengen (vgl. z. B. [Zimm01, S. 51]), und ein immer größerer Anteil davon entsteht digital. Weiters werden laufend Informationen von herkömmlichen Datenträgern in digitale Systeme übernommen. (In der Fachsprache heißt dieser Vorgang „Retrodigitalisierung“.) All diese digitale Information soll für die Zukunft, möglichst langfristig, aufbewahrt und nutzbar gehalten werden.

Die Sicherung der digitalen Langzeitverfügbarkeit kann definiert werden als „the planning, resource allocation, and application of preservation methods and technologies necessary to ensure that digital information of continuing value remains accessible and usable“ ([Day01, S. 161]).

Auf das überlieferte Wissen früherer Zeiten zurückgreifen zu können ist eine Grundlage unserer Kultur und Wissenschaft. Wir halten es für selbstverständlich, daß wir lesen können, was vor 50, 150 und 500 Jahren geschrieben wurde, manchmal aus den Originalquellen, sonst aus rechtzeitig angefertigten Kopien oder Neuauflagen des Originals.

Der einfache Vorgang, ein mehr als hundert Jahre altes Buch aufzuschlagen und darin zu lesen, setzt in Wirklichkeit eine Menge Dinge voraus, die uns im Alltag nicht auffallen. Die folgende Auflistung betrifft einige Bücher, die ich selbst zu Hause stehen habe; sie mag triviale Dinge enthalten, aber all das ist bei der digitalen Informationsspeicherung nicht selbstverständlich:

- Zur Zeit der Entstehung des Buches wurde ein Kodiersystem (z. B. das lateinische Alphabet) verwendet, das wir auch heute noch interpretieren können. Der Autor

hat in einer heute verständlichen Sprache geschrieben.

- Die kodierten Informationen wurden in ein Objekt (das Buch) eingebunden, dessen Gestaltung auf heute noch bekannten Konventionen beruht (z. B. sind die Seiten numeriert und in der Reihenfolge des Textes gebunden). Deswegen können wir damit heute leicht umgehen.

- Als Datenträger wurde ein Material verwendet, das seine Form auch nach hundert Jahren noch nicht geändert hat. Die Informationselemente (Buchstaben) wurden so am Datenträger befestigt, daß sie nach dieser Zeitspanne noch an ihrem ursprünglichen Platz sind, und ihre Farbe ist von der des Hintergrundes unterscheidbar.

- Das Buch wurde auf seiner Außenseite mit der Bezeichnung des Inhalts („Titel“) versehen. Auf diese Weise kann es schnell zwischen vielen ähnlichen Büchern gefunden werden.

- Das Objekt war hinreichend stabil und transportabel, es hat zwei Weltkriege, gesellschaftliche Umstellungen, mehrere Umzüge über Hunderte Kilometer und weitere private Entscheidungen überdauert. Sein Wert war nicht gering genug, um es z. B. in schwierigen Zeiten, wenn viele Leute in einer Wohnung wohnen mußten, loszuwerden, aber auch nicht groß genug, um Diebstahl oder Raub in chaotischen (Kriegs-)Zeiten zu provozieren.

- Alle technischen, organisatorischen und mentalen Voraussetzungen, das Buch zu lesen, sind vorhanden.

Betrachten wir im Vergleich dazu das Beispiel einer zehn Jahre alten, auf einer Diskette gespeicherten hypothetischen Schularbeit.

- Wahrscheinlich wurde die Arbeit damals unter DOS oder Windows 3.1 mit einem der seinerzeit üblichen Textverarbeitungsprogramme (z. B. MS-Word 6, MS Word für Windows 6, WordPerfect 5 usw.) auf der Diskette abgelegt. Die Diskette wurde mit dem FAT (File Allocation Table)-Dateisystem formatiert, das DOS (und das damals noch darauf basierende Windows) als einziges beherrschten. Das impliziert auch, daß für den Namen der Datei nur maximal 8 Zeichen zur Verfügung standen. Es wurde das eigene Format des Textverarbeitungsprogramms verwendet, da keine systemübergreifenden, standardisierten Formate, die alle Informationen über die Gestaltung des Dokuments speichern konnten, zur Verfügung standen oder aus dem Programm heraus speicherbar waren.

- Nehmen wir an, daß eine 3,5-Zoll-Diskette verwendet wurde. Für solche Disketten, im Gegensatz zu den damals noch üblichen 5,25-Zoll-Disketten, sind heute noch Laufwerke erhältlich.

- Wenn wir damals viele Daten auf Disketten gespeichert haben, haben wir hoffentlich auch Vermerke über die Inhalte der einzelnen Disketten angelegt. Ohne diese ist es mühsam, die richtige Diskette wiederzufinden.

- Wir brauchen einen Computer mit Diskettenlaufwerk und einem auf diesem Computersystem lauffähigen Betriebssystem, dessen Bedienung wir kennen. Der Computer braucht natürlich Strom, wir können ihn daher nicht irgendwo betreiben, sondern nur in der Nähe einer Steckdose.

- Nehmen wir an, daß die Diskette noch physisch vom Laufwerk lesbar ist. Dies ist nach 10 Jahren schon eine optimistische Annahme (vgl. etwa [Henz99]).

- Das FAT-Dateisystem kann von heute üblichen Betriebssystemen noch gelesen werden. Wenn mehrere Dateien auf der Diskette sind, kann es jedoch nach 10 Jahren schwer sein, aus den maximal 8 Zeichen langen Dateinamen noch auf den Inhalt zu schießen.

- Wenn wir die Datei gefunden haben, müssen wir versuchen, sie zu lesen. Der simple Datenstrom ist für Menschen nicht verwendbar, der Text ist mit Kontrollzeichen und anderen internen Daten des Textverarbeitungsprogramms vermischt. Wir brauchen also ein Textverarbeitungsprogramm, das dieses Format noch interpretieren kann. Das frei verfügbare OpenOffice.org z. B. kann noch Dateien im Format von Microsoft Word 6 öffnen, hat aber für WordPerfect keinen Importfilter. Für WordPerfect müßte also ein anderes Programm gesucht werden, das eventuell Geld und sicher Zeit (für die Installation und das Erlernen der Bedienung) kostet.

Erst wenn all diese Bedingungen zutreffen, können wir unsere 10 Jahre alte Datei lesen.

Das illustriert schon die Probleme, die sich mit zehnmal jüngeren Daten im Vergleich zum Buch stellen. Um ein altes digitales Originaldokument zu lesen, sind eine Menge Voraussetzungen zu erfüllen:

- Der Datenträger muß noch gefunden werden können und zur Verfügung stehen.

- Der Datenträger muß noch in einem lesbaren Zustand sein, das heißt sowohl das Trägermaterial als auch die Schicht, die die Information speichert, müssen unbeschädigt sein.

- Wir brauchen die dazu passende Hardware in funktionsfähigem Zustand mit einer Schnittstelle zu einem funktionierenden Computer. (Ein Bandlaufwerk aus den 1970-er-Jahren, das zwar ein damaliges Magnetband lesen kann, aber keinen Anschluß für heute verfügbare Computer hat, ist wertlos, es sei denn man hat die Ressourcen und das Wissen, eine solche Schnittstelle zu bauen. Ohne Baupläne des Bandlaufwerks und eine Beschreibung seiner Schnittstelle ist das praktisch unmöglich oder extrem aufwendig, und selbst mit diesen Plänen dürfte nur ein größeres Hardware-Labor in der Lage sein, eine solche Schnittstelle zu bauen. Der Aufwand dürfte mindestens einige Monate betragen.)

- Die Bits können nun in den Computer übertragen werden. Wir müssen in Erfahrung bringen, welche Bedeutung die Daten haben, etwa ein Dateisystem, eine Archivdatei, die selbst mehrere Dateien enthält oder direkt die gesuchte Datei. Es ist gut, wenn Aufzeichnungen über das logische Format der Daten am Datenträger vorhanden sind; wenn nicht, wird es ziemlich aufwendig, da wir dann im Extremfall nicht einmal mehr davon ausgehen können, daß 8 Bits ein Byte ergeben (früher wurden auch nur 7 verwendet, um Speicherplatz zu sparen), oder daß Buchstaben nach einem uns bekannten System in Bytes kodiert wurden.

- Wenn es sich um ein Dateisystem oder eine Archivdatei handelt, müssen wir diese interpretieren können, um Beginn und Ende der gesuchten Datei zu finden. Bis in die Mitte der 1980-er-Jahre wurden in der EDV eine unüberschaubare Zahl von Betriebssystemen verwendet, deren Dateisysteme und Archivdateiformate fast alle unterschiedlich waren. Ohne Aufzeichnungen über diese Formate ist ihre Dekodierung sehr schwierig.

- Wenn die Datei extrahiert werden konnte, müssen wir wieder feststellen, in welchem Format sie vorliegt. Was ist der Inhalt der Datei? (Z. B. Texte, statistische Daten, Bilder, ein Programm etc.) Wie ist sie kodiert? (Z. B. ASCII oder Unicode[3]?) In welchem Format sind Zahlen gespeichert, als Text oder binär? Wenn binär, enthält das erste oder letzte Byte den kleinsten Anteil der Zahl[4]? Wie sind Datenfelder und Datensätze voneinander getrennt? usw.

[3] ASCII: Siehe Kap. 3.5.3 auf Seite 29
Unicode: Siehe Kap. 3.5.3 auf Seite 30

[4] Diese Unterscheidung wird *big-endian* oder *little-endian* genannt. Beide Methoden sind heute auf verbreiteten Computerplattformen üblich, eine Einigung ist nicht in Sicht, da die Abwärtskompatibilität gefährdet wäre, und das würde genau dem System schaden, das zuerst die Umstellung wagt.

Keine der beiden Methoden hat offensichtliche Vorteile gegenüber der anderen, es handelt sich um Konventionen. Die Namen stammen aus „Gullivers Reisen“ (Jonathan Swift: Gulliver's Travels, Wordsworth, Ware, 1992, Seite 34); dort ist die Ursache des Konflikts zwischen den Staaten Lilliput und Blefuscu, daß sie die Frühstückseier an unterschiedlichen Enden aufschlagen.

> Das Feststellen des Formats einer komplett unbekannten Datei kann extrem aufwendig oder unmöglich sein, je nachdem, wie sehr die Konventionen, die bei ihrer Entstehung zeitlich und örtlich gültig waren, von den heutigen abweichen (vgl. [Roth95a, S. 28]).

Ein wesentlicher Unterschied zwischen dem hundert Jahre alten Buch und der zehn Jahre alten Diskette ist also, daß sich die Zugriffsmethoden auf das Buch in hundert Jahren kaum geändert haben[5], während einzelne Aspekte des Zugriffs auf die Datei auf der Diskette in den zehn Jahren schon anders geworden sind, oder drohen, in nächster Zeit anders zu werden. Es ist gar nicht abzuschätzen, wie sich die Benutzung der Computer in den restlichen neunzig Jahren entwickelt, und wie viele von den notwendigen Rahmenbedingungen komplett geändert oder aufgegeben werden.

1.6 Beispiele verlorengegangener Information

Es gibt im privaten Bereich natürlich eine Menge Beispiele dafür, daß Daten verlorengehen: Disketten werden kaputt, jemand löscht zufällig eine noch benötigte Datei, manche Webmail-Dienste löschen grundsätzlich e-mails, die ein bestimmtes Alter erreicht haben, usw. Solche Beispiele sind leicht durch Umfragen im eigenen Bekanntenkreis zu finden.

Es gibt aber auch für die Öffentlichkeit relevantere Fälle. In der ersten Zeit der elektronischen Datenverarbeitung hatten vor allem große Firmen und öffentliche Einrichtungen Zugriff auf EDV-Systeme; gleichzeitig gab es durch die geringere Verbreitung und den damit verbundenen geringeren Standardisierungsdruck einen größeren Wildwuchs an Speichersystemen (Hard- und Software) und kürzere oder mit den heutigen vergleichbare Produktzyklen.

Der Bericht an den US-Kongress „Taking a byte out of history: The archival preservation of federal computer records“ [Cony90, S. 23] nennt mehrere problematische Fälle, in denen Behörden der Vereinigten Staaten wichtige Daten verloren haben: Es handelt sich um Volkszählungsdaten aus 1960, die in den 1970-er-Jahren nicht mehr lesbar waren, Listen von im Vietnamkrieg getöteten und vermißten US-Soldaten usw. Mehrere Veröffentlichungen (z. B. [Step98]) beschreiben, daß die US-Raumfahrtbehörde NASA verschiedene Daten diverser Missionen wegen der Unlesbarkeit der Magnetbänder und der verwendeten Formate verloren hat. Für die Interpretation der Daten, die technisch noch lesbar waren, mußten später teilweise die damaligen ProgrammiererInnen, bereits in Pension, zur Hilfe gerufen werden.

[5] Die Produktionsmethoden natürlich schon, aber die sind für den Zugriff auf den Inhalt des Buches irrelevant.

Eine Veröffentlichung der NASA selbst[6] beschreibt, daß ein Teil der Daten auf ca. 2.000 Magnetbändern der International Ultraviolet Explorer-Mission nicht in aktuell lesbaren Formaten vorhanden ist und schätzt die Kosten der Rettung dieser Daten auf ca. 35.000 US-Dollar. Da das angeblich teurer wäre als der Wert der Information, schlägt die NASA vor, die Bänder ohne Rettung der Inhalte aus dem Archiv auszusondern.

Das „Memory of the World"-Programm der UNESCO ([Abid98]) publiziert weitere Beispiele[7], auch solche aus den letzten zehn Jahren: z. B. wurde die Webseite des Weißen Hauses beim Amtsantritt von George W. Bush komplett geleert und neu begonnen; Teile des früheren Inhalts, die sonst nirgends gespeichert waren, gingen verloren. Die schwedische Zeitschrift Aftonbladet verlor das Archiv ihrer Online-Ausgabe von ca. zweieinhalb Jahren.

Überhaupt ist die „Langzeit"-Verfügbarkeit von Informationen im Internet nur als katastrophal zu bezeichnen. Brewster Kahle, der Betreiber des Internet-Archivs[8] gibt in [Kahl97] die durchschnittliche Lebensdauer von Internet-Adressen mit nur 44 Tagen an; eine andere Gruppe, die sich mit Links, die in wissenschaftlichen Journalen publiziert wurden, beschäftigt hat, kommt zum Ergebnis, daß nach 15 Monaten bereits 10 % der Adressen nicht mehr gültig waren (siehe [Del+03]). Das ist erschreckend, weil es sich wohl zu einem Großteil um relevante wissenschaftliche Publikationen handelt, deren Nachvollziehbarkeit eigentlich wichtig wäre.

Natürlich ist das Problem der verlorenen Information nicht auf digitale Daten beschränkt; auch Bücher und andere Dokumente auf Papier haben eine beschränkte Lebensdauer, selbst wenn keine Katastrophen wie der Brand der Bibliothek von Alexandria (vgl. [Canf98]) oder die Überschwemmung von Bibliotheken und Archiven im Sommer 2002 in Mitteleuropa eintreten (siehe auch Kap. 4.1.2 auf Seite 54).

[6] Disposition of Original IUE Tapes at the National Space Science Data Center http://nssdc.gsfc.nasa.gov/astro/iuepaper.html

[7] z. B. Digital Information Poses Problems For Conservationists http://portal.unesco.org/ci/ev.php?URL_ID=2235&URL_DO=DO_TOPIC&URL_SECTION=201&reload=1089541768

[8] Internet Archive http://www.archive.org/

2 Forschungsfragen und Hypothesen

2.1 Wie groß ist das Problem?

Hypothese: Alle Informationen, die ohne besondere Berücksichtigung der Langzeitverfügbarkeit digital geschaffen oder digitalisiert und digital gespeichert wurden, sind innerhalb von Jahren vom Verfall bedroht. Selbst die Beachtung der erarbeiteten Empfehlungen etwa von Jeff Rothenberg kann die Langzeitverfügbarkeit nicht in jedem Fall sichern, und es gibt Arten von Daten, auf die die Empfehlungen nicht anwendbar sind.

2.2 Was sind die Ursachen des Problems?

Hypothese: Der Großteil der Computer-Industrie ist wegen des mangelnden Interesses auf der Nachfrageseite nicht oder nur marginal daran interessiert, Langzeitverfügbarkeit in ihre Produkte einzubauen.

Hypothese: Die „inhaltsproduzierende Industrie“ ist nicht oder nur marginal daran interessiert, die Langzeitverfügbarkeit ihrer Produkte zu sichern.

2.3 Was sind aktuelle Trends?

Hypothese: Große Teile der Computerindustrie und der Unterhaltungsbranche arbeiten an Wegen, die die Sicherung der Langzeitverfügbarkeit digitaler Daten noch stärker als bisher behindern.

2.4 Sind die in der Literatur vorgeschlagenen Verfahren in der Praxis umsetzbar und lösen sie das Problem?

Hypothese: Weder Migration noch Emulation sind in der Lage, alle auftretenden Probleme zu lösen. Beide Verfahren haben ihre Stärken und Schwächen; für die Praxis kann eine Mischung aus beiden die beste Lösung sein. Es gibt Informationen, die mit keinem der genannten Verfahren langfristig zugänglich gehalten werden können; für diese müssen speziellere Verfahren gefunden werden oder die Information geht verloren.

2.4.1 Sind die vorgeschlagenen Verfahren im privaten Bereich anwendbar?

Hypothese: Im privaten Bereich sind die Mittel und Kenntnisse, die für die Anwendung der vorgeschlagenen Verfahren der Langzeitverfügbarkeit nötig wären, der-

zeit kaum vorhanden. Es besteht ein Bedarf an vereinfachten Verfahren und einfach nachvollziehbaren Anleitungen, um die Langzeitverfügbarkeit zu sichern.

2.4.2 Unterstützt das Rechtssystem die Langzeitverfügbarkeit digitaler Information?

Hypothese: Alle Industrieländer haben bereits Gesetzgebung, oder sie sind dabei, Gesetze zu verabschieden, die dazu führen, daß die wichtigsten Verfahren der Langzeitverfügbarkeit, nämlich Migration und Emulation, in manchen Fällen illegal werden. Das wird zu Informationsverlust führen, wenn das Problem nicht in spezieller, neuer Gesetzgebung anerkannt und gelöst wird.

3 Speicherung und Digitalisierung der Information

Der Mensch begann sehr früh, „Information“ festzuhalten, oder zu „speichern“: ein Beispiel sind die Höhlenmalereien, deren Alter auf 50.000 Jahre geschätzt wird (vgl. [Vö96, S. 6]). Diese bildliche Information, und auch die sich später entwickelnden Schrift- und Zahlensysteme hatten ein wesentliches Merkmal gemeinsam: Sie waren alle unmittelbar durch die menschlichen Sinne zu erfassen. Nur die „Kodierung“ wurde immer abstrakter: die ursprünglichen Bilder wurden immer mehr stilisiert und vereinfacht, bis diese Bildzeichen überhaupt durch Zeichen, die Laute darstellen, abgelöst wurden. Zahlen wurden ursprünglich mit Strichen dargestellt, daraus entwickelten sich später „Kodes“, um größere Zahlen (etwa 100) einfach und eindeutig zu kennzeichnen. (Es ist weder beim Schreiben noch beim Lesen praktisch, mehrere hundert einzelne Striche zu zählen.)

Je nach „Datenträger“ (etwa: Steintafel, Wachstafel, oder ein Faden) und „Schreibmethode“ (Ritzen, mit Griffel schreiben, oder bei den Mayas Knoten knüpfen) war mehr oder weniger technischer Aufwand für die *Erstellung* erforderlich. Und je nach Abstraktionsgrad des Alphabets oder des Zahlensystems war ein unterschiedlicher Kenntnisstand notwendig, um die Information wieder zu erkennen, aber kein zusätzliches technisches Gerät.

Ein viel höherer Grad der Abstraktion entstand, als mit dem Fortschritt der Technik echte Datenträger entstanden, die nur mit den dazupassenden technischen Geräten, kaum mehr einfach durch Hinsehen interpretierbar waren. Ein frühes Beispiel: Lochkarten zur Steuerung von Jacquard-Webstühlen vom Anfang des 19. Jahrhunderts (vgl. [Klin59, S. 164]). Die Daten auf der Lochkarte hatten eine Bedeutung, die nun auf unterschiedliche Arten kommuniziert werden konnte: z. B. verbal oder schriftlich, wenn ein Mensch den Webvorgang durchführen sollte, oder eben mechanisch für die Maschine. Natürlich konnte ein Mensch, der sowohl mit der Technik des mechanischen Webstuhls als auch mit dem Vorgang des Webens vertraut war, mit etwas Aufwand und durch Anschauen der Konstruktionspläne der Maschine die Information auf der Karte deuten. Aber dieser Aufwand war viel höher als etwa das Lesen der verbalen Beschreibung, und weitere Informationen über die Maschine waren erforderlich.

An diesem Beispiel ist ersichtlich, daß es nicht ganz einfach ist, zwischen „rein menschlich interpretierbaren“ und „menschlich nicht interpretierbaren“ Informationen zu unterscheiden. Generell kann aber gesagt werden, daß Maschinen, die die Informationsverarbeitung und -speicherung erleichtern, gleichzeitig eine Voraussetzung (und ihre Abwesenheit eine Barriere) darstellen, wenn es um den Zugriff auf die gespeicherte Information geht. Dies gilt gleichermaßen für analoge und digitale maschinelle Informationsspeicherung.

3.1 Analoge und digitale Speicherung

Analoge Datenspeicherung funktioniert nach dem Prinzip, das Signal „ähnlich“ („analog“) abzubilden. So speichert etwa ein Magnetband in der Audiokassette das Abbild der Luftschwingungen, die vom Mikrofon aufgenommen und in ein durchgehendes elektrisches Signal umgewandelt wurden. Beim Abspielen wird dieses Signal im Lautsprecher dazu verwendet, einen Körper in Schwingung zu versetzen und so mehr oder weniger die selben Luftschwingungen zu erzeugen wie sie das Mikrofon aufgenommen hat.

Die Daten werden in einem analogen System nicht interpretiert. Ein rein analoges System kann aufgrund solcher Daten auch nicht Entscheidungen fällen oder feststellen, ob die Information bestimmten Regeln entspricht (z. B. eine bestimmte Struktur hat). Aus diesem Grund können analoge Speichermethoden nicht garantieren, daß die Information verlustlos von einem Datenträger auf einen anderen kopiert werden kann. In der Praxis läßt sich etwa bei VHS-Videokassetten schon bei der dritten oder vierten Kopiergeneration[9] ein kaum tolerierbarer Qualitätsabfall feststellen. (Siehe Experiment 7.1 auf Seite I.) Selbst bei Mikrofilm, einem beliebten Datenträger für die Langzeitarchivierung von Büchern und anderen Papierdokumenten, ist ein ungefährer Qualitätsabfall von 10 % bei jedem Kopiervorgang zu erwarten (vgl. [Smit99b, S. 7]).

Bei jeder analogen Aufnahme der Umgebung tritt auch eine Informationsreduktion auf, die jedoch so gering sein kann, daß sie nicht mehr menschlich erfaßbar oder meßbar ist. Die analogen Datenträger wie Fotofilm oder Magnetbänder haben eine „natürliche“ Obergrenze für die speicherbare Informationsmenge: auf dem Fotofilm wären das etwa die lichtempfindlichen Körnchen, am Magnetband die magnetisierbaren Partikel. Im Normalfall macht das keine Probleme (analoge Tonaufnahmen und Fotos sind „gut genug“), aber die Abweichung vom Original ist eben nicht oder nur schwer meß- und quantifizierbar.

Digitale Datenspeicherung bedeutet, daß der abzubildende Aspekt des Originals in eine Zahlendarstellung umgewandelt (wenn er nicht ursprünglich schon aus Zahlen bestand) und so gespeichert oder weiterverarbeitet wird. Diese Zahlendarstellung kann als Eingabe für logische und interpretierende Prozesse dienen, und bei weiteren Kopierschritten kann die Korrektheit der Übertragung sichergestellt werden, indem die Zahlen der Kopie mit denen des Originals verglichen werden. Bei Übertragungsfehlern wird einfach so lange wiederholt kopiert, bis die Kopie identisch mit dem Original ist.

Die Eigenschaft digitaler Daten, daß logische Prozesse auf sie anwendbar sind, macht – neben der Möglichkeit, sie verlustfrei zu kopieren – den großen Unterschied zu ana-

[9] Kopiergeneration: Jede Kopie vom Original ist die erste Kopiergeneration. Wenn von einer solchen Kopie weiterkopiert wird, entsteht die zweite Kopiergeneration usw. Es ist generell besser, analoge Daten von einer möglichst „geringen“ Kopiergeneration (im Idealfall vom Original) zu kopieren, aber das kann an der Abnutzung oder Unzugänglichkeit des Originals scheitern.

logen Daten aus:

> Der nicht technische, sondern organisatorisch entscheidende Schritt besteht in der Digitalisierung der übertragenen Nachricht.
>
> Dies bedeutet – auch im Hinblick auf das Zusammenwirken von Rechner- und Kommunikationstechnik –, daß
>
> - jedes übermittelte Signal durch den Betreiber des Kommunikationsinstruments (Netz, Vermittlung etc.) einer schnelleren Informationsverarbeitung zugänglich ist (Speichern, Auswerten, Manipulieren, Filtern, Codieren, Decodieren, Umrechnen etc.), und zwar all dessen, was durch rechentechnische (sprich algorithmische) Verfahren möglich ist,
> - das, was rechentechnisch möglich ist, aus prinzipieller Sicht nur durch das begrenzt wird, was noch formal möglich ist, und dies wird festgelegt durch das, was sich im Rahmen logischer Kalküle definieren läßt.
>
> [Korn93, S. 59]

Mit anderen Worten: Wenn eine Umformung denkbar und mathematisch/logisch auszudrücken ist, können wir sie auf digitale Daten anwenden.

Wie bei analogen tritt auch bei digitalen Aufnahmen der Umgebung eine Informationsreduktion auf. Es ist nämlich meist gar nicht möglich, etwas „wirklich digital“ aufzunehmen: Hörbare Töne z. B. bestehen nun einmal aus Schallwellen, nicht aus Zahlen. Dazu kommt, daß digitale Aufnahmegeräte genau definierte Parameter für die speicherbare Informationsmenge haben: eine digitale Videokamera etwa nimmt den Ton mit 48.000 Hz auf zwei Kanälen mit 16 Bit Genauigkeit auf, und das Bild mit 720 * 576 Pixeln, mit jeweils 3 Bytes für jeden Pixel (Bildpunkt). Hier ist die Reduktion also auch vorhanden (die Welt ist deutlich komplexer als sie mit 720 * 576 Bildpunkten abgebildet werden kann), aber wenigstens meßbar und konstant. Die *Weiterverarbeitung* wird jedoch im Vergleich zu analogen Daten stark erleichtert, weil eben beliebig oft weiterkopiert werden kann, und es eröffnen sich komplett neue Möglichkeiten der Bearbeitung.

Da die Daten als Zahlen vorliegen und definierte Strukturen haben, können nahezu beliebige Transformationen auf sie angewendet werden. Das geht zwar auch mit analogen Daten bis zu einem gewissen Grad, doch ist dazu häufig eine Trennung der unterschiedlichen Komponenten (etwa bei der Nachvertonung von Videos) und fast immer ein Umkopieren mit den bekannten Qualitätseinbußen notwendig. Digitale Transformationen hingegen sind meistens verlustlos, häufig auch umkehrbar oder dynamisch

anwendbar, und wirken nur auf die zu bearbeitenden Daten ein, ohne die anderen Daten anzutasten (Beispiel: digitale Nachvertonung von Videos).

Ein weiterer wichtiger Aspekt der digitalen Datenverarbeitung und -speicherung ist die Möglichkeit, Fehler im Datenstrom zu erkennen und unter Umständen zu korrigieren. Die Erkennung geschieht (stark vereinfacht ausgedrückt; die echten Verfahren sind um einiges komplexer, vgl. etwa [Vö96, S. 96ff]), indem z. B. die Anzahl der gesetzten Bits in einem vorangegangen Datenabschnitt übertragen wird; stimmt diese Summe nicht mit der empfangenen überein, bittet der Empfänger um neuerliche Übertragung oder versucht, die Daten anhand von zusätzlich vorhandenen Korrekturinformationen mit Hilfe mathematischer Verfahren zu korrigieren.

Auf den ersten Blick sieht es daher so aus, daß die Digitalisierung von Daten die Gefahr von Informationsverlust komplett bannen kann: schließlich können die Daten ja jederzeit verlustfrei umkopiert werden, bevor der Datenträger unbrauchbar wird. (Dies ist bei analogen Datenträgern ein großes Problem der Medienarchive: durch jedes analoge Umkopieren verschlechtert sich die Qualität der Aufnahmen – aber wenn nicht umkopiert wird, erreicht der Datenträger irgendwann das Ende seiner Lebensdauer und dann geht die Aufnahme überhaupt verloren.)

Wenn digitale Verarbeitung und Speicherung so viel besser sind als analoge, warum wurde dann nicht von Anfang an die digitale Technik entwickelt?

Voraussetzung für digitale Technik ist eine ziemlich weit entwickelte Elektronik. Andere Datenaufzeichnungstechnologien wurden aber bereits viel früher entwickelt, etwa fotochemische (Fotographie, Film) und mechanische (z. B. Thomas Alva Edisons Phonograph).

Eigentlich waren mit der Entwicklung des Telegraphen von Samuel Morse (1840) alle Voraussetzungen für digitale Speicherung, Übertragung und Vervielfältigung von Text in kodierter Form vorhanden. Die Jacquardschen Lochkarten oder Lochstreifen hätten als Eingabe eines geringfügig modifizierten Telegraphen dienen können, und dessen Ausgabe am anderen Ende der Leitung hätte auch auf Lochstreifen passieren können (statt wie bei Morse mit einem Stift die Punkte und Striche auf Papier zu zeichnen). Dadurch wäre es etwa möglich gewesen, die Übertragungsgeschwindigkeit zu steigern und die Leitungen besser auszulasten, indem die Kodierung durch Menschen auf Lochstreifen erfolgt, wenn nötig, auch parallel auf mehreren Maschinen. Die Übertragungsgeschwindigkeit wäre nicht mehr durch den Menschen (und seine „Klopfgeschwindigkeit“) limitiert gewesen. Ähnliche Lösungen sind aber – soweit es mir bekannt ist – erst viel später entwickelt worden; wahrscheinlich hatte die Menschheit um 1840 noch keine Verwendung für eine solche Technologie. (Bereits zehn Jahre später wurde ein digitaler „Bildtelegraph“ vorgestellt, dem kein kommerzieller Erfolg beschieden war. Erst

um 1910 war die Technik soweit, daß etwa aktuelle Pressefotos elektronisch übertragen werden konnten. Vgl. [Lü02, S. 119])

Die Ansprüche an die Elektronik für die Bearbeitung von digitalisierten Daten können sehr hoch sein. Weiter oben habe ich die Anforderungen für digitales Video beschrieben. Weit verbreitete Computersysteme haben erst in den 1980-er-Jahren die Fähigkeit bekommen, digitale Bilder zu bearbeiten (Speicherbedarf: einige hundert Kilobytes); in den 90ern, mit Ton umzugehen[10] (Speicherbedarf: einige Megabytes für einige Minuten Ton; der Rechner muß bestimmte Dinge in einer definierten Zeit erledigen können); und schließlich hat sich die digitale Bearbeitung von Videos an privaten Computern erst um 2000 herum durchsetzen können, weil dafür sehr große Kapazitäten (bis zu drei Megabytes an Daten pro Sekunde) notwendig sind.

3.2 Gründe und Impulse für die Digitalisierung

Eine neue Technologie muß sehr große Vorteile bieten, um ihre Vorgänger komplett zu verdrängen. Viele Aufgaben, für die wir heute praktisch ausschließlich Computer einsetzen, konnten aber vorher zufriedenstellend mit anderen Geräten, oder überhaupt ohne Technologieeinsatz gelöst werden.

Der Grund für die fast vollständige Umstellung ist wahrscheinlich in der Konvergenz zu suchen, die der Computer bietet. Er kann die Schreibmaschine ablösen (Textverarbeitung), genauso das Rechnen auf Papier oder mit Taschenrechner (Tabellenkalkulation), persönliche Besuche oder Telefonanrufe vermeiden helfen (e-mail, instant messaging[11]), die Stereoanlage ersetzen (MP3- und Ogg Vorbis-Dateien[12]), Fernsehsendungen statt des Videorecorders aufzeichnen, Videos abspielen und Vieles mehr. Wichtig ist auch die Möglichkeit, diese Daten aus verschiedenen Quellen zusammenführen und gemeinsam verarbeiten zu können, sodaß das Ganze mehr als die Summe seiner Teile wird (vgl. [Korn93, S. 3] und [Lü02, S. 121]).

Zuerst wurden Computer im II. Weltkrieg für solche Aufgaben verwendet, die Menschen nur langsam und mit häufigen Fehlern durchführen konnten: komplexe und/oder sich ständig wiederholende mathematische Berechnungen wie das Brechen der Ver-

[10] Erste Prototypen für digitale Sprachübertragung setzten die USA und Großbritannien bereits im 2. Weltkrieg ein. Der hohe Bandbreitenbedarf der digitalen Technologie war zwar ein Nachteil gegenüber der analogen Übertragung, aber die digitalen Signale besaßen einen für den Krieg entscheidenden Vorteil: Sie ließen sich abhörsicher verschlüsseln. Vgl. [Lü02, S. 121]

[11] Software, die eine Liste von Kontakten verwaltet und ermöglicht, diesen Personen kurze Mitteilungen, Internet-Adressen usw. zu schicken. Zu den bekanntesten Vertretern zählen ICQ, AOL Instant Messenger, Yahoo Chat und MSN Messenger.

[12] MP3: Abkürzung für MPEG (Motion Picture Expert Group) Layer 3, eine Technologie für die verlustbehaftete Kodierung von Ton.

Ogg Vorbis: wegen patentrechtlicher und technischer Unzulänglichkeiten von MP3 in internationaler Zusammenarbeit entstandene, frei verwendbare Audiokodierungstechnologie.

schlüsselung der deutschen Wehrmacht im zweiten Weltkrieg (vgl. [Smit00]) oder die Erstellung von Projektil-Flugbahn-Tabellen für die Artillerie. Zu dieser Zeit hatten auch nur militärische Einrichtungen Zugang zu Computern.

Über Jahrzehnte hindurch war Rechnen der wichtigste Anwendungsbereich der Computer („Rechner“). Erst als ein Entwicklungsstand erreicht wurde, auf dem die Computer freie Kapazitäten hatten und daher für mehr Aufgaben zur Verfügung standen, wurden andere Einsatzgebiete gesucht und gefunden, etwa die Textverarbeitung und Tabellenkalkulation (die eine komplett andere, interaktivere Form des Rechnens darstellt; die mathematischen Berechnungen stehen nicht im Vordergrund), oder sogar Spiele.

Die einfache Übertragung der Daten in identischer Form und der gemeinsame Zugriff darauf wurden mit der Verbreitung lokaler Computernetzwerke für Firmen und Behörden interessant. Vorher war etwa die Textverarbeitung „nur“ eine Erleichterung gegenüber der Benutzung der Schreibmaschine – mit den Netzwerken konnten erstmals auch gemeinsam Datenbestände geschaffen und gepflegt werden. Die betriebliche Kommunikation wurde verändert, manche Geschäftsprozesse komplett umgestellt. Diese Entwicklung hat dazu geführt, daß sehr große Datenmengen in komplexen elektronischen Systemen gespeichert sind, die nur mehr für SpezialistInnen zu überblicken sind.

Mit der vom Computer ermöglichten Konvergenz der Unterhaltung (Spiele), Kommunikation und Informationsbeschaffung (Internet) und Pflege privater Beziehungen (etwa die Bearbeitung von Familienfotos) drangen die Rechner auch in die Privathaushalte ein. Diese Systeme zeichnen sich durch geringere Komplexität als Behördensysteme, aber eine große Vielfalt und wegen mangelnder Fachkenntnisse nicht immer optimale Wartung aus.

Die Digitalisierung kann den Zugriff auf Information stark verbessern (vgl. [Smit99b, S. 7]). Thematisch zusammengehörige Informationseinheiten, die physisch in der ganzen Welt verstreut sind, lassen sich am Bildschirm nebeneinander betrachten und direkt miteinander vergleichen. Häufig nachgefragte Objekte sind gleichzeitig an mehreren Orten darstellbar, ohne daß dem Original Schaden durch intensive Benutzung droht. Durch die Möglichkeit verschiedener Ansichten auf die Informationen und (häufig) Volltextsuche sind die Inhalte auch meist besser aufzufinden.

3.3 Datenträger

3.3.1 Magnetische Datenträger

In einigen Metallverbindungen ändert der Kontakt mit einem (elektro)magnetischen Feld den magnetischen Zustand der enthaltenen Partikel. Diese Änderung ist ziemlich permanent und stabil, solange kein anderes Feld aufs Material einwirkt. Die Magnetfelder lösen wiederum in anderen Metallen elektronische Zustandsänderungen aus. Auf diese Weise wird die Information wieder gelesen.

Magnetische Datenträger eignen sich sowohl für analoge als auch für digitale Datenspeicherung. Für digitale Verfahren ist eine höhere Präzision und damit eine weiter entwickelte Technologie erforderlich. (Das Gerät muß etwa immer ganz genau wissen, an welcher Position des Datenträgers es sich befindet; das ist bei analogen Geräten selten notwendig.)

Magnetische Datenträger können auf zwei Arten angeordnet werden: entweder als Platte oder als Band.

Platten haben den Vorteil, daß ihre Fläche der Lese-Schreib-Komponente praktisch gleichmäßig zugänglich ist (*random access*, wahlfreier Zugriff). Bänder hingegen können aufgewickelt werden und dadurch eine viel größere Fläche und entsprechend höhere Kapazität bieten – aber sie müssen bis zu der zu lesenden oder beschreibenden Stelle vor- und rückgespult werden (*sequential access*, sequenzieller Zugriff).

Platten kommen einzeln vor (z. B. in Disketten) oder sie werden gestapelt (z. B. in Festplatten). Bänder sind heute zur leichteren Handhabung und wegen des besseren Schutzes vor Umwelteinflüssen eher in Kassetten eingeschlossen, seltener auf eine Rolle aufgewickelt; früher waren Rollen vorherrschend.

3.3.2 Magneto-optische Datenträger

Bei dieser Datenträgerart wird ein Laser zum Lesen verwendet. Das Licht wird abhängig von der Polarität der Magnetisierung von Partikeln im Datenträger reflektiert oder abgelenkt; der Lesekopf kann die Daten aus dem zurückkommenden Laserlicht ableiten.

Beim Schreiben wird die Datenträger-Schicht von einem Laser erhitzt. Es werden solche Materialien verwendet, die nur bei hohen Temperaturen (sog. *Curie-Temperatur*) magnetisiert werden können und die geänderte Polarität nach dem Abkühlen permanent aufbewahren. Diese Datenträger sind daher durch Magnetfelder, wie sie in normalen Umgebungen vorkommen, nicht gefährdet. (Es sei denn sie werden gleichzeitig auf mehrere hundert °C erhitzt.) (Vgl. [Schn97, S. 57])

Bei MO-Datenträgern hat sich kein Standard durchsetzen können. Es gibt und gab verschiedene am Markt konkurrierende, unkompatible Systeme (Datenträger + Le-

segeräte) mehrerer Hersteller. Da sich die standardisierten CD- und DVD-basierten einmal oder wiederbeschreibbaren Medien stark durchsetzen, ist der Marktanteil der MO-Speichersysteme heute gering, sie konnten sich nur in einzelnen Bereichen etablieren.

3.3.3 Optische Datenträger

In diese Kategorie gehören die Compact Disc und die DVD (Digital Versatile Disc) sowie ihre geplanten Nachfolger wie DVD-Audio oder Blueray.

„Optisch“ werden diese Datenträger genannt, weil das Lesen mit Hilfe eines Laserstrahls stattfindet. Der Strahl wird auf die Oberfläche des Datenträgers projiziert und dort entweder reflektiert oder abgelenkt. Die reflektierten Strahlen oder ihr Fehlen werden als Daten interpretiert.

Industriell massengefertigte optische Datenträger werden gepreßt, die Lichtbrechung entsteht durch den Wechsel von Erhöhungen und Vertiefungen in der Datenträgerschicht. Selbst beschreibbare Datenträger hingegen enthalten meist organische Farbstoffe oder spezielle Metalllegierungen, die mit einem im Vergleich zum Lesen viel stärkeren Laserstrahl permanent (CD-R) oder immer wieder änderbar (CD-RW) dazu gebracht werden, ihre Reflexionseigenschaften zu ändern und damit den gleichen Effekt zu erzielen (vgl. [Vö96, S. 285]).

Optische Datenträger dominieren heute die Verbreitung von Unterhaltungsinhalten. Aus diesem Grund müssen sie nicht nur technischen Anforderungen entsprechen, sondern sie sind auch nach wirtschaftlichen Überlegungen gestaltet. Das bedeutet unter anderem, daß die Daten auf allen neueren Datenträgertypen (DVD, DVD Audio) häufig verschlüsselt abgelegt sind, und die Geräte, die sie auslesen, müssen sich an Regeln halten, die von den Lizenzgebern des Datenträgerformats vorgeschrieben werden. Diese Regeln sind deutlich strenger als die Vorschriften des Urheberrechts. Aus diesem Grund gibt es heute (genauer seit der Implementierung der EU-Urheberrechts-Richtlinie im Jahr 2003 in Österreich) keine legale Möglichkeit, Film-DVDs zu kopieren (siehe auch Kap. 5.10.1 auf Seite 107). Bei Audio-CDs, die ursprünglich ohne Maßnahmen zur Verhinderung von Kopien spezifiziert wurden, verstoßen einige Hersteller mittlerweile bewußt gegen die Spezifikation, um ein digitales Auslesen auf Computern zu verhindern (vgl. [Volp03]). Solche CDs dürfen das „CompactDisc Digital Audio“-Logo nicht tragen und ihre Abspielbarkeit in normalen CD-Abspielern ist auch manchmal eingeschränkt (vgl. [Hans03]; betroffen sind vor allem CD-Player in Autoradios sowie tragbare Abspieler[13]).

[13] Das c't-CD-Register erfaßt für den deutschsprachigen Musikmarkt die Abspielbarkeit von nicht standardkonformen Audio-CDs in verschiedenen Abspielgeräten.

3.3.4 Flash-Datenträger

Während beim üblichen RAM (Random Access Memory, Direktzugriffsspeicher) ständige Stromzufuhr erforderlich ist, weil die Schaltungen im Speicher ihre Ladung verlieren, gibt es Materialien, die ihre Ladung permanent speichern können (vgl. [Vö96, S. 52]). Sie sind teurer und in der Herstellung komplexer als normales RAM und langsamer beschreibbar und auslesbar. Die Zahl der Schreiboperationen ist auf ca. eine Million begrenzt, danach ist der Speicher nicht mehr zu beschreiben. Da Flash-Speicher aber eine permanente Datenspeicherung auf kleiner Fläche ohne bewegliche Teile bieten, werden sie mit fallenden Preisen und steigenden Kapazitäten immer beliebter; das Hauptanwendungsfeld ist heute der Bereich der digitalen Foto-Kameras.

Flash-Speicher spielen wegen ihrer derzeit eher begrenzten Kapazität (Speicherkarten bis ca. 1 GB sind erhältlich) und des Preis-Leistungs-Verhältnisses (dieselbe Datenmenge wie auf z. B. einer CD-RW zu speichern ist ca. 60mal teurer[14]) in der längerfristigen Archivierung von Information noch keine größere Rolle. Das könnte sich jedoch ändern; eine 512-MB-Speicherkarte ist bereits heute für Privatpersonen bezahlbar und kann durchaus die Fotoproduktion eines Jahres einer Familie aufnehmen. Es ist also durchaus vorstellbar, daß im privaten Bereich bald Flash-Speicherkarten wegen ihrer einfachen und problemlosen Handhabung zur dominierenden transportierbaren Speichertechnologie werden.

3.4 Dateisysteme

Ein Datenträger erscheint gegenüber dem Computersystem im Grundzustand als eine leere Fläche mit einer gewissen Kapazität. Wir Menschen können damit nicht besonders viel anfangen, da wir auf Dinge wie Dateinamen, -eigenschaften, -größen, Unterverzeichnisse usw. Wert legen, die es ohne ein Dateisystem am Datenträger nicht geben kann. Deswegen unterstützt jedes Betriebssystem auch ein oder mehrere Dateisysteme, von denen heute über 30 in Verwendung sind[15]. Am Höhepunkt der Vielfalt der Computerplattformen, in den 1980-er-Jahren, hatten viele Plattformen ihre eigenen Betriebs- und Dateisysteme; manche von denen sind heute vergessen.

Größere Datenträger (z. B. Festplatten) können auch in Bereiche, sogenannte Partiti-

c't-CD-Register http://www.heise.de/ct/cd-register/

[14] Quelle: Preisangaben der Firma ditech Computer vom 19. August 2004.
CD-RW Rohling SENTINEL 700 MB: 1,50 €
Compact Flash Memory Card, 512 MB: 67,90 €

[15] Das Betriebssystem Linux unterstützt in der Version 2.6.6 nicht weniger als 26 verschiedene datenträger-basierte Dateisysteme zumindest so weit, daß es Daten von ihnen lesen kann. Microsoft Windows unterstützt nur „viereinhalb“: Das alte MS-DOS-Dateisystem FAT mit der Variante VFAT (die lange Dateinamen erlaubt), NTFS und die auf CDs und DVDs üblichen Dateisysteme ISO-9660 und UDF. Andere Betriebssysteme liegen irgendwo zwischen diesen beiden Werten.

onen, unterteilt werden, z. B. um das Betriebssystem von den Dokumenten der BenutzerInnen zu trennen. Das Format der Partitionstabelle, die über Ort und Ausdehnung der einzelnen Partitionen Auskunft gibt, ist meist pro Computerplattform festgelegt, aber da heute häufiger Datenträger auf verschiedenen Plattformen verwendet werden, ist es hilfreich, wenn ein Betriebssystem mehrere Partitionstabellenformate kennt[16].

Kleinere Datenträger (etwa Disketten) enthalten meistens keine Partitionstabelle.

Datenträger, die keinen direkten Zugriff unterstützen (z. B. Magnetbänder – sie können nur sequenziell gelesen werden) enthalten häufig nur Archivdateien, die eine ähnliche Funktionalität für den Zugriff auf Dateien wie Dateisysteme bieten.

Wenn wir einen Datenträger in einen Computer einlegen, muß dieser folgende Schritte durchführen (vereinfacht):

1. Die Kapazität und den Schreibschutzstatus des Datenträgers feststellen.

2. Feststellen, ob eine Partitionstabelle vorhanden ist oder ob der Datenträger nur ein Dateisystem besitzt.

3. Wenn vorhanden, die Partitionstabelle lesen und die Dateisysteme auf den verschiedenen Partitionen (oder das einzige vorhandene) identifizieren.

4. Wenn die Dateisysteme bekannt sind und vom Betriebssystem unterstützt werden, müssen sie „eingebunden“ (*mount*) werden. Das passiert unter Windows, indem ein Laufwerksbuchstabe vergeben wird; Unix und darauf basierende Systeme wie Linux, MacOS X usw. binden neue Datenträger in die Dateisystem-Hierarchie ein.

Erst wenn alle diese Schritte durchgeführt wurden, sind die Dateien auf dem Datenträger auf die übliche Weise (also über ihre Dateinamen) zugänglich.

Es gibt Möglichkeiten, von Datenträgern, deren Partitionsformat und/oder Dateisystem das Betriebssystem nicht unterstützt oder wenn die entsprechenden Tabellen beschädigt wurden, noch intakte Dateien zu retten, dies ist jedoch arbeitsintensiv und nicht in allen Fällen erfolgreich. Hierzu muß festgestellt werden, wo eventuelle Partitionen beginnen und enden und wo im Dateisystem die Dateinamen und die zu ihnen gehörenden Inhalte gespeichert sind.

[16] Linux kennt 20 verschiedene Formate. Andere Betriebssysteme, die es meist nur für eine Computerarchitektur gibt, unterstützen üblicherweise weit weniger.

3.5 Dateiformate

3.5.1 Begriffsbestimmungen

Auf den nächsten Seiten muß ich einige Fachbegriffe verwenden, die ein gewisses Wissen voraussetzen. Ich werde versuchen, sie zuerst allgemein verständlich zu erklären.

Bit: Eine binäre Zahl („binary digit“), die die Werte 0 und 1 annehmen kann. Die kleinste logische Informationseinheit.

Byte: Eine gewisse Anzahl von Bits, heute meist 8. Bits werden zu Bytes zusammengefaßt, weil sie allein für die meisten Aufgaben ungeeignet sind. Bytes hingegen können ganze Buchstaben und Zahlen ausdrücken.

Kodierung: Konventionen oder Vorschriften, um menschliche Informationselemente (z. B. den Buchstaben „A“, die Ziffer 7 oder eine Zeilenschaltung) am Computer auszudrücken. Eine Kodierung ist fast immer willkürlich, da die Informationselemente selten eine „natürliche Ordnung“ besitzen.
Eine der wichtigsten Kodierungen ist ASCII[17]; sie gibt vor, welcher numerische Wert in einem (7 oder 8 Bits langen) Byte welchem Buchstaben, Steuerzeichen oder Ziffer entspricht. Der Buchstabe „A“ hat zum Beispiel den ASCII-Wert 65.

Quellcode: Programmcode mit Anweisungen für Computer in einer für Menschen verständlichen Programmiersprache. Der Quellcode wird meistens mit Hilfe eines *Compilers* (Übersetzungsprogramm) in Maschinencode (Binärcode) übersetzt, der für Menschen im Allgemeinen nicht oder nur extrem mühsam lesbar ist. Quellcode kann mit Hilfe eines geeigneten anderen Compilers auch in Binärcode für andere Computersysteme übersetzt werden; das gilt nicht für den Maschinencode. Es ist daher nützlich, sowohl Quellcode als auch den Maschinencode eines Programms zu besitzen. Kommerzielle Softwarehersteller erlauben den Zugriff auf den Quellcode allerdings nur in Ausnahmefällen.

Open Source: Eine Bewegung in der Software-Entwicklung und eine Vertriebsform für Software. Open-Source-Programme werden mit Quellcode, im Allgemeinen gratis und ohne Beschränkung des Kopierens und der Weitergabe vertrieben. (Im Gegensatz dazu darf sog. *proprietäre* oder *kommerzielle* Software nicht kopiert werden, und sie ist auch selten gratis.) Open Source kann für die Langzeitverfügbarkeit digitaler Daten eine sehr wichtige Rolle spielen, da durch die Verfügbarkeit des Quellcodes das ganze System transparenter ist und leichter auf neue Computersysteme portiert werden kann (und typischerweise auch wird; die

[17] American National Standard Code for Information Interchange

meist-portierten Betriebssysteme NetBSD[18] und Linux sind beide Open Source; NetBSD läuft derzeit auf 17 verschiedenen Prozessortypen, Linux auf 16[19]). Es ist eine große Unabhängigkeit von Herstellern (deren Überleben, Entscheidungen über die Einstellung von Produkten sowie Preisvorstellungen in Abhängigkeitssituationen) gegeben. Nur in der Open-Source-Welt ist ein und dasselbe Programm mit minimalem Aufwand von kleinen Taschencomputern bis zu den größten Supercomputern portierbar.

3.5.2 Dateiformate - Überblick

Die Menschheit hat im Laufe ihrer Geschichte eine große Anzahl von Methoden erfunden, Wissen zu strukturieren. Hierzu gehören z. B. Alphabete nach den unterschiedlichen Überlegungen (Bild/Wort/Laut-Alphabet), Zahlensysteme (deren Basis ja nicht einmal in jeder Kultur 10 ist) oder auch die Schreibrichtung (von links nach rechts oder umgekehrt).

Es überrascht daher nicht, daß diese Vielfalt auf die Speicherung von Computerdaten übertragen wurde. Es gibt zwar strukturelle Beschränkungen, zum Beispiel ist die Anzahl der Bits in einem Byte meistens pro Computersystem festgelegt, aber diese lassen noch einen extrem großen Freiraum für die Gestaltung der Dateiformate. Es gibt zwar „Moden“, das sind zum jeweiligen Zeitpunkt anerkannte Methoden, doch werden in der Praxis nach wie vor viele Methoden nebeneinander angewendet. (Derzeit scheinen XML-basierte Formate die Mode zu sein.)

Dateiformate lassen sich nach unterschiedlichen Kriterien gruppieren: etwa nach Strukturierung oder Art der Verwendung (eine ähnliche Kategorisierung findet sich in [Clau04, S. 4]). Ich werde hier einige wichtige Gruppen von Dateiformaten beschreiben, in Kap. 4.4 auf Seite 69 folgt die Beschreibung ihrer für die Langzeitverfügbarkeit relevanten Aspekte.

[18] NetBSD-Projekt http://www.netbsd.org/

[19] Quellen:
NetBSD: Hardware supported by NetBSD http://www.netbsd.org/Ports/
Linux: Zählen der Architekturen im Quellcode des Linux-Kernels, Version 2.6.6
Verbreitete kommerzielle Betriebssysteme unterstützen oft nur eine Plattform (z.B. Microsoft Windows und Apple MacOS X) oder einige wenige (Sun Solaris). Windows NT wurde ursprünglich auch für die Plattformen MIPS und Alpha umgesetzt, diese werden jedoch nicht mehr unterstützt, was erhebliche Investitionen der Microsoft-Kunden, die auf diese Plattformen gesetzt haben, vernichtet hat.

3.5.3 Unstrukturierte (Freiform-) Textdateien

Diese Dateien werden meist von Menschen mit Hilfe eines Editorprogramms[20] erstellt, oder sie stammen aus der Ausgabe eines Programms. Sie müssen, da sie nicht als Eingabe eines anderen Computerprogramms gedacht sind, keinerlei Vorgaben genügen. Sie werden nur von Menschen interpretiert.

Auf dem Speichermedium liegen sie fast genauso, wie sie am Bildschirm erscheinen. (Es gibt natürlich einige Kontrollzeichen, die z. B. einen Zeilenwechsel o. Ä. bewirken.)

Solche Dateien werden häufig für Notizen oder zur Dokumentation verwendet, da keine spezielle Software, eben nur ein Editor (auf jedem Computersystem vorhanden), für die Anzeige oder Bearbeitung der Texte notwendig ist.

Freiform-Textdateien sind ziemlich universell verwendbar, es gibt jedoch Probleme mit der Kodierung, wenn sie zwischen Betriebssystemen und Computerplattformen ausgetauscht werden.

Das erste Problem ist die Kodierung der Zeichen. Folgende *Zeichensätze* sind heute vorwiegend in Verwendung[21]:

- ASCII: Diese Kodierung wird (mit Erweiterungen) auf praktisch allen heutigen Computerplattformen verwendet. Sie enthält notwendige Kontrollzeichen, die Buchstaben des lateinischen Alphabets, die Satzzeichen und die arabischen Ziffern.

- EBCDIC[22]: Kodierung auf manchen Großrechnern der Firma IBM. EBCDIC ist mit ASCII nicht kompatibel, es gibt jedoch Programme, die Dateien von einer Kodierung in die andere umwandeln können. Eine EBCDIC-Datei erscheint auf einem ASCII-System (und umgekehrt) als sinnlose Anhäufung von Zeichen, und da es auch andere Kodierungssysteme gibt, können unerfahrene BenutzerInnen nicht einfach feststellen, was in der Datei steht und wie sie den Inhalt interpretieren sollen[23].

[20] Ein Editor ist ein Textverarbeitungsprogramm, das keine Funktionen zur Formatierung (z. B. Kursivschrift, unterschiedliche Schriftgrößen etc.) des Dokuments besitzt. Beispiele: Windows Notepad, vi, EMACS.
Editoren werden heute in der Büroarbeit selten eingesetzt, da die echten Textverarbeitungsprogramme den dortigen Aufgaben viel besser gerecht werden. Für Leute mit technischen Tätigkeiten (Systemadministration, Software-Entwicklung) sind Editoren jedoch unerläßlich.

[21] Quelle: auf meinem Computer installierte Info-Seite des Unix-Programms „recode“ (universales Programm zur Konvertierung zwischen über 300 verschiedenen Zeichensätzen)
Diese Info-Seite ist z. B. unter folgender Adresse im World Wide Web abrufbar:
Info Node: recode.info http://olympus.het.brown.edu/cgi-bin/info2www?(recode)

[22] Extended Binary Coded Decimal Interchange Code

[23] Auf UNIX- und darauf basierenden Systemen gibt es ein kleines Programm namens „file“, das über tausend Dateiformate kennt und auch recht zuverlässig die Kodierung von Textdateien feststellen kann. Microsoft Windows enthält, soweit mir bekannt, kein solches Programm.

- ASCII mit internationalen Erweiterungen („erweitertes ASCII“): Ein 8 Bits langes Byte kann 256 verschiedene Zeichen ausdrücken. Das reicht jedoch nicht, um die Zeichen aller Sprachen der Welt, oder auch nur Europas, aufzunehmen. Deswegen wurden von der International Standards Organisation ISO verschiedene internationale Erweiterungen definiert. Sie alle enthalten auf den ersten 128 Stellen die ASCII-Zeichen und geben auf den zweiten 128 die unterschiedlichen Zeichen einer Sprachgruppe an.
  ISO-8859-1 enthält die für Westeuropa notwendigen Zeichen wie ß, ä, á usw. Eine Variante von ISO-8859-1 ist ISO-8859-15, mit dem einzigen Unterschied, daß das Euro-Zeichen enthalten ist.
  ISO-8859-2 enthält die in osteuropäischen Ländern, die das lateinische Alphabet benutzen, gebräuchlichen Zeichen wie ő und ş.
  Weitere ISO-Zeichensätze enthalten kyrillische, arabische, hebräische und griechische Zeichen.
  Es ist nicht möglich, ohne Informationsreduktion zwischen den einzelnen ISO-Zeichensätzen zu konvertieren, da die internationalen Zeichen von z. B. ISO-8859-2 keine Entsprechung in den anderen Zeichensätzen haben. Für die sinnvolle Anzeige eines Dokuments muß auch der Zeichensatz bekannt sein, was bei fremdsprachigen Dokumenten manchmal nicht leicht zu bestimmen ist.

- Unicode oder UCS (Universal Character Set)[24]: Ein einheitlicher Zeichensatz mit dem ehrgeizigen Ziel, alle heutigen und historischen Zeichen, mathematische, technische und weitere spezielle Symbole (z. B. Musiknoten) in einem einzigen Standard zusammenzufassen. Würden alle Dokumente Unicode verwenden (und wären alle Computer in der Lage, damit umzugehen), wäre es nicht mehr notwendig, zwischen Zeichensätzen zu konvertieren. Unicode wird von aktuellen Betriebssystemen und Web-Browsern recht gut unterstützt, aber der größte Teil der heute verwendeten Textdateien benutzt noch eine ASCII- oder erweiterte ASCII-Kodierung oder sogar einen proprietären Zeichensatz.
  Leider wird auch Unicode nicht einheitlich kodiert. Im Internet wird meistens die Kodierungsmethode UTF-8[25] verwendet; Microsoft hat sich in Windows (NT,

[24] Vom Unicode-Konsortium. Webseite des Unicode-Konsortiums http://www.unicode.org/
Das Konsortium arbeitet mit der ISO zusammen, deswegen ist ein Teil des Unicode-Standards auch als ISO/IEC 10646 bekannt.

[25] UTF: Unicode Transformation Format. UTF-8 kodiert die unterschiedlichen Zeichen in einem, zwei oder vier Bytes; die am häufigsten verwendeten Zeichen (die 128 von ASCII) brauchen nur ein Byte, die meisten nicht-ASCII-Zeichen der heute gesprochenen Sprachen zwei, weitere Zeichen vier. Dadurch sind Textdokumente in Englisch, Deutsch und anderen europäischen Sprachen mit dem lateinischen Alphabet kaum größer als in ISO-8859-Kodierungen.

2000 usw.) für UCS-2[26] entschieden.

- Proprietäre Zeichensätze: Manche Hersteller wie Apple, Atari und Microsoft hatten, als noch keine ISO-Standards zur Verfügung standen, oder selbst dann, eigene Zeichensätze entworfen. Heute sind sie auf die ISO-Standards oder auf Unicode umgestiegen[27].

Neben dem Zeichensatz spielt speziell bei Textdateien auch die Herkunft eine Rolle. Windows-basierte Editoren trennen Zeilen standardmäßig mit einem CR[28]- *und* einem LF[29]-Zeichen; Unix-Systeme verwenden nur LF, und MacOS früher nur CR. Editoren, die nicht auf die Konventionen der anderen Systeme vorbereitet sind, zeigen die Dateien häufig etwas seltsam, z. B. ohne Zeilenschaltungen und/oder mit sinnlosen Kontrollzeichen an den Zeilenenden an. Glücklicherweise sind diese Unterschiede Leuten, die mit mehreren Systemen arbeiten, recht bekannt, und die Textdateien sind einfach und ohne Informationsreduktion zwischen den verschiedenen Zeilenenden-Konventionen konvertierbar.

3.5.4 Strukturierte Textdateien

Diese sind meist (auch) als Eingabe für Programme gedacht. Strukturierte Textdateien sind z. B. Programmcode (Eingabe für ein Übersetzer- oder Interpreter-Programm), Konfigurationsdateien (Eingabe für die zu konfigurierende Software), oder auch anwendungsspezifische Formate, wenn die Anwendung ihre Daten im Textformat speichert[30].

Natürlich sind alle Probleme, die unstrukturierte Textdateien betreffen, auch bei strukturierten vorhanden. Hinzu kommt, daß eben das Format der Datei festgelegt ist.

Die Strukturierung der Datei hängt von dem Programm ab, als dessen Eingabe die Datei fungieren soll. Ich beschreibe einige wichtige Arten von strukturierten Textdateien, ohne Anspruch auf Vollständigkeit.

3.5.4.1 Programmcode (Quellcode) Solche Dateien müssen den Syntax-Regeln und allen anderen Vorschriften der jeweiligen Programmiersprache entsprechen. Diese Regeln sind je nach Programmiersprache unterschiedlich streng; in manchen Programmiersprachen kommt es auf die Groß- und Kleinschreibung der Befehle an, in anderen

[26] Eine Kodierung, die für jedes Zeichen fix 2 Bytes vorsieht. Dadurch ist jedes Textdokument zweimal so groß wie in ASCII, aber es werden nicht alle Zeichen des Unicode-Standards abgedeckt.

[27] Siehe z. B. Unicode Enabled Products http://www.unicode.org/onlinedat/products.html

[28] Carriage Return, „Wagenrücklauf“ (wie auf der Schreibmaschine, wo der Schreibkopf an den Anfang der Zeile zurückkehren mußte). ASCII-Code: 13

[29] Line Feed, Zeilenvorschub. ASCII-Code: 10

[30] Dies ist bei Unix-Software meist der Fall, aus Tradition. Unter Windows waren Textformate lange Zeit weniger üblich, sie sind aber auch da auf dem Vormarsch, und zwar dank XML (siehe Kap. 3.5.4.5 auf Seite 36).

nicht, usw. Ohne Kenntnis der Programmiersprache fällt es einem Menschen schwer, die verwendete Sprache zu identifizieren.

3.5.4.2 Konfigurationsdateien Traditionell werden unter Unix die Anwendungen mit Hilfe von Textdateien konfiguriert. Unter Windows war das (in Form sogenannter Ini-Dateien) bis Windows 95 auch üblich (dann wurde die Konfiguration der meisten Programme in eine hierarchische „Datenbank“ namens *registry* verlagert).

Diese Textdateien unter Windows und Unix haben häufig eine solche Struktur, daß am Anfang der Zeile der Name der Einstellung, dann ein Ist-Gleich-Zeichen oder Doppelpunkt und dann der einzustellende Wert steht. Ein Beispiel:

```
font=/usr/lib/j2se/1.4/jre/lib/fonts/LucidaSansRegular.ttf
```

(Aus der Konfiguration eines Videoabspielprogramms auf meinem Computer; die Einstellung bewirkt, daß die genannte Schriftarten-Datei für die Anzeige von Texten im Video (z. B. Laufzeit, Untertitel) verwendet wird.)

3.5.4.3 Separierte Textdateien Eine häufige Anwendung der Computer ist die Erstellung von Listen von Datensätzen. Datensätze sind kleine Sammlungen zusammengehörender Datenfelder, zum Beispiel kann jede Person in einem Adreßbuch ein Datensatz sein; die Datenfelder sind dann etwa Vorname, Nachname, Telefonnummer und Adresse.

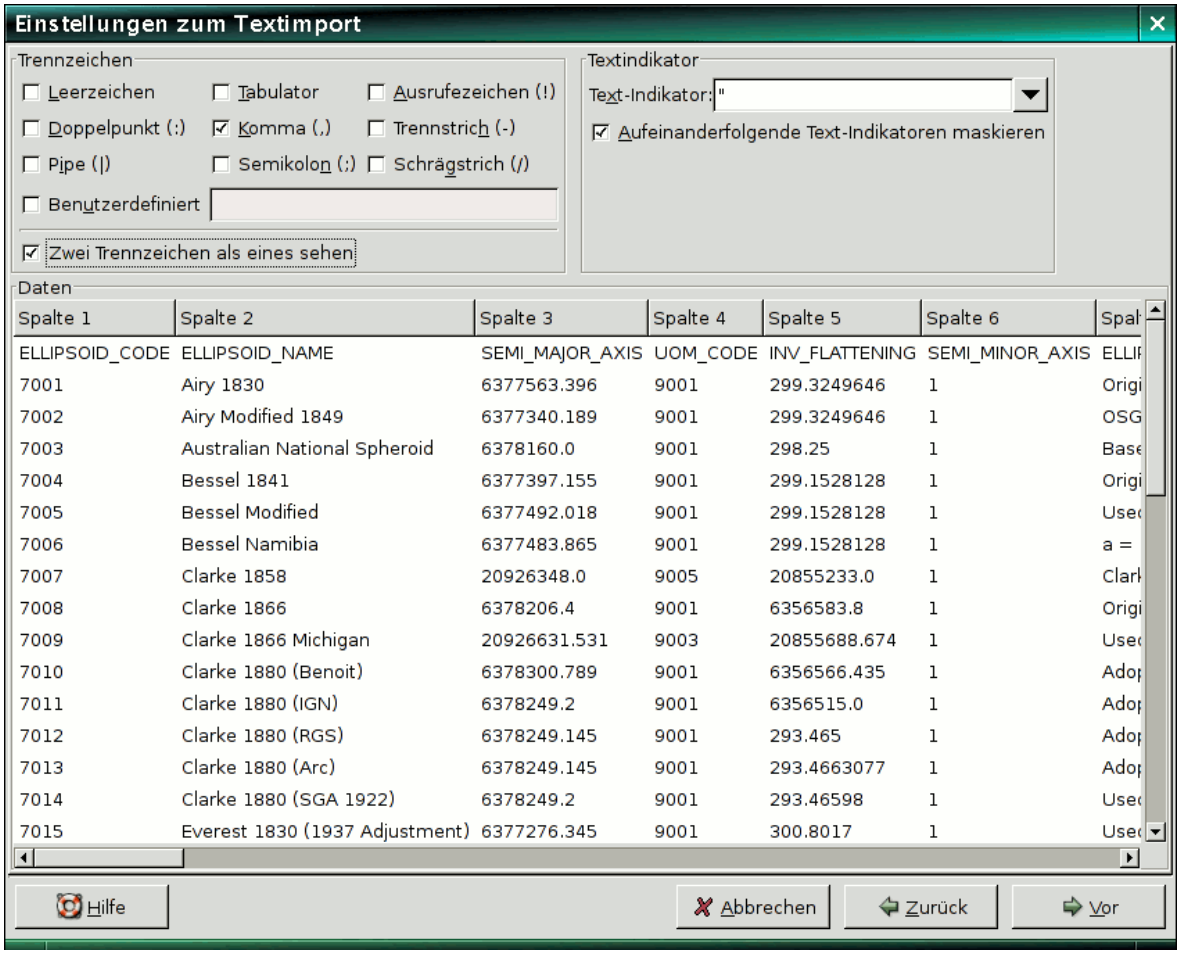
Zahlen werden im Allgemeinen als Ziffern (also nicht in ihrer binären Form) gespeichert.

Solche Listen werden häufig zwischen verschiedenen Programmen ausgetauscht, wofür diese ein gemeinsames Dateiformat verstehen müssen. Wegen der einfachen Struktur von Textdateien und dem daraus resultierenden geringen Programmieraufwand sind separierte Textdateien für den Datenaustausch recht beliebt.

Separiert (getrennt) heißt, daß die Datenfelder mit einem definierten Trennzeichen (häufig sind Komma, Strichpunkt und das Tabulatorzeichen) voneinander getrennt werden. Dabei stellt sich das Problem, daß das Trennzeichen auch in den Datenfeldern selbst vorkommen kann, aber das ist lösbar, z. B. durch geeignete Markierung (*escaping*, *quoting*) der nicht als Trennzeichen gedachten Zeichen oder Beschränkung des Formats der zu speichernden Daten (z. B. nur Zahlen).

Obwohl es sehr viele Variationen von separierten Textdateien gibt (mit unterschiedlichen Trennzeichen; Texte besonders markiert oder nicht; Unix- oder Windows-Zeilenenden; Kodierung usw.), ist ihre Struktur hinreichend einfach und bekannt, um sie für Datenaustausch geeignet zu machen. Viele Programme, die für die Verarbeitung

von Datensatz-Listen geeignet sind, bieten relativ einfach benutzbare, flexible Import-Funktionen.



Spalte 1	Spalte 2	Spalte 3	Spalte 4	Spalte 5	Spalte 6	Spal
ELLIPSOID_CODE	ELLIPSOID_NAME	SEMI_MAJOR_AXIS	UOM_CODE	INV_FLATTENING	SEMI_MINOR_AXIS	ELLI
7001	Airy 1830	6377563.396	9001	299.3249646	1	Origi
7002	Airy Modified 1849	6377340.189	9001	299.3249646	1	OSG
7003	Australian National Spheroid	6378160.0	9001	298.25	1	Base
7004	Bessel 1841	6377397.155	9001	299.1528128	1	Origi
7005	Bessel Modified	6377492.018	9001	299.1528128	1	Usec
7006	Bessel Namibia	6377483.865	9001	299.1528128	1	a =
7007	Clarke 1858	20926348.0	9005	20855233.0	1	Clark
7008	Clarke 1866	6378206.4	9001	6356583.8	1	Origi
7009	Clarke 1866 Michigan	20926631.531	9003	20855688.674	1	Usec
7010	Clarke 1880 (Benoit)	6378300.789	9001	6356566.435	1	Adop
7011	Clarke 1880 (IGN)	6378249.2	9001	6356515.0	1	Adop
7012	Clarke 1880 (RGS)	6378249.145	9001	293.465	1	Adop
7013	Clarke 1880 (Arc)	6378249.145	9001	293.4663077	1	Adop
7014	Clarke 1880 (SGA 1922)	6378249.2	9001	293.46598	1	Usec
7015	Everest 1830 (1937 Adjustment)	6377276.345	9001	300.8017	1	Usec

Abbildung 2: Konfigurierbarer Textimport (Gnumeric Version 1.2.13 unter Linux)

Separierte Textdateien eignen sich selten als Dateiformat komplexerer Programme, da sie im Allgemeinen weder Hierarchien noch von der vorgegebenen Struktur abweichende Daten aufnehmen können.

3.5.4.4 Escape-markierte Textdateien Es gibt einige Arten von Dateien, die zwar zu einem großen Teil Text für Menschen enthalten, aber auch Anweisungen für Computer benötigen, um weitere Funktionalität wie Sprünge in den Dokumenten, Hervorhebung wichtiger Textteile etc. zu realisieren. Solche Formate benutzen genau definierte Zeichen, um das Darstellungsprogramm in einen anderen Modus zu schalten (*escape*: Flucht). In diesem Modus führt das Programm die Anweisungen aus und schaltet dann wieder zum Text um.

Häufig sind Escape-markierte Textdateien im Unix-Umfeld. Zum Beispiel speichert

das Programm LyX, in dem ich diese Arbeit schreibe, diesen Textbereich so:

```
\layout Paragraph
Escape-markierte Textdateien
\layout Standard
Es gibt einige Arten von Dateien, die zwar zu ...
```

Solche Dateien sind notfalls auch lesbar, ohne das dazugehörige Programm installiert zu haben. Gleichzeitig sind sie leicht mit anderen Textwerkzeugen ver- und bearbeitbar, auch wenn das Originalprogramm nicht alle Funktionen anbietet, die mit dem Dateiformat möglich sind.

Das Problem mit escape-basierten Textdateien ist, daß logische Beziehungen wie Dokumenthierarchien mit den Escape-Codes häufig nicht leicht eindeutig auszudrücken sind[31]. Das macht bei komplexeren Dokumenten die Verarbeitung schwieriger und führt eventuell zu falschen Resultaten, wenn die eingegebenen Befehle mehrdeutig sind.

3.5.4.5 Tag-markierte (Markup) Textdateien Da Escape-basierte Textdateien aus den genannten Gründen nicht für alle Arten von Daten gut geeignet sind, wurden Markup-Dateien eingeführt. Sie unterscheiden sich von den Escape-basierten Dateien dadurch, daß jeder Moduswechsel (vom Dokumenttext zu den Programmanweisungen und zurück) mit Anfang- *und* Ende-Markierungen (engl. *tag*) versehen ist. Auf diese Weise kann die Dokumentstruktur viel genauer angegeben und auch maschinell überprüft werden.

Wie in Textdateien üblich, speichern tag-basierte Dateien Zahlen mit Ziffern, nicht binär.

Die erste erfolgreiche Markup-Sprache war die Standard Generalized Markup Language SGML. Sie wurde 1986 zum internationalen Standard (vgl. [Kasd98]).

SGML legt nicht fest, welche Sprachelemente in einem Dokument vorkommen können und was sie bedeuten, sondern nur die Syntax der Sprachelemente und ihre zulässige Anordnung in der Datei. Deswegen müssen für einzelne Aufgaben dazugehörige SGML-„Anwendungen“ definiert werden. Dies passiert in einer Document Type Definition, DTD[32]. Die DTD beschreibt, welche Tags in welcher Anordnung im Dokument vorkommen können und welche zusätzliche Attribute und Tags sie enthalten dürfen.

[31] Es kann dafür zusätzliches, anwendungsbezogenes Wissen notwendig sein. Im LyX-Beispiel etwa muß die Software wissen, wie die Absatzarten (z. B. `Paragraph` und `Standard`) hierarchisch zusammenhängen.

[32] Es gibt neben dem textbasierten DTD-Standard einen neueren markup-basierten namens XML-Schema. Beide dienen demselben Zweck, derzeit ist DTD noch weit verbreitet, aber mit der weiteren Verbreitung von XML ist es denkbar, daß sich XML-Schema durchsetzt, weil es selbst in XML formuliert ist und dadurch das Lernen einer weiteren „Sprache“ unnötig macht.

Ein Beispieldokument in SGML (Ausschnitt aus der Beispieldatei für das SGML-basierte Dokumentationssystem LinuxDoc[33]):

```
<!doctype linuxdoc system>
<!-- Here's an SGML example file. Format it and print
out the source, and use it as a model for your own
SGML files. As you can see this is a comment. -->
<article>
<title>Quick Example for Linuxdoc DTD SGML source</title>
<author>
<name>originally written by Matt Welsh as
&dquot;Quick SGML Example&dquot;,<newline></name>
<and>
<name>recently updated by Taketoshi Sano for linuxdoc-
tools<newline></name>
</author>
[...]
</article>
```

Die `!doctype`-Zeile gibt an, welchem Schema (in diesem Fall „linuxdoc“) das Dokument entspricht. Darauf folgt, zwischen `<!--` und `-->`, ein Kommentar, dessen Inhalt nicht im fertig formatierten Dokument erscheint. (Kommentare werden vom Computer nicht beachtet, sie enthalten Hinweise für Menschen, die die Datei lesen. In strukturierten Dateien könnte freier Text sonst nicht leicht untergebracht werden.) Danach beginnt, mit `<article>` markiert, der für die Ausgabe relevante Teil der Datei. Dieser Teil wird am Ende mit `</article>` abgeschlossen.

SGML hat sich insbesondere für technische Dokumentationsaufgaben bewährt. Da sie jedoch relativ komplex ist und so viele DTDs existieren, hat sie sich in anderen Bereichen in ihrer Originalform nicht durchsetzen können.

Mit dem Siegeszug des World Wide Web hat sich jedoch eine SGML-Anwendung namens Hypertext[34] Markup Language HTML weit verbreitet. HTML hat einen (anfangs) relativ begrenzten Vorrat an verwendbaren Tags definiert, etwa um Dokumentteile (z. B. `<head>`, `<body>`), Textelemente (`<h1>`, `<p>`), Dokumenteneigenschaften (`<title>`, `<link>`), Formatierungen (`<br>`, `<b>`, `<big>` usw.) und, ganz wichtig für Hypertext, Verknüpfungen (`<a href...>`) festzulegen. Weiters definiert HTML sogenannte Entitäten (*entities*) für Zeichen, die die Sprache selbst benutzt (z. B. `>` = `&gt;`

[33] LinuxDoc erlaubt, aus einer standardkonformen Quelldatei unterschiedliche Arten von Dokumenten (Text, HTML, PDF etc.) für verschiedene Anwendungen zu generieren.

[34] Hypertext enthält im Gegensatz zu normalem Text Verknüpfungen zwischen Textteilen und zu komplett anderen Texten, um so die Linearität der einfachen Texte zu durchbrechen. Diese Eigenschaft war wesentlich am Erfolg des hypertext-basierten World Wide Web beteiligt.

und < = <), die zum erweiterten ASCII-Zeichensatz gehören (z. B. ä = ä) oder andere Aufgaben erfüllen (z. B. das Copyright-Zeichen ©).

Der HTML-Standard ist immer komplexer geworden (siehe etwa [Wor99]), deswegen wurde es immer schwieriger, die Dokumente zu verifizieren (maschinell zu überprüfen, ob sie den Regeln des aktuellen HTML-Standards entsprechen). Gleichzeitig wurden für den immer wichtiger werdenden plattformunabhängigen Datenaustausch neue Sprachen notwendig, weil HTML als Textauszeichnungssprache für manche Aufgaben des Datenaustauschs nicht gut geeignet ist.

Aus diesem Bedarf ist die Extensible Markup Language XML [Bra+04] entstanden. XML ist eine Vereinfachung von SGML; die Erfahrungen mit SGML haben geholfen, das Gleichgewicht zwischen Funktionalität und Einfachheit zu finden. Wie SGML (und im Gegensatz zu HTML) legt XML keine Sprachelemente fest, sondern nur die Syntax und die Anordnung der Elemente. XML benutzt auch DTDs.

Die Einfachheit von XML gegenüber SGML hat ermöglicht, schnell Programme zu schreiben oder anzupassen, die damit umgehen können. Gleichzeitig ist es einfacher als bei anderen Dateiformaten, die Gültigkeit des Dokuments (Einhaltung des Standards, vollständige Übertragung, alle notwendigen Informationen angegeben etc.) zu verifizieren. Manche Software, die früher (mangels verwendbarer Textformate oder aus Wettbewerbsgründen) eher binäre Dateiformate benutzt hat, wurde in den letzten Jahren auf die Speicherung XML-basierter Dateiformate umgestellt. Das beinhaltet die aktuellen Versionen von Bürosoftware wie OpenOffice.org (Standardformat ist XML-basiert) oder Microsoft Office (Export in XML ist wählbar).

Auch HTML wurde unter dem Namen XHTML (Extensible Hypertext Markup Language, [Wor02]) in XML neu formuliert. Durch die gemeinsame SGML-Basis ergaben sich dadurch keine größeren Änderungen (vgl. [Wor02, Kap. 4]), sodaß auch alte Internet-Browser die neuen XHTML-Dokumente anzeigen können; gleichzeitig profitiert neue Software von der Eindeutigkeit und einfacheren Verarbeitung des XML-basierten Formats.

Tag-basierte Formate wie XML haben den Vorteil, daß sie leicht erweiterbar sind (das „Extensible“ im Namen von XML soll auch das ausdrücken). Es können neue Tags eingeführt werden, wenn die Software neue Funktionen hat; alte Versionen können die Tags, die sie nicht verstehen, einfach ignorieren und immer noch die für sie relevanten Informationen auslesen.

Ein Nachteil von markup-basierten gegenüber anderen Textdateien und besonders gegenüber binären Dateien ist ihr etwas größerer Platzverbrauch. „Sprechende“ tag-Namen und die Notwendigkeit, alle tags abzuschließen, verursachen einen gewissen Mehraufwand, der sich allerdings bei den heutigen Speicherkapazitäten kaum auswirkt.

Markup-basierte Formate sind kaum für Dateien geeignet, die nur oder fast nur aus binären oder numerischen Daten bestehen und bei denen es auf die Dateigröße oder die Zugriffszeit ankommt. Das sind fast alle Bild- und Videoformate, interne Formate von Datenbanken, Archivdateien, Programmdateien usw. Bei solchen Formaten hätte eine Speicherung z. B. in XML (auch ohne Rücksicht auf die Dateigröße) jedoch auch keine Vorteile, weil die endlosen Zahlenreihen in diesen Dateien für Menschen sowieso nicht interpretierbar sind.

3.5.5 Binäre Dateiformate

In den ersten Jahrzehnten der elektronischen Datenverarbeitung war Speicherplatz knapp und teuer. Die ProgrammiererInnen trachteten danach, möglichst wenig davon zu verbrauchen, um mit dem eingeschränkten Arbeitsspeicher des Computers und dem Platz auf den Datenträgern auszukommen. (Vgl. [Wett97, S. 739])

Wenn ein Programm genau weiß, in welchem Format Daten in einer Datei oder auf einem Speichermedium gespeichert sind, braucht es keine Kennzeichnungen für Datenfeld- und Datensatzgrenzen, Zeilenenden, Kodierung usw., wodurch Speicherplatz gespart werden kann. Allerdings ist dadurch die Datei für Menschen nur lesbar, wenn ihnen dieselben Informationen über den Aufbau der Datei zur Verfügung stehen und sie die (nicht unerheblichen) Kenntnisse haben, eine technische Datenstrukturbeschreibung zu verstehen; selbst in diesem Fall wird es mühsam.

In binären Dateien stehen also Informationen in einem Format, das implizit (im Programmcode) oder in besseren Fällen explizit (in Standards oder in der Software-Dokumentation) festgelegt wurde. Es wird meist versucht, für alle Datentypen (z. B. Zahlen, Texte, Ja/Nein-Angaben) eine möglichst speichersparende Form zu finden.

Zahlen werden im Computer anders gespeichert als wenn sie in einem Text stehen. Im Text sind die Ziffern nur allgemeine Zeichen, sie verbrauchen (im Dezimalsystem ausgedrückt) pro Stelle ein Byte. Wenn ein Computer jedoch Zahlen als Zahlen verarbeitet, tut er das in Größenordnungen von einem, zwei, vier, acht oder mehr Bytes. Mit einem Byte können Zahlen von -127 bis 127 (oder 0 bis 255), mit zwei von -32.767 bis 32.768 (oder 0 bis 65.535) usw. ausgedrückt werden. Es liegt auf der Hand, daß es Speicherplatz spart, die Zahlen in der internen Darstellung des Computers (also „binär“) zu speichern[35]. Außerdem müssen sie so für die Verarbeitung im Programm nicht aus Text in Zahlen und zurück konvertiert werden, das verkürzt die Verarbeitungszeit, was früher auch ein wichtiger Faktor war.

Allerdings gibt es keine eindeutige Methode, Zahlen binär zu speichern. Die Proble-

[35] Es gibt auch andere Darstellungsarten für Zahlen, etwa interne Dezimaldarstellungen. Diese Darstellungen sind in einer Datei jedoch genausowenig als Text erkennbar wie binäre Zahlendarstellungen.

matik mit little-endian und big-endian wurde bereits beschrieben (siehe Fußnote auf der Seite 12). Es ist auch unklar, ob eine Zahl mit Vorzeichen (z. B. -127 bis 127) oder ohne (z. B. 0 bis 255) gespeichert wurde, oder wie sie interpretiert wird (z. B. haben früher manche Systeme in einem Byte die Anzahl der Jahre nach 1900 oder 1970 gespeichert).

Zeichenketten in Datenfeldern, z. B. Titel von Dokumenten oder Namen von Menschen sind meist unterschiedlich lang. Um sie möglichst sparsam zu speichern, werden unterschiedliche Methoden verwendet: die eine ist, die Länge der Zeichenkette in einem oder zwei Bytes vor der Zeichenkette zu speichern (das beschränkt die mögliche Länge des Textes), die andere, daß das Ende der Zeichenkette im Speicher mit einem speziellen Zeichen (z. B. dem Null-Zeichen) gekennzeichnet („terminiert“) wird (dadurch kann dieses Zeichen natürlich nicht im Text selbst vorkommen). Andererseits gibt es einige beliebte Programmiermethoden, auf große Datenmengen schneller zuzugreifen, die nur funktionieren, wenn die Datensatzlänge fix ist. In diesem Fall werden alle Zeichenketten auf eine bestimmte Anzahl von Zeichen beschränkt; längere Texte können nicht eingegeben werden und kürzere verbrauchen trotzdem die volle Länge.

Daten, die nur Ja oder Nein speichern, sind in der Informatik recht häufig. Sie können mit einem Bit ausgedrückt werden, das bedeutet, daß bis zu 8 solche Informationselemente in ein Byte hineinpassen. Es ist zwar nicht ganz einfach, einzelne Bits in einem gespeicherten Byte zu manipulieren, aber für den ersparten Speicherplatz gingen die ProgrammiererInnen früher diesen Kompromiß ein.

Für die Speicherung von Datums- und Zeitangaben gibt es – je nach gewünschtem Zeitrahmen und Präzision – eine extrem große Anzahl von Möglichkeiten: Eine Dokumentation der Wiederherstellung von Daten aus der DDR-Zeit [Wett97] listet für eine einzige Datei drei verschiedene Datumsdarstellungen auf. Der Grund ist wieder das Einsparen von Speicherplatz: ein Datum, das in Textform (z. B. „20040731“) 8 Bytes braucht, kann mit einigen Tricks auf zwei Bytes „geschrumpft“ werden[36].

Alle diese Elemente (Zahlen mit unterschiedlicher Byte-Anzahl, Texte, aus Bits zusammengesetzte Bytes) und noch mehr können nun in einer binären Datei drinnenstehen. Ohne Dokumentation des Formats ist es für den Menschen extrem aufwendig (arbeits- und wissensintensiv) oder unmöglich, die Bedeutung von einzelnen Bytes herauszufinden (vgl. [Wett97, S. 739] oder [Roth95a, S. 28]). (Das soll nicht heißen, daß es diese Schwierigkeiten nur bei binären Dateiformaten gibt. Textdateien oder Mischformen können durchaus auch schwer erkennbare Strukturen haben.)

[36] Eine Methode ist, 7 Bits für die Jahreszahl nach 1900 oder 1970 (übliche Basis unter UNIX) und 9 für den Tag des Jahres zu verwenden. Damit lassen sich Tage über 127 Jahre (jeweils nach dem Basisdatum) in zwei Bytes auszudrücken.

3.5.6 Nicht-Text-Formate für Textdokumente (und andere Bürosoftware)

Heutige Textverarbeitungsprogramme[37] bieten eine Unzahl von Funktionen an, um Texte zu formatieren, mit weiteren Informationen zu versehen, Sprünge zwischen Dokumentteilen herzustellen etc. Die Dokumente, die sie erstellen, enthalten neben dem aktuellen Text des Dokuments häufig auch eine komplette Änderungsgeschichte, eingebettete Bilder und manchmal sogar Programmteile (Makros). Diese Informationen lassen sich nicht in einfachen Textdateien speichern. Deswegen hat heute jedes verbreitete Textverarbeitungsprogramm ein eigenes Dokumentformat, das seine Anforderungen erfüllt. Der größte Teil dieser Formate ist binär, erst in den letzten Jahren sind Programme entwickelt worden, die ihre Dokumente in XML-Dateien oder in Archivdateien mit XML-Inhalt speichern können.

Da die Formatvielfalt schon seit einiger Zeit störend ist, haben sich viele Hersteller auf ein textbasiertes Format namens RTF (Rich Text Format) geeinigt, das die meisten Formatierungen speichern kann. Die Produkte dieser und der meisten anderen Hersteller unterstützen RTF mehr oder weniger gut, aber beim Export in RTF gehen fortgeschrittene Funktionen wie Bearbeitungsgeschichte, Makros usw. verloren.

Es ist normalerweise nicht einfach, Unterstützung für neue Funktionen in binäre Dateiformate einzubauen. Aus diesem Grund bedeutete früher jede neue Version eines Textverarbeitungsprogramms auch ein neues Dateiformat. Theoretisch sollte das bei neuen XML-basierten Dateiformaten nicht notwendig sein, aber neue Dateiformate haben für Hersteller mit großer Marktmacht auch einen nicht-technischen Vorteil: Sobald eine „kritische Masse“ an AnwenderInnen erreicht wurde, die ihre Dokumente im neuen, nur von der neuesten Version unterstützten Format erstellen und an andere weitergeben, steigt damit der Druck auf alle anderen, auch auf diese neueste Version umzusteigen.

Es bleibt abzuwarten, ob sich XML-basierte Dateiformate für Textverarbeitungsdokumente durchsetzen. Die OASIS-Gruppe (Organization for the Advancement of Structured Information Standards) hat eine Spezifikation unter dem Namen „Open Office XML Format“ (nicht zu verwechseln mit der Bürosoftware „OpenOffice.org“) erstellt[38], und zwei Projekte (OpenOffice.org und KOffice) haben bereits angekündigt, das Format in Zukunft zu unterstützen. Der Erfolg hängt jedoch vor allem vom Verhalten des Marktführers (Microsoft Word) ab: Entscheidet sich Microsoft, das XML-basierte neue Format als Standard einzustellen, werden die AnwenderInnen älterer Versionen diese

[37] Was ich in diesem Kapitel über Textverarbeitungsprogramme schreibe, gilt fast unverändert für andere Bürosoftwareprodukte, also etwa Tabellenkalkulationen oder Präsentationsprogramme. Die Formate und die Probleme mit ihnen sind in der Regel ähnlich.

[38] OASIS Open Office XML Format TC http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=office

Dokumente nicht lesen können. (Andere Hersteller bauen erfahrungsgemäß innerhalb von Monaten die Unterstützung neuer Microsoft-Formate in ihre Produkte ein. Das sollte bei einem XML-basierten Format sogar noch einfacher sein.)

In der Gegenwart dominieren jedoch Textverarbeitungsdokumente in binären Formaten wie Microsoft Word 97, WordPerfect usw.

Ein Problem, das sich in der Zukunft eventuell stark äußern wird, besteht in der Verschlüsselung von Dokumenten. Die Möglichkeit, ein Dokument mit Paßwort zu schützen, besteht schon länger, wurde aber nur bei Bedarf verwendet, weil es einfach mühsam ist, sich Paßwörter zu merken, sie bei jedem Öffnen der Datei einzugeben und beim Dokumentenaustausch mitliefern zu müssen. Heute stehen jedoch Technologien mit Namen wie „Information Rights Management“[39] (IRM) zur Verfügung, die verhindern sollen, daß in einer Organisation erstellte vertrauliche Dokumente nach außen gelangen (vgl. [Bror03]). Diese Lösungen funktionieren so, daß die Dokumente mit Schlüsseln, die ein zentraler Dienst (innerhalb der Organisation oder beim Hersteller der Software) vergibt, verschlüsselt werden; ihre Entschlüsselung ist ausschließlich mit Hilfe desselben Dienstes möglich. Das Lesen der Dokumente stößt also auf eine weitere, ziemlich große Hürde, die auch für die aktuelle Benutzung große Probleme bedeuten kann (z. B. alle leseberechtigten MitarbeiterInnen sind auf Urlaub oder nicht erreichbar; Netzwerkausfall; Ausfall des Computers, der die Nutzungsrechte vergibt; Datenverlust am IRM-Server etc.).

3.5.7 Dokumentformate für seitenweise Ausgabe

Es gab schon vor Jahrzehnten eine große Anzahl von Betriebssystemen, Textverarbeitungsprogrammen und Druckern. Damit beliebige Programme auf beliebigen Druckern etwas ausgeben können, sind entweder Standards nötig, oder die Unterstützung für jeden Drucker muß in jedes Programm eingebaut werden. Ersteres ist deutlich effizienter, weswegen schon recht früh Dokumentformate zur Beschreibung von Seiten geschaffen wurden.

Diese Formate sind dazu gedacht, die gedruckte Ausgabe beliebiger Programme möglichst geräteunabhängig darzustellen. Die Dokumente enthalten alle Informationen, die für eine bestmögliche Ausgabe sowohl auf kleinen Bürodruckern als auch auf Belichtungsgeräten in großen Druckereien sorgen sollen, also alle Texte, die exakte Position aller Buchstaben, die Seitengröße, Bilder und in manchen Fällen sogar die notwendigen Schriftarten.

In der ersten Hälfte der 1980-er-Jahre entstanden DVI (Abkürzung von „device in-

[39] z. B. in Microsoft Office 2003: Information Rights Management in Office Professional Edition 2003 http://www.microsoft.com/office/editions/prodinfo/technologies/irm.mspx

dependent“, gerät-unabhängig) und PostScript.

DVI, ein binäres Format, wurde fast nur in Verbindung mit dem Satzssystem TEX verwendet. Es gibt zwar Anzeigeprogramme für jedes verbreitete Betriebssystem, aber DVI hat sich wegen der Bindung an TEX nicht breit durchgesetzt. Mittlerweile steigen viele TEX-AnwenderInnen auch wegen der Vorteile des neueren PDF-Formates auf die Generierung von PDF-Dateien um. Es liegen trotzdem viele technische, mathematische und andere Publikationen im DVI-Format vor, was jedoch für die absehbare Zukunft keine großen Probleme bedeutet, da das Format gut dokumentiert ist und mehrere DVI-Anzeigeprogramme im Quellcode vorliegen. Es gibt auch frei verfügbare Konvertierprogramme, die DVI-Dateien in PostScript oder PDF umwandeln können.

Etwas später entstand PostScript mit dem Ziel, alle für die Steuerung von Druckern notwendigen Aufgaben zu lösen. PostScript ist eine vollständige Programmiersprache, also nicht nur für die Beschreibung von Seiteninhalten geeignet. PostScript-Dateien sind Textdateien, bestehend aus PostScript-Befehlen sowie kodierten Datenblöcken für binäre Daten. Es gibt drei abwärtskompatible Versionen (Postscript 1, 2 und 3), neuere PostScript-Geräte können mit älteren Dokumenten umgehen.

PostScript hatte großen Erfolg, es konnte sich im professionellen Druckbereich stark durchsetzen. Insbesondere im Unix-Bereich sind viele Programme entstanden, um PostScript-Dateien zu generieren; die meisten Drucksysteme auf Unix-Systemen basieren überhaupt auf PostScript.

Als Dokumentenaustauschformat hat PostScript einige Nachteile, aber für diesen Zweck wurde es auch nicht konzipiert. Da es ein textbasiertes Format ist, sind die Dateien ziemlich groß und nur sequenziell lesbar (zum „Blättern“ in einem Dokument müssen alle vorherigen Seiten durchgelesen werden), und durch die konsequente Ausrichtung auf das Drucken unterstützt es heute übliche Funktionalitäten wie Hyperlinks und andere Formen der Interaktivität nicht.

Die Firma Adobe, die auch schon PostScript entwickelt hat, veröffentlichte 1993 die Spezifikationen des Portable Document Format PDF in der Version 1.0. PDF wurde als wirklich universelles Dokumentenaustauschformat, also auch für Anwendungen, die übers Drucken hinausgehen, konzipiert. Adobe publiziert die Spezifikationen frei zugänglich im Internet[40] und gibt sie auch als Bücher heraus; es ist ohne Beschränkungen erlaubt, Programme zu schreiben, die PDF-Dokumente lesen, schreiben und verarbeiten. Adobe hält allerdings Patente auf einzelne Aspekte der PDF-Technologie; manche von diesen sind gratis zur allgemeinen Verwendung lizenziert, solange die Programme, die sie implementieren, für die Verarbeitung von PDF-Dateien bestimmt sind[41]. In die-

[40] Adobe PDF - Specifications http://partners.adobe.com/asn/tech/pdf/specifications.jsp
[41] Legal Notices for Developers http://partners.adobe.com/asn/developer/legalnotices.jsp

sem Bereich gibt es trotzdem Potenzial für Probleme, wenn Adobe beschließen sollte, sich durch die restriktivere Handhabung der Patentrechte am Markt besser durchsetzen zu können. Aus diesem Grund kann PDF nicht uneingeschränkt als offene Lösung gelten.

PDF scheint sich stark durchzusetzen, da es weit verbreitet und beliebt ist und viele technische Vorteile hat. PDF-Dokumente enthalten alle für Druck und Anzeige notwendigen Informationen, können relativ stark komprimiert werden, und Produktions-, Verarbeitungs und Anzeigesoftware unterschiedlicher Hersteller stehen als Open Source, gratis oder relativ günstig für alle relevanten Betriebssysteme zur Verfügung. In neueren Versionen können PDF-Dateien auch „in einem Stück“ wie Webseiten erscheinen, ohne die am Bildschirm sinnlose Aufteilung in einzelne Seiten.

Laut Spezifikation [Ado03, S. 948] sind die bisher erschienenen Versionen des Dateiformats und der Anzeigeprogramme so weit untereinander kompatibel, daß ältere Dokumente ohne Probleme in neueren Versionen des Anzeigeprogramms angezeigt werden können. Dokumente in neueren Formaten könnten mit älteren Anzeigeprogrammen angezeigt werden, allerdings würden Teile dieser Dokumente, die auf neue Funktionen angewiesen sind (z. B. durchsichtige Objekte in PDF 1.4 und höher) nicht korrekt erscheinen. Diese Angabe stimmt jedoch nicht: Adobe selbst bietet genau diese Spezifikation in zwei Versionen an, weil ältere Versionen des Anzeigeprogramms die neuere Version (im PDF-1.5-Format) wegen veränderter Strukturen nicht lesen können.

Jedenfalls sollte es keine Probleme geben, wenn eine aktuelle Version des Anzeigeprogramms verwendet wird.

3.5.8 Bildformate für Rasterbilder (Bitmap-Bilder)

Rasterbilder (Bitmaps) entstehen, indem das darzustellende Bild in Bildpunkte (sog. Pixel) aufgeteilt wird, deren Farbwerte (z. B. aus Rot, Grün und Blau zusammengesetzt) gespeichert werden. Wieviele Bildpunkte es sind, hängt von der Auflösung des Bildes ab.

Diese Information über die Bildpunkte macht für Menschen im Allgemeinen wenig Sinn: das Bild wird nur sichtbar, wenn es als Bild interpretiert wird. Aus diesem Grund ist es auch nicht sinnvoll, textbasierte Formate zu verwenden, weil wir auch mit Angaben wie „`rot:10, grün:170, blau:85; rot:11, grün:180, blau:90;` usw.“ nichts anfangen könnten. Das angegebene Beispiel (zugegebenermaßen extrem) verbraucht außerdem 22 bis 28 Bytes für einen einzigen Pixel, während bei einer binären Speicherung der Pixel ohne Kompression nur drei Bytes gebraucht werden (unter der Annahme, daß für jede Farbe 256 Farbstufen gespeichert werden; im professionellen Bereich wird manchmal mit 2 oder sogar 4 Bytes pro Farbstufe gearbeitet).

Es gibt historisch bedingt eine große Anzahl von Rasterbildformaten[42]. Natürlich sind nur einige von ihnen weit verbreitet, aber wir müssen davon ausgehen, daß in den meisten Formaten zumindest einige Bilder existieren.

Ein wesentlicher Unterschied ist, ob die Bilddaten *unkomprimiert*, *komprimiert* oder mit einem *reduzierenden* Algorithmus gespeichert sind. Da Bilddateien recht groß werden können, werden unkomprimierte Formate selten für den Datenaustausch benutzt. Reduzierende Formate werden auch als „verlustbehaftet“ und die anderen dementsprechend als „verlustfrei“ bezeichnet.

Der Unterschied zwischen Kompression und Reduktion ist, daß ein komprimiertes Bild nach der Dekompression wieder mit dem unkomprimierten Original identisch ist. Bei der Reduktion ist das nicht der Fall: hier werden mit Hilfe von komplexen Algorithmen die für Menschen nicht gut sichtbaren Informationen (z. B. kleine Farbänderungen) reduziert, bei der Konvertierung zurück in ein unkomprimiertes Format ist das reduzierte Bild mit dem Original nicht mehr identisch.

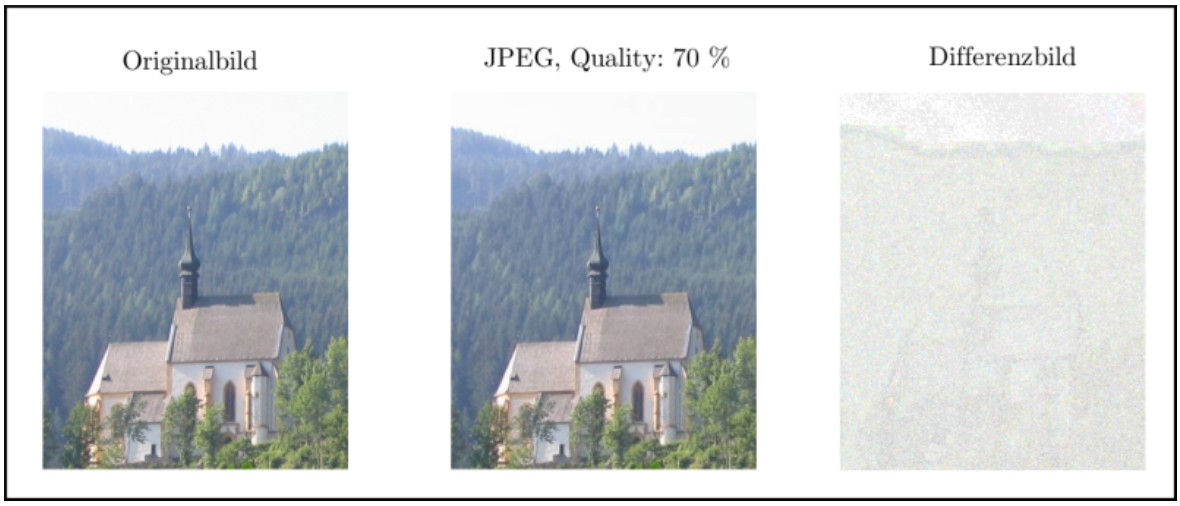


Abbildung 3: Unterschied zwischen dem Original und dem mit 70 % Qualität gespeicherten JPEG-Bild

Für das Beispiel in Abbildung 3 habe ich den Original-Bildausschnitt in einem reduzierten Format (JPEG, Qualitätseinstellung: 70) gespeichert und dann im Bildbearbeitungsprogramm GIMP mit dem Original verglichen. Die Farbwerte der Bildpunkte des einen Bildes wurden aus den anderen subtrahiert, das sich so ergebende Bild invertiert und, um die geänderten Punkte besser hervorzuheben, der Kontrast geändert. Wo das Differenzbild nicht weiß ist, besteht zwischen dem Original und dem reduzierten JPEG-Bild ein Unterschied. (Natürlich nicht in der abgebildeten Intensität; die Un-

[42] Das frei verfügbare Konvertierprogramm ImageMagick beherrscht laut Webseite 89 Formate, das sind ziemlich sicher nicht alle, die jemals entwickelt wurden.
ImageMagick - Image Formats http://www.imagemagick.org/www/formats.html

terschiede sind in Wirklichkeit sehr klein.) Das menschliche Auge vermag jedoch noch keine Verschlechterung des Bildes festzustellen.

Der Vorteil der reduzierenden Verfahren liegt in der wesentlich kleineren Dateigröße. Das Bild braucht in einem gut komprimierenden verlustlosen Format (PNG) 169.517 Bytes, als JPEG mit Qualitätsstufe 70 (auf einer Skala von 0 bis 100) nur 15.565 Bytes. (Mit Qualitätsstufe 80 auch nur 26.231 Bytes.)

Während der Bearbeitung und für die Archivierung sollten, wenn möglich, verlustfreie Formate benutzt werden, da die Reduktionen sich, ähnlich wie bei analogen Verfahren, während der wiederholten Bearbeitung summieren können. Das Endergebnis kann dann bei Bedarf für die Darstellung z. B. im WWW – damit die Übertragungszeit klein bleibt oder mehr Bilder auf ein Speichermedium passen – in ein verlustbehaftetes Format gebracht werden (das verlustlose Original soll natürlich als Archivkopie bestehen bleiben).

Leider bieten viele der heute üblichen Amateur-Digitalfotoapparate nur das JPEG-Verfahren zur Speicherung der Bilder an. Das bedeutet, daß die BenutzerInnen gar keine Möglichkeit haben, an ein nicht reduziertes Bild zu kommen. Glücklicherweise wird meistens eine so kleine Reduktion durchgeführt, daß die entstehenden Bilder ohne sichtbare Verluste weiterverarbeitet werden können. Für häufige Bearbeitung kann es aber sinnvoll sein, das Original aus der Kamera in einem verlustfreien Format zu speichern und dann bewußt nur damit weiterzuarbeiten.

Einige der verbreiteten Rasterbildformate sind so wichtig, daß ich sie separat beschreiben möchte.

3.5.8.1 BMP (Bitmap) Darunter wird meistens das Bitmap-Format von Microsoft Windows verstanden. Es gibt einige Varianten, je nach Anzahl der speicherbaren Farben und nach Kompression (unkomprimiert oder mit einer einfachen Methode komprimiert). BMP-Dateien brauchen wegen der schlechten oder nicht vorhandenen Kompression im Vergleich zu anderen Formaten viel Platz, aber frühere Versionen der mit Windows mitgelieferten Werkzeuge konnten nur BMPs speichern. Daher kommen BMP-Bilder großteils nur in reinen Windows-Umgebungen, die nicht viel Wert auf Datenaustausch legen, vor. Die Spezifikation ist offen zugänglich und es gibt frei verfügbare Implementierungen.

3.5.8.2 TIFF (Tag Image File Format) TIFF (vgl. [Bor+03, S. 110]) wurde mit dem Anspruch geschaffen, ein universell verwendbares Format für den Austausch von Bitmap-Bildern zu bieten. Die öffentlich zugängliche Spezifikation beschreibt das Format der Dateien und die Algorithmen für die (optionale) Komprimierung oder Reduktion der Bilddaten. Weitere Angaben wie Farbräume, Beschreibungen in Textform usw.

können in TIFF-Dateien enthalten sein.

Durch die Flexibilität des Formats ist TIFF für verschiedene Aufgaben geeignet, aber gleichzeitig ist es ziemlich komplex, weswegen es zu Problemen mit dem Datenaustausch zwischen unterschiedlichen Programmen kommen kann.

In der neuesten TIFF-Spezifikation (Version 6.0) ist die Möglichkeit hinzugekommen, die Daten mit dem JPEG-Algorithmus zu speichern. Seitdem können also verlustbehaftet gespeicherte TIFF-Dateien vorkommen; dieser Umstand sollte in der Archivierung festgestellt und entsprechend behandelt werden.

3.5.8.3 GIF (Graphics Interchange Format) Dieses Format wurde in den 1980-er-Jahren von der Firma Compuserve geschaffen. Compuserve war ein früher Anbieter von Online-Dienstleistungen, weswegen es beim Format vor allem auf die Dateigröße ankam, um schnelle Übertragung über die damals ziemlich langsamen Kommunikationswege zu erreichen. Aus dieser Tradition heraus hat GIF sich für manche Arten von Bildern im Internet sehr weit verbreitet.

GIF ist ein Dateiformat, das die Wahl zwischen der Kompressionsmethode LZW[43] und dem Verzicht auf Kompression läßt. In der Praxis sind jedoch nur die LZW-komprimierten GIF-Dateien verbreitet. Das war bis zum Sommer 2004 ein Problem, da bis dahin der LZW-Algorithmus patentrechtlich geschützt war, und die Firma Unisys, Inhaberin des Patents, verlangte Lizenzgebühren, wodurch manche Arten von freier oder billiger Software nicht in der Lage sein konnten, LZW-komprimierte GIF-Bilder zu erzeugen.

Für Fotos und andere „natürliche“ Bilder ist GIF nicht gut geeignet, da es ein „index-basiertes“ Format ist. Jede GIF-Datei enthält eine Tabelle von maximal 256 Farben, die die im jeweiligen Bild benötigten Farbwerte definiert. Die einzelnen Bildpunkte geben dann nur mehr die Nummer der definierten Farbe an. Deswegen wird pro Bildpunkt nur ein Byte verbraucht; dafür muß die Anzahl der Farben reduziert werden. (Dieses Verfahren heißt „dithering“.)

Eine GIF-Datei kann mehrere Bilder enthalten, die „animiert“ werden können, indem sie in definierten Zeitabständen gezeigt werden.

Im Internet ist GIF für Navigationselemente, Animationen und kleinere Bilder, bei denen die Beschränkung auf 256 Farben kein Problem ist, sehr verbreitet. Es gibt frei zugängliche Softwarekomponenten für das Lesen und Schreiben von GIF-Dateien.

3.5.8.4 PNG (Portable Network Graphics) PNG entstand als internationaler Standard ([DuBo03]) aus dem Wunsch heraus, die Nachteile von GIF (lizenzpflichtig, be-

[43] Abkürzung für Lempel-Ziv-Welch (die Erfinder des Algorithmus)

schränkt auf 256 Farben) zu vermeiden und ein allgemein verwendbares, verlustloses Format für Bitmap-Dateien insbesondere für die Verwendung im World Wide Web zu schaffen. Gleichzeitig wurden frei verwendbare Softwarekomponenten publiziert, um es möglichst leicht zu machen, PNG in eigenen Programmen zu verwenden. Aus diesem Grund haben heute die meisten grafikbezogenen Programme sehr gute und kompatible Unterstützung für PNG-Dateien.

PNG unterstützt wie GIF die index-basierte, platzsparende Speicherung, aber auch die vollständige Speicherung aller Farbinformationen (ein Byte für jede Farbe jedes Bildpunktes) und auch einen „Alpha-Kanal“, der für jeden Bildpunkt den Grad der Durchsichtigkeit (Transparenz) angeben kann. Leider ist die Unterstützung gerade dieser Funktion im verbreiteten Microsoft Internet Explorer fehlerhaft, weswegen durchsichtige PNG-Dateien im Internet selten vorkommen.

Innerhalb der Datei werden die Datenblöcke mit Prüfsummen versehen, um Beschädigungen der Datei feststellen zu können.

Animationen sind mit dem auf PNG basierenden MNG-Format möglich.

Portable Network Graphics dürfte für verlustfreie Speicherung heute eines der am besten geeigneten Grafikformate neben TIFF sein. Die Einfachheit und Eindeutigkeit der Spezifikation, dadurch die geringere Komplexität gegenüber TIFF, der Verzicht auf lizenzpflichtige Verfahren und die gute Kompression sind eindeutige Vorteile.

3.5.8.5 JFIF (JPEG File Interchange Format, häufig JPEG genannt) JPEG ist die Abkürzung für „Joint Photographic Experts Group“ und der Name einer verlustbehafteten Methode der Bildspeicherung, die 1990 von der ISO standardisiert wurde.

Der Unterschied zwischen JFIF und JPEG ist, daß JFIF das Dateiformat beschreibt, und JPEG die Reduktionsmethode. Wie erwähnt können TIFF-Dateien auch mit dem JPEG-Verfahren reduzierte Daten enthalten. Das JPEG-Verfahren verlangt (in den meisten Implementierungen) einen Parameter zwischen 1 und 100, der das Ausmaß der Reduktion angibt. 1 ergibt sehr kleine Bilddateien in sehr schlechter Qualität, 100 minimale Reduktion auf Kosten der Dateigröße. In der Praxis werden für Fotos meist Qualitätseinstellungen zwischen 70 und 85 verwendet, da darunter die Qualität schon sichtbar schlechter wird, während darüber nur mehr die Dateigröße ansteigt, ohne daß Menschen noch einen Unterschied sehen können.

Bei Fotos und anderen natürlichen Bildern kann JPEG mit kaum bis gar nicht sichtbaren Verlusten sehr kleine Dateigrößen (ca. 1/20 des unkomprimierten Originals) erreichen. Für Diagramme, Bilder mit wenigen Farben und plötzlichen Farbübergängen (die z. B. auch bei Buchstaben in Beschriftungen auftreten) ist es hingegen weniger geeignet, da an den Rändern der Flächen sichtbare sog. „Artefakte“ entstehen.

Die verpflichtenden Teile des JPEG-Standards werden allgemein als frei von Patentansprüchen angesehen, es haben jedoch einige Firmen Ansprüche auf einzelne Aspekte[44] angemeldet. Die patentrechtliche Situation ist daher nicht zu 100 % klar, aber es wird allgemein angenommen, daß JPEG in dieser Hinsicht für die Gegenwart und Zukunft akzeptabel ist (eventuelle Patente würden auf jeden Fall in wenigen Jahren auslaufen).

Die JPEG2000-Spezifikation[45] ist als Nachfolgerin von JPEG konzipiert. Der Algorithmus enthält andere mathematische Verfahren („Wavelets“), die bei kleinerer Dateigröße eine bessere Bildqualität (auf Kosten der Verarbeitungszeit, aber die ist bei den heutigen Rechenkapazitäten nebensächlich) bieten sollen. Es gibt mehrere Patentansprüche auf den Algorithmus. Frei verfügbare Werkzeuge sind erhältlich, aber wegen der Patent-Situation ist ihr Status unklar. Bilder im JPEG2000-Format sind noch selten.

3.5.8.6 Anwendungsspezifische Formate Komplexe Grafikprogramme können keines der genannten Formate oder irgendeinen anderen Standard verwenden, da sie mehr Informationen speichern müssen als nur die Bilddaten: z. B. die Anordnung von „Bildschichten“ („Layers“), die Bearbeitungsgeschichte, weiterverarbeitbare Textblöcke usw. Deswegen verwendet jedes Grafikprogramm ein eigenes Format. Häufig können sie die Dateien von den verbreiteteren Konkurrenten lesen, aber nur selten schreiben.

Anwendungsspezifische Formate werden selten für den Datenaustausch oder im Internet verwendet, da sie für den Empfänger nur dann sicher verwendbar sind, wenn dieser dieselbe Version derselben Software verwendet. Zudem sind die in anwendungsspezifischen Formaten gespeicherten Bilddateien meist deutlich größer als wenn sie in einem für den Datenaustausch konzipierten Format gespeichert sind.

3.5.8.7 Bildformate in der Praxis Im World Wide Web werden üblicherweise GIF und JPEG verwendet, GIF wird aber zunehmend von PNG abgelöst, da PNG mehr Farben erlaubt und gleichzeitig bessere Kompression bietet. Wegen der Patent-Einschränkungen von GIF fehlte in Open-Source- und gratis verfügbaren Programmen lange Zeit die Unterstützung für komprimiertes GIF.

Dort, wo professionell Bildbearbeitung betrieben wird oder Bilder archiviert werden, kommen verlustbehaftete Formate nicht in Frage. Wegen der Vielseitigkeit und der Reife des Dateiformats wird dabei meist TIFF eingesetzt. TIFF kann mit geeigneten Methoden ähnliche Kompressionsgrade erreichen wie PNG.

[44] Vgl. z. B. heise online: Texanisches Unternehmen will Geld für JPEG http://www.heise.de/newsticker/meldung/29201

[45] JPEG 2000 http://www.jpeg.org/jpeg2000/

Wenn in den Dateien nur Bilddaten vorhanden sind, keine Metadaten, anwendungsspezifische Erweiterungen usw., was für die meisten heute vorhandenen Bilddateien gelten dürfte, ist eine Konvertierung zwischen geeigneten verlustlosen Formaten völlig problemlos. Eine Datei könnte zehnmal hintereinander auch mit unterschiedlicher Software zwischen TIFF und PNG hin und her konvertiert werden, das Endergebnis wäre noch identisch mit dem Ausgangs-Bild (siehe Experiment 7.2.2 auf Seite IV).

Mit TIFF und PNG stehen zwei Formate zur Verfügung, die praktisch alle Anforderungen an verlustlose Bilddatenspeicherung erfüllen. Aus diesem Grund erscheint es unwahrscheinlich, daß kurz- oder mittelfristig ein neues Format diese ablöst, oder Programme nicht mehr mit ihnen umgehen können.

Bei der verlustbehafteten Speicherung wird es höchstwahrscheinlich noch einiges an Entwicklung geben. Wegen der großen Verbreitung von JFIF/JPEG wird die Unterstützung für das Format nicht so schnell verschwinden. Zu beachten ist, daß Bilder, die nur als JFIF/JPEG zur Verfügung stehen, keinesfalls mit einer anderen verlustbehafteten Methode gespeichert werden sollten, da unterschiedliche Reduktionsverfahren auch unterschiedliche Verluste in der Bildinformation bewirken. Falls einmal – zum Beispiel wenn die Unterstützung für JPEG in verbreiteten Programmen eingestellt werden sollte[46] – eine Konvertierung von JPEG-Bildern in ein anderes Format notwendig wird, sollte dafür auf jeden Fall ein verlustloses Format gewählt werden, damit die Qualitätsverschlechterungen sich nicht summieren.

3.5.9 Bildformate für Vektorgrafik

Wenn Bilder nicht als Raster gespeichert, sondern aus Linien, Kurven, Punkten und anderen Objekten zusammengesetzt werden, sprechen wir von *Vektorgrafik*. Solche Bilder haben den Vorteil, daß sie beliebig vergrößerbar sind; schräge und kurvige Linien erscheinen bei jeder Vergrößerung korrekt, während bei der Vergrößerung von Bitmap-Bildern die Punkte, aus denen die Linien zusammengesetzt sind, zu „Treppen“ werden.

Es gibt weniger Vektor- als Bitmap-Formate, aber immer noch recht viele, mit unterschiedlichen Eigenschaften. Einige von ihnen sind binär (z. B. CGM und WPG), andere textbasiert (DXF, EPS), und in letzter Zeit scheint das XML-basierte SVG an Boden zu gewinnen.

Der Übergang zwischen Vektorbildern und Seitenbeschreibungssprachen wie PDF oder PostScript ist fließend, viele der Formate können für die jeweils andere Aufgabe auch eingesetzt werden. Da aber nicht jedes Format dieselben Funktionen unterstützt (z. B. Metadaten, Farbinformationen, Unterstützung für eingebettete Bitmap-Bilder

[46] Wofür jedoch kein Grund besteht. Wer auf die JPEG-Unterstützung in einer Software verzichtet, verringert die Nützlichkeit des eigenen Programms und verliert dadurch fast zwangsläufig Marktanteile.

und Schriftart-Daten usw.), kann durch eine Konversion in ungeeignete Formate Information verlorengehen.

3.5.10 Audio- und Videoformate

Moderne Audio- und Videoformate wie WAV („Wave“, Welle, abgekürzt), AVI (Audio Video Interleaved), Apple Quicktime, Ogg und Matroska sind sogenannte Container-Formate. Die Dateien enthalten Informationen über den Inhalt und einen oder mehrere Datenströme, z. B. für Bild- und Ton-Daten oder zwei Tonspuren für Stereo-Wiedergabe.

Die genannten Formate beschreiben nur den Aufbau der Datei. Wie die Datenströme kodiert sind, bestimmen die sogenannten Codecs (COder/DECoder). Es gibt eine große Anzahl von Codecs für Audio- und Videosignale.

Für die verlustlose Kodierung von Audio wird meist PCM, Pulse Code Modulation verwendet. Zum Beispiel arbeiten Audio-CDs und digitale Videokameras nach diesem Verfahren. Es gibt einige verlustlose Audiocodecs[47], die ungefähr auf die Hälfte der ursprünglichen Datenmenge komprimieren können[48]. Verlustlose Videocodecs sind sehr selten, da bei Video (mindestens 25 Einzelbilder/Sekunde!) sehr große Datenmengen anfallen und ein verlustloses Verfahren immer noch viel zu viele Daten ergäbe.

Verlustbehaftete Audiocodecs sind MP3 (MPEG 1 Layer 3), MP3Pro (Nachfolger von MP3), AAC (Advanced Audio Coding, auch MPEG-4 Audio genannt) und Ogg Vorbis. Alle bis auf Ogg Vorbis sind mit Patenten belegt, wobei diese bisher eher selektiv bis gar nicht durchgesetzt werden, um die Verbreitung des jeweils eigenen Codecs nicht zu gefährden.

Videocodecs verwenden meistens verlustbehaftete Algorithmen, die dem JPEG-Algorithmus ähneln, es gibt sogar einen Motion (Bewegungs-) JPEG-Codec. Heute beliebt sind MPEG (Motion Picture Expert Group) in den Versionen 1, 2 (auf DVDs zu finden) und 4 (unter dem Namen DivX sowie in Verbindung mit Apple Quicktime recht verbreitet), RealVideo (Entwicklung einer einzelnen Firma) und DV (Digital Video, in digitalen Videokameras verwendet). MPEG 2 und 4 sind von Patenten belegt, die sich nach Darstellung der Patentinhaber auch auf alle anderen Verfahren der verlustbehafteten Videocodecs erstrecken. Diese Patente auf Algorithmen und Software spielen jedoch derzeit in Europa keine große Rolle.

Die Spezifikationen der meisten Container-Formate sind publik. Unrühmliche Ausnahmen sind zwei Formate der Firma Microsoft, ASF und WMV; Microsoft hält Patente auf einzelne Aspekte der Dateiformate und bedroht andere, die Unterstützung für

[47] Beispiel: FLAC - Free Lossless Audio Codec http://flac.sourceforge.net/
[48] FLAC - Comparison http://flac.sourceforge.net/comparison.html

die Formate selbst entwickelt haben, mit Gerichtsverfahren (zumindest in den USA[49]). Dies passiert wahrscheinlich, um Konkurrenz möglichst auszuschalten, während das eigene Format durch Bevorzugung in den eigenen Produkten in den Markt gedrückt wird. Dadurch könnte ein Quasi-Monopol bei Multimedia-Dateiformaten entstehen.

Viele gute Codecs (MP3, Ogg Vorbis, MPEG-1/2/4 und noch viele mehr) sind auch öffentlich spezifiziert oder sie wurden von anderen Leuten „entziffert“ (*reverse engineering*) und dokumentiert, teilweise behindern jedoch Patentansprüche oder das Wettbewerbsverhalten einzelner Hersteller[50] ihre freie Verwendung. Einige Hersteller wie Microsoft und Real entwickeln eigene, nicht öffentlich dokumentierte Codecs, die ihren Aussagen nach besser sein sollen als jene mit publizierter Spezifikation. Im Markt ist jedoch ein gewisses Bewußtsein dafür vorhanden, daß durch die Wahl dieser Codecs eine Abhängigkeit vom Hersteller aufgebaut wird.

Audio- und Videoinhalte haben häufig einen finanziellen Wert. Aus diesem Grund ist ihre digitale Verfügbarkeit und somit die Kopierbarkeit den Inhaltsanbietern ein Dorn im Auge. Deswegen bevorzugen Inhaltsanbieter und mit ihnen kooperierende Software-Anbieter (Microsoft, Apple) solche Formate bzw. Erweiterungen, mit denen unauthorisierte Kopien (manchmal) unterbunden werden können. Diese Technologien zur Verhinderung von Kopien können auch die aktuelle und zukünftige Nutzung verhindern (siehe Kap. 5.10.1 auf Seite 107).

3.5.11 Datenbanken

Eine Datenbank ist eine systematische Sammlung von (meistens) vielen gleichartigen Daten. Diese werden in sogenannten *relationalen Datenbanken* ([Scha01, S. 27]) in Tabellenform abgelegt, jede Zeile der Tabelle ist ein Datensatz und jede Spalte enthält gleichartige Daten. In den letzten Jahren haben sich auch *objektrelationale Datenbanken* verbreitet (vgl. etwa [Ston99]), in denen die Daten zwar großteils gleichförmig oder sehr ähnlich gestaltet sind, aber die einzelnen Datensätze unterschiedliche Attribute (Eigenschaften) haben können.

Ein Datenbanksystem (vgl. [Scha01, S. 23ff]) besteht aus der Datenbanksoftware (z. B. PostgreSQL, Oracle, Microsoft Access), den Datenbankdateien, die die Datenbanktabellen sowie weitere notwendige Objekte[51] enthalten, häufig einer Datenbank-

[49] Siehe z. B.: Microsoft patents ASF media file format, stops reverse engineering http://www.advogato.org/article/101.html

[50] Z. B. liefert Microsoft nach wie vor keine Abspielmöglichkeit für standardkonforme MPEG-4-Dateien mit Windows aus; Unterstützung für die eigenen proprietären Codecs, die neuer sind, sehr wohl. Deswegen sind die AnwenderInnen gezwungen, Zusatzsoftware zum Ansehen von MPEG-4/DivX-Filmen zu installieren, was eine Hürde für die Akzeptanz bedeuten kann. Das Gleiche gilt für Ogg Vorbis, obwohl der Codec besser komprimiert als MP3, patentfrei ist und lizenzfreie fertige Softwaremodule für Abspielen und Erstellung von Ogg Vorbis zur Verfügung stehen.

[51] Z. B. Sichten (views), die häufig gebrauchte Datenbankabfragen enthalten; Regeln (rules), die sicher-

zugriffsschicht und einer oder mehreren Datenbank*anwendung(en)*. Datenbankanwendungen sind notwendig, weil es selbst bei kleineren Datenbanken unbequem ist, direkt mit den Rohdaten zu arbeiten oder sie sinnvoll zu verknüpfen.

Größere Datenbanken bestehen normalerweise aus mehreren Tabellen und deren Verknüpfungen. Diese Verknüpfungen werden häufig nur implizit durch die Datenbankanwendung definiert, nicht durch das Datenbanksystem selbst.

Die größten und komplexesten Software-Systeme, zum Beispiel die Auftragsbearbeitungssysteme großer Firmen und Behörden, sind meistens datenbankbasierte Anwendungen. Ein solches System kann aus folgenden Komponenten bestehen: Datenbanksystem(e), Middleware[52], Datenbankzugriffsschichten (mehrere, wenn unterschiedliche Software-Plattformen im Einsatz sind), Administrationswerkzeuge, Anwendungssoftware, Schnittstellen für den automatisierten Datenaustausch mit der Außenwelt sowie Berichtssysteme. Jede der genannten Komponenten kann auch mehrfach vorkommen.

Die verbreiteten Datenbanksysteme speichern die Datenbankobjekte in hochkomplexen, binären Formaten auf den Datenträgern. Die Daten können aber normalerweise vollständig über standardisierte Schnittstellen abgefragt und exportiert werden, wenn die Datenbanksoftware noch vorhanden ist und funktioniert. Heute dominiert dafür die Sprache SQL (Structured Query Language, Strukturierte Abfragesprache), deren aktuell gültige Spezifikation 2003 verabschiedet wurde, die Version davor 1999. Wenigstens ANSI SQL'99 ist in den verbreiteten Datenbankprodukten mehr oder weniger genau implementiert, aber bei Details, die in der Spezifikation nicht enthalten sind, weichen die Systeme voneinander ab[53]. Deswegen ist es, wenn solche meist unvermeidlichen Elemente verwendet werden, nicht leicht, eine Datenbanksoftware durch eine andere zu ersetzen.

3.6 Software

Programme oder Sammlungen von Programmen für die Lösung von Aufgaben gleich welcher Art werden *Software* genannt. Die Bezeichnung kommt daher, daß im Gegen-

stellen sollen, daß nur Daten, die gewissen Anforderungen entsprechen, in die Datenbank reinkommen; Auslöser (trigger), das sind Aktionen, die beim Einfügen oder Löschen von Daten ausgeführt werden; und häufig auch gespeicherte Prozeduren (stored procedures), Programmstücke, die von der Datenbanksoftware ausgeführt werden.

[52] Vermittelnde Schicht zwischen der eigentlichen Software-Anwendung und dem Datenbanksystem. Middleware wird häufig eingesetzt, um abgestufte Nutzungsrechte durchzusetzen, den Zugriff auf die Datenbank zu beschleunigen, zur Erhöhung der Verfügbarkeit durch Nutzung mehrerer Datenbanken, oder um eine gewisse Unabhängigkeit von einer bestimmten Datenbanksoftware zu erreichen.

[53] Ein Beispiel: Einstellen desselben Datumsformats (soweit möglich) in drei Datenbankservern:
Microsoft SQL Server 6.0: `SET DATEFORMAT ymd`
Oracle 8.1: `ALTER SESSION SET NLS_DATE_FORMAT='YYYY-MM-DD HH24:MI:SS'`
PostgreSQL 7.2: `SET DATESTYLE TO 'ISO'`

satz zu den „harten“, unveränderbaren Komponenten des Computers („Hardware“) die Software (mit Einschränkungen) änderbar ist.

Die ersten Computer konnten nur direkt in ihrem Maschinencode, mit binären Anweisungen programmiert werden. Da das sehr mühsam und fehleranfällig ist, wurden sogenannte „höhere Programmiersprachen“ entwickelt, die für Menschen mit entsprechender Ausbildung lesbare Anweisungen und Regeln sowie Kommentare über das beabsichtige Verhalten enthalten.

Diese Programme in höheren Programmiersprachen (als Quellcode bezeichnet) müssen für die Maschine entweder in Binärcode übersetzt (kompiliert) oder von einem geeigneten „Interpreter“-Programm direkt ausgeführt werden. Ein weiterer Ansatz ist die Verwendung sogenannter „virtuellen Maschinen“, das sind genau spezifizierte und als Software implementierte Ablaufumgebungen für Programme, die in Binärcode für die jeweilige virtuelle Maschine übersetzt sind[54]. (Siehe auch Kap. 5.7.8 auf Seite 96.) Die Idee dahinter ist, daß – bei Vorhandensein einer virtuellen Maschine für jede gewünschte Computerplattform – ein und dasselbe Programm auf mehreren Plattformen lauffähig ist.

Binärcode läuft ausschließlich auf dem Prozessortyp, für den er übersetzt wurde. Auf einem anderen Prozessortyp ist ein solches Programm komplett nutzlos. Da es aber für fast alle Sprachen Übersetzerprogramme für fast alle relevanten Computerplattformen gibt, sind die Programme theoretisch auf andere Plattformen übertragbar, wenn der Quellcode zur Verfügung steht. Die erfolgreichsten Open-Source-Programme wie der Webserver Apache, das Bürosoftwarepaket OpenOffice.org oder das Bildbearbeitungsprogramm GIMP etwa laufen auf jedem heute verbreiten Betriebssystem und Prozessortyp. Ähnliches gilt für interpretierte Programme, die nur einen geeigneten Interpreter auf jeder Zielplattform benötigen. Bei nicht-Open-Source-Programmen entscheidet der Hersteller allein, für welche Plattformen und Betriebssysteme er die Software verkaufen will, wodurch solche Software meist für nur ein oder zwei Systeme erhältlich ist.

Leider ist die Situation selbst bei verfügbarem Quellcode nicht problemlos. Ein (nichttriviales) Programm trifft immer Annahmen über die Umgebung, in der es abläuft, und Unterschiede verschiedener Betriebssysteme können durchaus verhindern, daß diese Annahmen zutreffen. Zum Beispiel verwenden Unix-artige Betriebssysteme das Zeichen „/“ zum Trennen von Verzeichnisnamen, Windows hingegen „\“. Die Probleme mit

[54] Beispiele für die drei Kategorien:
Kompilierte Sprachen: (meistens) C, C++, Pascal
Interpretierte Sprachen: Perl, PHP, Python, AppleScript, Unix Shellscripts, DOS/Windows Batchdateien, JavaScript
Für virtuelle Maschinen kompilierte Sprachen: Java, C#, Microsoft Visual BASIC

Zeilenenden und big/little-endian habe ich bereits beschrieben. Unterschiedliche Betriebssysteme bieten für erweiterte Aufgaben (z. B. Verschlüsselung, Bilder anzeigen, Abfrage der Daten der Hardware usw.) komplett unterschiedliche Programmierschnittstellen an. Moderne Betriebssysteme enthalten Zugriffsrechte für Dateien und andere Objekte, ältere, noch verbreitete Versionen von Microsoft Windows (98, ME) jedoch nicht; ein Programm, das davon ausgeht, daß es überall schreiben und lesen kann, läuft daher eventuell auf einem System mit Zugriffsrechten nicht. Solche Programme zu schreiben, die auf andere Systeme übertragbar sind, ist also relativ aufwendig und erfordert gute Kenntnisse über mehrere Betriebssysteme.

Das heute meistverkaufte Betriebssystem Microsoft Windows XP und viele kommerziell erhältliche Software-Produkte verwenden eine Art Kopierschutz, der verhindern soll, daß die Software von einem Computer auf andere kopiert wird. Die verwendeten Verfahren sind sehr unterschiedlich; jedenfalls können sie auch die rechtmäßigen BesitzerInnen daran hindern, die Software auf mehreren Computern *hintereinander* (z. B. wenn der alte kaputt geworden ist) zu verwenden. Neben den technischen Maßnahmen gab es gerade von Microsoft auch schon rechtliche Versuche, eine Software an eine bestimmte Hardware zu binden (vgl. [Dech00]).

4 Detaillierte Beschreibung des Problems

4.1 Physische Lebensdauer der Datenträger

Es ist interessant, die Lebensdauer der früheren Datenträger im Vergleich mit digitalen Speichermedien zu betrachten. Die Erfahrung zeigt nämlich, daß auch scheinbar fragile Informationsträger wie Papier sehr langlebig sein können.

4.1.1 „Alte“ Datenträger

Was wir über die Hochkulturen des Altertums wissen, stammt zu einem großen Teil aus der Auswertung von Datenträgern aus ihrer Zeit. Die Gesetzestafel des Hammurabi ist etwa 3.800 Jahre alt, ähnlich alte Aufzeichnungen gibt es aus Ägypten, China und noch einigen anderen Kulturen.

Diese ältesten heute erhaltenen Datenträger haben gemeinsam, daß sie aus sehr permanentem Material bestehen. Die erwähnte Steintafel kann 3.800 Jahre später nicht nur vollständig gelesen, sondern auch übersetzt und interpretiert werden.

Papyrus wurde in Ägypten schon im 3. Jahrtausend v. Chr. verwendet. Die mit Ruß beschriebenen Papyri sind auch heute noch lesbar, wenn sie die ganze Zeit über trocken gelagert wurden. Die Herstellung war aufwendig: Zwei Schichten der Papyrus-Pflanze wurden in Streifen geschnitten, kreuzweise angeordnet und mit einer Hebelpresse zusammengepreßt (vgl. [Klin59, S. 47]). Bis zum Beginn unserer Zeitrechnung wurde Papyrus in die damals bekannte Welt exportiert.

Deutlich später, vor ca. 3.000 Jahren wurde Pergament[55] erstmals hergestellt und benutzt. Es besteht aus dem geglätteten Fell junger Tiere; d. h. es konnte nur in beschränkter Menge, nach großen Anfangsinvestitionen und mit erheblicher Vorlaufzeit (auf Vorrat, nicht für den gerade aktuellen Bedarf) hergestellt werden (vgl. [Klin59, S. 80]).

4.1.2 Papier

Papier aus Pflanzenfasern war in China schon vor unserer Zeitrechnung bekannt. Da das Herstellungsverfahren geheimgehalten wurde, verbreitete sich das Papier etwa tausend Jahre lang kaum. Über den arabischen Raum gelangte es schießlich im 12. Jahrhundert nach Europa und begann, da deutlich billiger als Pergament, dieses in der Beliebtheit zu

[55] Der Name leitet sich von der Stadt Pergamon ab. Deren Bibliothek stand in Konkurrenz mit der Großen Bibliothek von Alexandria, und brauchte Schreibmaterial in großen Mengen. Als die ägyptischen Papyrus-Exporte wegen der Konkurrenzsituation zeitweise eingestellt wurden, begann die weitere Verbreitung des Pergaments (vgl. [Canf98, S. 57]).

überholen (vgl. ebda, S. 81). Aus dieser Zeit sind auch heute noch lesbare Manuskripte auf Papier bekannt.

Am Ende des 18., Anfang des 19. Jahrhunderts stieg der Bedarf nach Papier an, als Folge wurden neue Verfahren der Papierherstellung aus neuen Rohstoffen wie Holz und Stroh erforscht und gefunden. Ab 1845 war die Papierherstellung endgültig industrialisiert (vgl. ebda, S. 192).

Alte Papierdokumente können aus verschiedenen Gründen vom Verfall bedroht sein.

Viele Tinten, die früher verwendet wurden, enthalten Gallussäure, die den sogenannten „Tintenfraß“ verursacht:

> Ganze Blattbereiche bekommen haarfeine Risse, und auch wenn ein vom Tintenfraß befallenes Blatt auf den ersten Blick immer noch intakt und lesbar scheint, zerkrümelt es bei der ersten leisen Bewegung zu kleinen Flocken, einem Puzzle, das niemand je wieder zusammensetzen kann. Über zwei Drittel der Bach-Handschriften, deren Großteil – achttausend Blatt – sich im Besitz der Staatsbibliothek Berlin befindet, sind auf die Weise geschädigt.
>
> [Zimm01, S. 206]

Mit der Umstellung auf industrielle Papierherstellung konnte zwar die erzeugte Menge stark gesteigert werden, und das Papier wurde billiger, doch auch seine Haltbarkeit nahm ab:

> Sauer wurden die Papiere seit 1850 aus zwei Gründen. Erstens wurde die kostbare Cellulose gestreckt, und zwar mit zerfasertem Nadelholz, Holzschliff, der säurebildendes Lignin enthält, das Harte am Holz. Zweitens wurde der Faserbrei, aus dem das Papier geschöpft wird, unter Zusatz von Alaun oder Aluminiumsulfat mit Harzen geleimt. Dadurch gelangte Schwefelsäure ins Papier.
>
> (ebda)

„Saures“ Papier hält im Normalfall nur 50-80 Jahre, unter optimalen Lagerbedingungen bis zu 200. Es vergilbt, wird spröde, zerbröckelt und schließlich zerfällt es zu Staub.

Das Problem ist zwar seit längerer Zeit bekannt (erste Warnungen: 1823, vgl. [Wä95, S. 105]), doch Lösungen gibt es erst seit ca. 1990. Erstens steht säurefreies Papier zur Verfügung, das kaum teurer in der Herstellung ist als säurehaltiges[56], zweitens können

[56] Zu beachten ist allerdings, daß Recyclingpapier, das aus Umweltschutzgründen gerne verwendet wird, nicht haltbarer ist als säurehaltiges Papier. Für Dokumente, die auch in Jahrzehnten noch lesbar sein sollen, sollte also keinesfalls Recyclingpapier, sondern säurefreies Papier verwendet werden.

Bücher mittlerweile in speziellen Maschinen „entsäuert“ werden. Eine „Anlage steht im Keller der Deutschen Bücherei in Leipzig und kann jährlich etwa 200.000 Bücher entsäuern“ ([Zimm01, S. 209]).

Ein weiteres Problem unabhängig von der Beschaffenheit des Papiers ist die Luftverschmutzung: im Blatt bildet sich schweflige Säure, wenn es Schwefeldioxid aus der Luft aufnimmt ([Bred95, S. 98]).

Bücher, bei denen die Schäden schon zu weit fortgeschritten sind, können nur mehr durch aufwendige Restaurationsmaßnahmen wie Laminieren, Papierspalten und Anfasern (vgl. [Bred95, S. 102]) gerettet werden. Bei solchen Exemplaren kann auch die Digitalisierung eine Rettungsmaßnahme sein, wenn dabei die Spezifika der Sicherung digitaler Informationen beachtet werden.

4.1.3 Andere nicht-technische Datenträger

Vor allem in der grafischen Kunst wurden viele andere Datenträger neben dem Papier verwendet: Textilien, Holz oder Stein und noch viele andere. Manche von ihnen können haltbarer sein als Papier, aber aus verschiedenen Gründen manchmal auch unpraktisch: sie sind meist teurer, ihre Aufbewahrung ist durch den größeren Platzbedarf komplizierter, und etwa eine bemalte Wand ist nicht oder nur schwer transportierbar. Ihre Oberflächeneigenschaften setzen auch häufig die Verwendung eigener Techniken (z. B. größerer Druck) oder spezieller Farben voraus.

Generell kann gesagt werden, daß ein größerer Aufwand (an Mühe oder Kosten) beim Festhalten der Information (z. B. etwas in Stein oder Metall hauen, oder auf schwierig zu beschaffende Materialien zu schreiben) zu einer größeren Lebensdauer führen kann, und je robuster das Material ist, desto mehr widersteht es zwar dem Schreiben, aber später auch der Abnutzung durch feindliche Umweltbedingungen.

4.1.4 Mechanische Datenträger

Beispiele: Lochkarte, Phonograph, Langspielplatte.

Bei mechanischen Datenträgern erfolgt das Lesen durch körperlichen Kontakt des Lesegerätes mit dem Datenträger. Dies führt zu einigen Problemen für die Langzeit-Haltbarkeit.

Beim Phonographen und bei der Langspielplatte fahren spitze Nadeln in den Rillen des Tonträgers. Kleine Unebenheiten in der Rille bewegen die Nadel, daraus wird mit unterschiedlichen Verfahren das Tonsignal gewonnen. Dazu ist ein gewisser Druck notwendig, weil die Nadel auch in die tiefsten Vertiefungen hineingedrückt werden muß.

Leider neigt die spitze Nadel wegen des Drucks bei häufigem Abspielen dazu, Unebenheiten in der Rille auszugleichen, besonders Erhöhungen werden „abgetragen“. Dieses

Problem ist bei alten Phonographen-Rollen besonders ausgeprägt, da sie aus relativ weichem Wachs oder Zelluloid hergestellt wurden. (Die Aufnahme passierte auf dem gleichen Gerät, weswegen es notwendig war, daß die Schreibnadel auf die Oberfläche der Rolle einwirken kann.) Es gibt aufwendige technische Verfahren, um solche Rollen noch berührungsfrei (z. B. mit Hilfe von Laser-Abtastung, siehe [PeHa03]) auszulesen und so die bis zu 100 Jahre alten Aufnahmen zu retten. (Ob die digitalisierten Kopien in 100 Jahren noch nutzbar sind, ist eine andere Frage.)

Schallplatten sind aus widerstandsfähigeren Materialien, und sie wurden auch anders hergestellt, sodaß dieses Problem sich bei ihnen nicht so stark auswirkt. Wenn eine Platte sehr oft abgespielt wurde, kann sie aber durchaus in einem schlechten Zustand sein. (Und es wird immer schwieriger, Abspielgeräte zu finden – siehe auch Kap. 4.2 auf Seite 63.)

Lochkarten sind von den Problemen von Papier betroffen, allerdings wegen der mechanischen Beanspruchung viel stärker als etwa Bücher (vgl. [Wett95, S. 463]). Außerdem brauchen sie enorm viel physischen Platz, da eine Karte nur 80 bis 96 Zeichen speichert.

4.1.5 Fotochemische Datenträger

Einige chemische Substanzen ändern durch Lichteinfall ihre Farbe. Dieses Verhalten wird seit ca. 170 Jahren dazu benutzt, stehende und später auch bewegte Abbildungen (Fotos und Filme) herzustellen.

Die Datenträger bestehen im Normalfall aus mindestens zwei relevanten Komponenten: der ursprünglich fotosensitiven Schicht und dem Trägermaterial.

Bei den fotosensitiven Partikeln bewirkt der Lichteinfall einmal eine schnelle Zustandsänderung: das Bild entsteht; es wird danach mit chemischen Mitteln fixiert. Aber späterer Lichteinfall kann je nach Verfahren weitere Änderungen bewirken, da die Fotosensitivität der verwendeten Materialien beim Entwickeln häufig nicht komplett eliminierbar ist. Daher sollten Fotos möglichst vor Lichteinfall und anderer Strahlung geschützt werden.

Die ältesten heute noch erhaltenen Lichtbilder stammen aus den 40-er-Jahren des 19. Jahrhunderts. Es handelt sich um Daguerrotypien; das sind versilberte Kupferplatten, versiegelt und mit Glas geschützt. Ihre Haltbarkeit wird auf noch mindestens 500 Jahre geschätzt[57].

Heutige Fotos bestehen aus verschiedenen Teilen, die gesondert betrachtet werden müssen. Schwachstelle kann das Papier sein (z. B. saures Papier wie bei Büchern), oder die fotosensitive Schicht. Gute Filme sollen bei geeigneter Lagerung bis zu 300 Jahre

[57] Doris Krumpl: *Diesmal wirklich eine Sensation.* Der Standard, Donnerstag, 8. Mai 2003, Seite 17

halten (vgl. [Smit99a, S. 8]), natürlich gibt es keine so lang zurückreichenden Erfahrungen. Auf jeden Fall sollten Fotos und Negative, ob schwarzweiß oder farbig, kühl und möglichst dunkel gelagert werden (vgl. [Duch88, S. 57]).

Bewegtbilder (Filme) brauchen ein flexibles Trägermaterial, das mit hoher Geschwindigkeit (20-30 Bilder pro Sekunde) bewegt und zur Aufbewahrung aufgerollt werden kann. Früher wurde Zelluloid verwendet; dieses Material ist wegen seiner Brandgefährlichkeit und Neigung zur Selbstentzündung berüchtigt, aber es zersetzt sich auch gern selbst, ohne zu brennen, dafür unter Freisetzung giftiger Gase (vgl. [Webe91, S. 72]).

Einige alte Kinofilme sind bereits verlorengegangen[58] oder stark gefährdet.

Die heute verwendeten Materialien bestehen aus Azetylzellulose oder Polyester, sind weniger brandgefährlich und zersetzen sich nicht so leicht. Wirkliche Langzeiterfahrungen mit ihnen gibt es wegen ihrer relativen Jugend noch nicht, und möglicherweise wird es diese auch nicht mehr geben müssen: in der Filmbranche geht der Trend zu digitalen Datenträgern bei der Produktion und beim Abspielen. Diese neuen Datenträger sind nicht mehr fotochemisch, da fotochemische Datenträger eher nur für analoge Datenaufzeichnungsverfahren geeignet sind.

4.1.6 Magnetische Datenträger

Alle magnetischen Datenträger bestehen mindestens aus dem Trägermaterial und der magnetisierbaren Schicht. Diese Komponenten und ihre Verbindung sind unterschiedlichen für die Langzeitsicherung relevanten Einwirkungen ausgesetzt.

Die magnetisierbare Schicht ist anfällig für andere Magnet- und elektrische Felder. Wird der magnetische Einfluß zu stark, ändert sich die Magnetisierung der Partikel, und die gespeicherten Daten werden verfälscht. Magnetfelder, die Datenträger gefährden können, kommen vielerorts vor:

- Elektronische Geräte wie Fernseher und Lautsprecher arbeiten mit Magnetfeldern oder senden elektrisch geladene Teilchen aus, die auf die Aufzeichnungen einwirken können. Aus diesem Grund wird empfohlen, Video- und Audiokassetten nicht in der Nähe des Fernsehers oder von Lautsprechern aufzubewahren.

- In Kassetten und bei Band-Rollen liegen die einzelnen aufgerollten Schichten des Bandes nahe aneinander. Wenn irgendwo durch ein stärkeres Signal (z. B. eine laute Stelle in der Musik) eine größere Anzahl von magnetischen Teilchen zusammen ein stärkeres Feld hat als die „Widerstandsfähigkeit“ der Partikel in den angrenzenden Bandteilen noch verkraftet, kann das Feld „übersprechen“. Das wirkt

[58] Z. B. vom Fritz-Lang-Klassiker „Metropolis“ aus 1923 ist kein vollständiges Exemplar mehr vorhanden.

sich etwa bei jahrelang aufgewickelt gelagerten Musikkassetten hörbar aus: um laute Stellen herum gibt es ein periodisches Echo („Kopiereffekt“, vgl. [Webe94, S. 405]).

- Andere Einwirkungen von elektromagnetischen Wellen, Hitze, Strahlung (z. B. kosmischer) und sogar des Magnetfeldes der Erde sind auch denkbar. Je dichter die Daten gepackt werden, desto weniger Partikel stellen eine Informationseinheit dar: damit nimmt die Anfälligkeit für zufällige Schäden zu, da der magnetische Zustand sich leichter ändert.

Eine schwache Entmagnetisierung kann mit der Zeit auch ohne äußere Gründe auftreten (vgl. [Schn96]).

Es ist notwendig, zwischen digitalen und analogen magnetischen Datenträgern zu unterscheiden. Analoge Information läßt sich nur verlustbehaftet kopieren und sollte daher möglichst selten, d. h. möglichst spät, aber noch vor dem Unbrauchbarwerden des Datenträgers umkopiert werden (es ist natürlich schwer, den idealen Zeitpunkt zu erwischen). Im Gegensatz dazu kann digitale Information jederzeit umkopiert werden.

In der professionellen Informatik ist es üblich, Magnetbänder mit wichtigen Informationen alle drei bis fünf Jahre umzukopieren (vgl. [Wett95, S. 465]), damit ihre Magnetisierung „aufgefrischt“ wird. Die Herstellerempfehlungen geben 10 bis 15 Jahre als Obergrenze der Haltbarkeit an (vgl. [Schn96]). Disketten sollten sogar alle 4-5 Jahre umkopiert werden (vgl. [Wett95, S. 463]).

Die Trägermaterialien werden ständig weiterentwickelt, weil sie noch lange nicht perfekt sind. Alte Bänder sind bereits brüchig und verletzlich, ihr Material (meist eine Kunststoff-Folie, früher auch PVC oder Acetatzellulose) altert teilweise schon innerhalb eines Jahrzehnts, besonders bei Beanspruchung (Zug beim Bewegen des Bandes).

Ein weiteres Problem kann die Befestigung der magnetisierbaren Schicht auf dem Kunststoffband darstellen. Die dafür verwendeten Klebstoffe oder andere Verfahren sind unterschiedlich haltbar; insbesondere in tropischen Gebieten oder solchen mit hoher Luftfeuchtigkeit sowie bei Bändern schlechter Qualität ist es schon passiert, daß der Klebstoff nachgegeben hat oder ausgetreten ist, eventuell die Abspielgeräte beschädigte, und die magnetischen Partikel vom Band „gewischt“ wurden (vgl. [Wett97, S. 738]). Diese Daten sind natürlich komplett verloren.

Die magnetische Speicherung erlaubt eine sehr große Datendichte: die einzelnen Partikel(gruppen), die eine Informationseinheit enthalten, sind sehr klein und nah beieinander. Deswegen muß sich der Schreib-Lese-Kopf sehr nahe (z. B. bei Festplatten ca. 10-15 Nanometer (vgl. [Brem03, S. 138])) an der Datenträgeroberfläche bewegen. Auch minimale Störungen wie ein Fingerabdruck oder ein Staubkorn vergrößern den Abstand

zwischen dem Kopf und der Oberfläche so stark, daß die Daten nicht mehr lesbar sind. Dies dürfte einer der wichtigsten Gründe für Lesefehler von Computer-Disketten sein.

Wenn etwa durch eine Erschütterung der Lesekopf zu nahe an die sich teilweise sehr schnell bewegende Platte[59] kommt, kann er die Oberfläche beschädigen, also die magnetisierbare Schicht herunterkratzen. Dieser gefürchtete Effekt heißt *head-crash*; an der Stelle des Kratzers kann danach weder gelesen noch geschrieben werden (vgl. [Vö96, S. 246]). Und während ein einige Millimeter großer Kratzer auf einem analogen Datenträger (z. B. Audio- oder Videokassette) höchstens eine kleine Störung im Signal verursacht (z. B. eine hörbare Unterbrechung oder Flackern des Bildes), speichert eine moderne Festplatte auf derselben Fläche mehrere Kilobyte oder sogar Megabyte Daten.

Die Ausfallssicherheit von Festplatten läßt sich mit einigen Arten der RAID-Technologie (Redundant Array of Inexpensive Discs) erhöhen, indem z. B. fünf Festplatten zu einer logischen Einheit zusammengefaßt werden, auf die dann die Hardware oder Software die Daten mehrfach speichert. Fällt eine der Festplatten aus, kann sie in der Regel im Betrieb ausgetauscht werden, die darauf enthaltenen Daten werden automatisch aus den Kopien auf den anderen Festplatten wiederhergestellt. RAID-Systeme sind für die kurzfristige Ausfallssicherheit sehr nützlich, aber nicht für die langfristige Speicherung gedacht, da sie ständige oder zumindest regelmäßige Überwachung und gelegentliche Wartung erfordern, und weil sich die Festplattentechnologie zu oft ändert, sodaß die Versorgung mit Ersatz-Festplatten nach spätestens 8-10 Jahren unsicher erscheint.

4.1.7 Magneto-optische Datenträger

Die Technologie ist nur ca. 15 Jahre alt. Es gibt noch keine Langzeit-Erfahrungen mit der Haltbarkeit der Datenträger, aber es wird angenommen, daß wegen der stabileren Magnetisierung und des berührungslosen Ableseverfahrens die Datenträger die Daten länger behalten sollten als magnetische Datenträger, es wird von 30 bis 50 Jahren Haltbarkeit ausgegangen ([Vö96, S. 290]); so lange dürfte jedoch kein Lesegerät halten. Anders als die standardisierten optischen CD-R und DVD-R-Formate hat sich kein magneto-optisches Speicherverfahren im Markt durchsetzen können; deswegen gibt es eine Menge unterschiedlicher Systeme, deren zukünftige Weiterentwicklung und Pflege unkalkulierbar erscheinen. Wenn ein Hersteller die Produktlinie aufgibt, sind keine neuen Lesegeräte oder Ersatzteile für die alten Lesegeräte mehr erhältlich.

4.1.8 Optische Datenträger

Da es CDs erst seit ca. 20 Jahren überhaupt gibt, sind wirkliche Langzeiterfahrungen mit ihnen noch nicht vorhanden. Es gibt einige Verfahren, um beschleunigte Alterung

[59] In schnelleren Festplatten sind bis zu 10.000 Umdrehungen/Minute üblich.

zu simulieren, zum Beispiel mit Hilfe von Hitze, erhöhter Luftfeuchtigkeit oder starken Lichtquellen (vgl. [StBe97]). Es muß sich aber noch herausstellen, ob diese Tests auf die Wirklichkeit übertragbar sind.

Optische Datenträger müssen schonend behandelt und unter günstigen Bedingungen gelagert werden, um lang zu leben. Einige der Gefahrenquellen, die sie bedrohen können:

- Kratzer auf der „unteren“, Daten tragenden Seite: Sie können den Laserstrahl ablenken und so zu Lesefehlern führen (vgl. [Arps93, S. 97]). Diese Schäden sind manchmal mit sehr feinem Schmirgelpapier korrigierbar.

- Mechanische Beschädigungen auf der Beschriftungsseite: Da die reflektierende Schicht auf dieser Seite angebracht ist und nur von einer dünnen Lackschicht geschützt wird (das gilt nur für CDs; DVDs haben die reflektierende Schicht in der Mitte (vgl. [Gies04, S. 135])), wird diese Seite als die empfindlichere angesehen. Mechanische Beschädigungen und Kratzer (es reicht schon, mit einem Bleistift oder Kugelschreiber darauf zu schreiben) beschädigen leicht die Lackschicht, die dort nicht mehr den Laserstrahl reflektiert, wodurch die Daten nicht mehr lesbar sind.

- Verformung: Es ist wichtig, daß die Scheiben genau im rechten Winkel zum Laserstrahl rotieren, da sie das Licht sonst nicht in die richtige Richtung reflektieren. Sie dürfen sich daher nicht verbiegen, sonst werden sie unlesbar. Deswegen wird empfohlen, CDs und DVDs aufrecht zu lagern, nicht flach, weil das Eigengewicht der Scheiben langfristig zu einer Biegung nach unten führt. Aufgeklebte Etiketten schädigen CDs nicht sehr, DVDs jedoch in einem gefährlichen Ausmaß (siehe [Gies04, S. 136]; dort wird das Ergebnis eines Tests mit DVDs als „katastrophal“ bezeichnet).

- Chemische Beschädigungen auf der Beschriftungsseite: Ungeeignete Filzstifte o. Ä. enthalten aggressive Lösungsmittel, die die reflektierende Schicht angreifen und ihre Reflexionseigenschaften ändern können.

- Anscheinend gibt es bei extrem tropischem Klima sogar Mikroorganismen, die sich von verschiedenen Schichten der Datenträger ernähren können. Es gibt Berichte von CDs, die auf diese Weise beschädigt wurden (vgl. z. B. [Gar$^+$01]).

- Zu viel Licht und eine hohe Luftfeuchtigkeit sowie häufige Änderungen der Luftfeuchtigkeit und der Temperatur können unterschiedlich auf die verschiedenen Komponenten der Datenträger einwirken und langsam dazu führen, daß die Bedingungen für eine vollständige Lesbarkeit der Daten nicht mehr erfüllt sind.

Für gepreßte CDs werden in der Literatur bei richtiger Lagerung über 100 Jahre Lebensdauer prognostiziert, für die einmal beschreibbare CD-R 10-20 Jahre (es gibt allerdings Berichte über unlesbare CD-Rs innerhalb von nur zwei Jahren, z. B. in [StBe97, S. 244] oder im Internet[60]). Einzelne Hersteller garantieren (bei vorgeschriebenen Lagerungsbedingungen) jedoch 25 oder sogar 100 Jahre Haltbarkeit ihrer (teureren) Medien.

Da DVDs und spätere Formate mit ähnlichen Verfahren arbeiten, aber auf derselben Fläche wesentlich mehr Daten speichern, steigt mit jeder Ungenauigkeit und Beschädigung die Wahrscheinlichkeit, daß die Daten nicht mehr lesbar sind, im Vergleich zur CD stark an. Aus diesem Grund müssen wir davon ausgehen, daß DVDs (ob selbst beschrieben oder industriell gefertigt) nicht so lange halten wie CDs derselben Kategorie.

Das in wiederbeschreibbaren Medien für die Datenträgerschicht verwendete Material ist weniger stabil als jenes in einmal beschreibbaren Medien (vgl. [Byer03, S. 15]). Das bedeutet, daß CD-RW und DVD+/-RW für die langfristige Speicherung weniger empfehlenswert sind als CD-R und DVD+/-R (auch weil sie absichtlich oder unabsichtlich überschrieben oder gelöscht werden könnten – außerdem sind sie teurer).

Wie wir noch sehen werden, ist die Lebensdauer der Medien wegen des Einflusses anderer Faktoren (Dateiformate, Software) ab einer bestimmten Grenze (wahrscheinlich ca. 10-15 Jahre) nicht mehr relevant, da die Daten zwar noch lesbar, aber nicht mehr wirklich nützlich sind. Aus diesem Grund dürften sich optische Datenträger für heutige Bedürfnisse bei richtiger Lagerung (konstante Temperatur bei ca. 20 °C, niedrige Luftfeuchtigkeit, lichtgeschützt, aufrecht, keine Aufkleber, von mechanischer Beschädigung beider Seiten geschützt) für die Langzeitarchivierung der reinen Datenströme eignen, da sie nicht das schwächste Glied der Kette darstellen. Es sollten gute Medien verwendet werden, und wichtige Daten sollten nicht nur auf einem, sondern mindestens zwei Datenträgern (am besten von verschiedenen Herstellern) gesichert werden.

4.1.9 Flash-Datenträger

Es gibt noch keine Langzeiterfahrungen mit Flash-Speichern, da die Technik sehr neu ist. Da weder die Datenträger noch die Laufwerke bewegliche Teile enthalten, und die Datenträger deswegen recht stabile Gehäusen haben können, ist die Gefahr mechanischer Beschädigungen eher gering[61]. Die Karten können ohne Probleme mehrere Meter runterfallen, arbeiten im gesamten für Menschen erträglichen Temperaturbereich, und

[60] CD-Recordable discs unreadable in less than two years http://www.cdfreaks.com/news/7751
Slashdot: Say Goodbye To Your CD-Rs In Two Years? http://it.slashdot.org/article.pl?sid=03/08/24/1253248&tid=198&tid=137&tid=126

[61] In einem nicht ganz wissenschaftlichen Test von *Digital Camera Shopper* wurden verschiedene Speicherkarten (CompactFlash, Secure Digital, xD, Memory Stick und SmartMedia) unter anderem in Cola getaucht, in die Waschmaschine gegeben und gewaschen und mit einem Skateboard überfahren. Alle Karten überlebten diese Tests, die enthaltenen Daten waren noch lesbar.

da sie nicht optisch oder magnetisch funktionieren, haben Lichtstrahlen und Magnetfelder keine Wirkung auf sie. Ihre winzige Größe (Secure Digital-Karten sind etwas kleiner als eine Zwei-Euro-Münze) führt aber zu einer neuen Gefährdung: sie können im Vergleich mit anderen Datenträgern recht leicht verlorengehen.

Die Hersteller sind zuversichtlich, daß die Daten in Flash-Datenträgern langfristig sicher sind und geben gerne 7- bis 10-Jahres-Garantien oder sogar Garantie für die Lebenszeit der Datenträger ab. Diese Dauer könnte die Lebensdauer der Lesegeräte um einiges übersteigen, da der Markt sich derzeit rasant entwickelt; es ist also wahrscheinlich, daß nicht die Lebensdauer der Datenträger der limitierende Faktor sein wird.

4.2 Lebensdauer der Abspielgeräte

Als Abspielgeräte bezeichne ich jene Hardware-Komponenten, die einen Datenträger aufnehmen, um die auf diesem gespeicherten Informationen über eine Schnittstelle an den Computer zu vermitteln. Beispiele sind Diskettenlaufwerke, Magnetband-Laufwerke oder auch Lesegeräte für Flash-Datenträger.

Bis in die 1980-er-Jahre hinein waren die Schnittstellen zwischen Computer und Abspielgerät nicht standardisiert; die Geräte mußten an die Schnittstellen der einzelnen Computerhersteller angepaßt werden. Später haben sich Standards wie SCSI und USB durchgesetzt, die nicht an einzelne Plattformen gebunden sind, sodaß heute die meisten externen Abspielgeräte mit praktisch allen relevanten Computertypen betreibbar sind. Es gibt eine Motivation für die Hersteller, im Schnittstellenbereich mit den etablierten Technologien zu arbeiten: so ist ihr potenzieller Markt am größten, und sie müssen nicht in die Entwicklung der Schnittstellen und der notwendigen Schnittstellen-Treibersoftware investieren.

Ein wichtiger Trend in der Speichertechnologie, ähnlich wie im Rest der Computerindustrie, geht zur Miniaturisierung. Früher hieß das, daß die Datenträger immer kleiner wurden (etwa die Disketten und Magnetband-Kassetten), aber heute sind sie zumindest im Endkunden-Bereich schon so klein, daß eine weitere Verkleinerung auf Kosten der Handhabbarkeit ginge. Gleichzeitig steigen die pro Raum- oder Oberflächen-Einheit gespeicherten Informationsmengen, weswegen alte Abspielgeräte mit den neueren Datenträgern meistens nichts anfangen können, selbst wenn ihr Formfaktor identisch ist. Aber gerade um Verwechslung, Fehlbedienung und eventuell daraus resultierende Schäden an den Geräten zu vermeiden, werden auch manchmal die Formfaktoren geändert.

Abspielgeräte mit beweglichen Teilen (bei mechanischen, magnetischen, magneto-optischen und optischen Medien sind das prinzipbedingt alle) haben nur eine beschränk-

BBC News: Technology: Digital Memories survive extremes http://news.bbc.co.uk/2/hi/technology/3939333.stm

te Lebensdauer, die eher in Jahren als in Jahrzehnten zu messen ist. Sie müssen unter schwierigen Bedingungen (Staub, Vibrationen, Temperaturänderungen etc.) extrem präzise arbeiten; wenn diese Präzision etwa durch zu viel Staub, die Abnutzung einzelner, viel beanspruchter Teile oder aus anderen Gründen unter die spezifizierten Grenzen fällt, ist das Gerät nicht mehr verwendbar, und kann während der Leseversuche sogar die Datenträger beschädigen.

Einige „Abspielgeräte“, vor allem Festplatten, sind fix mit den Datenträgern verbunden. Hier sind die eigentlichen Datenträger (die Scheiben mit magnetisierbaren Partikeln) in einem staubgeschützen Gehäuse mit der kompletten Elektronik und den beweglichen Leseköpfen eingebaut. Dadurch kann die Kapazität im Vergleich zu separaten Datenträgern sehr stark gesteigert werden, da eine Menge Probleme (vor allem Staub) wegfallen. Aber da die Festplatten auch an den Grenzen des technisch Machbaren betrieben werden, führt dieser Schutz nicht zu einer längeren Lebensdauer. Die meisten Festplatten-Hersteller geben nur eine dreijährige Garantie auf ihre Produkte, das bedeutet, daß sie selbst die durchschnittliche Lebensdauer auf nicht viel mehr als drei Jahre schätzen.

4.2.1 Die Interessen der Computer-Industrie

Bei der Computer-Industrie handelt es sich um eine sehr dynamische. Ein Produkt ist unter Umständen weniger als ein Jahr lang am Markt, dann gilt es als überholt. Ca. alle anderthalb Jahre verdoppeln sich die um den selben Preis erhältliche Rechenleistung und Speicherkapazität[62]. Das ist in mancherlei Hinsicht eine gute Sache, aber es hat auch einige negative Konsequenzen.

Für die Archivierung und Langzeitverfügbarkeit sind Verarbeitungsgeschwindigkeit und insbesondere Kapazität und Preise der Speichertechnologien natürlich auch relevant. Viel wichtiger ist jedoch die Kompatibilität zu älteren Systemen, um auf alte Daten zugreifen zu können („Abwärtskompatibilität“).

Für die Hersteller von neuen Geräten ist das weniger wichtig; sie müssen nur zu den aktuell am Markt befindlichen Schnittstellen kompatibel sein. Einige Hersteller mit genügend Marktmacht in ihrem Segment können sich sogar den Bruch mit ihrer Ansicht nach veralteten Technologien erlauben, so liefert z. B. die Firma Apple ihre Computer seit Jahren standardmäßig ohne Diskettenlaufwerk aus (ein Laufwerk kann natürlich bei Bedarf extra gekauft und angeschlossen werden).

Insbesondere bei ganz neuen Produktkategorien (z. B. Funknetzwerke oder Digitalkameras) fehlen zuerst Industrie- oder andere Standards. Die Firmen, die sich als erste auf die neuen Felder wagen, produzieren ihre Geräte komplett nach eigenen Vorstellun-

[62] Das wird als Moores Gesetz bezeichnet.

gen und sorgen (z. B. mit Treiberprogrammen) selbst dafür, daß die Geräte benutzbar sind. Diese Geräte funktionieren dann z. B. nur mit denen des selben Herstellers (die vor zehn Jahren erhältlichen Funknetzwerk-Lösungen waren so, weil es keinen Standard gab), oder nur mit dem vom Hersteller gelieferten Treiber. Das hat zur Folge, daß die Geräte nur unter den ursprünglichen Bedingungen (Hardware, Betriebssystemversion, eventuell andere Voraussetzungen) funktionieren, sie können nur mit freiwilliger Unterstützung des Herstellers (z. B. in Form von neuen Treibern) unter geänderten Bedingungen (wenn z. B. eine neue Version des Betriebssystems installiert wird) wieder benutzt werden. Bei später auf den Markt kommenden Produkten derselben Kategorie ist dieses Problem meist weniger ausgeprägt, da durch die Standards die Marktdurchdringung höher ist, was z. B. positive Auswirkungen auf die Treiberunterstützung in den Betriebssystemen hat, und bei Bedarf können die Geräte auch leichter durch andere ersetzt werden.

In solchen Bereichen, die nicht viel mit der Infrastruktur zusammenspielen müssen, besteht kaum ein Zwang, zum Umfeld kompatibel zu sein. Hierzu gehört mit kleinen Einschränkungen auch die Speichertechnologie; es gibt zwar einen Bedarf an Geräten, die die bestehenden Datenträger lesen können, aber diese werden am Markt nur so lange angeboten, wie sie zu für den Hersteller akzeptablen Preisen nachgefragt werden. Das hängt wiederum von der Verfügbarkeit besserer Nachfolgetechnologien und auch der Lebenszeit der Medien ab. Deswegen sind z. B. schon seit Jahren keine 5,1/4-Zoll-Diskettenlaufwerke im normalen Computerfachhandel erhältlich: die Laufwerke für 3,5-Zoll-Disketten haben sich schon vor so langer Zeit durchgesetzt, und sind in so vielen Punkten überlegen[63], daß die alten Laufwerke seit geraumer Zeit kaum verkauft werden konnten. (Wer jetzt noch ein Laufwerk kaufen würde, müßte wahrscheinlich auch feststellen, daß die Lesbarkeit der Disketten schon stark abgenommen hat.)

Im Bereich der Bandlaufwerke, die wegen ihrer Kapazität und Geschwindigkeit im Vergleich mit scheibenförmigen Wechseldatenträgern seit fast 50 Jahren einen Platz in der professionellen Speichertechnik haben, geht die Entwicklung ähnlich schnell voran wie beim Rest der EDV. Hier hat sich allerdings wegen des mehr punktuellen Einsatzes[64] kein so weitgehend akzeptierter und „langlebiger“ Standard wie bei der 3,5-

[63] Im Gegensatz etwa zur Netzwerktechnologie. Dort hat die 10-Mbit-Technologie den immer noch relevanten Vorteil gegenüber den Nachfolgetechnologien, daß mit Hilfe eines einzigen Kabels und ohne Zusatzgeräte mehrere Rechner verbunden werden können.

[64] Bandlaufwerke sind wegen ihrer großen Kapazität, aber schlechter Zugriffszeit für die Archivierung großer Datenbestände prädestiniert. Diese großen Datenbestände werden seltener mit anderen ausgetauscht als etwa Disketten im Privatbereich, da die Inhalte teilweise geheim (z. B. Firmen-Aufzeichnungen), häufig sehr wertvoll und meist nur im EDV-Kontext der jeweiligen Organisation verwendbar sind. Archive geben auch sehr ungern ihre „Originale“ (einzigen Exemplare) aus. Wenn ein Datenaustausch gewünscht wird, findet das über andere Medien (z. B. CD-ROM) oder ein gemeinsam festgelegtes Bandformat statt.

Zoll-Diskette herausbilden können, deswegen gibt es einen viel größeren Wildwuchs an Formaten. Ein auf Magnetband-Konvertierung spezialisiertes Unternehmen listet auf seiner Webseite[65] 43 unterstützte Bandformate auf, und das sind sehr wahrscheinlich noch nicht alle, die jemals verwendet wurden.

Es ist wichtig, festzuhalten, daß es sich nicht um eine „Verschwörung der Hersteller“ handelt, oder daß diese bewußt gegen das öffentliche Interesse arbeiten. Sie versuchen ihre Gewinne zu maximieren, und das geht im dynamischen EDV-Massenmarkt nur, indem sie sich auf die profitabelsten Technologien konzentrieren, und das sind meistens die neuesten Geräte. Die externen Effekte dieser Geschäftsentscheidungen z. B. in Bezug auf die Langzeitverfügbarkeit lassen sich schlecht in Geld ausdrücken, und selbst wenn das möglich wäre, wäre nur ein kleiner Teil der Kundschaft bereit, einen Aufpreis für den Bonus der Langzeitverfügbarkeit zu zahlen.

Es gibt sogar nicht-technische Kriterien, die dazu führen können, daß Technologien schneller als nötig veralten und durch andere ersetzt werden. Ein Hersteller, der in einem Bereich besondere Vorteile (etwa Patente oder ein de-facto-Monopol am Markt) hat, wird danach trachten, die Technologien, die diese Vorteile ausnützen, in den Markt zu drücken, notfalls auch mit Hilfe sehr knapp oder sogar mit Verlust kalkulierter Preise, um die Konkurrenz auszuschalten und den eigenen „Standard“ durchzusetzen.

> Es hat sich im Verlaufe der Diskussion die Vorstellung herausgebildet, daß eine Technikentwickung wie die der Informations- und Kommunikationstechnik von bestimmten Faktoren abhängt und daß sie Auswirkungen und Folgen auf bestimmten Feldern zeigen wird. Faktoren und Auswirkungen sind wohl bei jeder Teiltechnik verschieden. Wir nennen der Einfachheit halber einen Satz von Variablen, die als beeinflussend angesehen werden können, einen Faktor, wenn diese Variablen inhaltlich zusammengehören.
>
> Eine Reihe von Faktoren wird aus Beeinflussungsgrößen bestehen, die das Verwertungsinteresse an einer bestimmten Technik abbilden. Hier geht es um die Interessen von Herstellern sowohl der Endgeräte, der Komponenten, der großen Anlagen und so fort, aber auch um die Verwertung im Sinne von Marktanteilen, Handelsspannen, Konkurrenz, Monopolbildung, Marktsättigung u.a.m. Die Beeinflussung der technischen Entwicklung durch die Vorstellungen der Hersteller dürfte wohl primär sein.
>
> [Korn93, S. 33]

In einem solchen Markt sind einzelne Käufergruppen mit speziellen Interessen (etwa Archive mit Bedarf an Systemen, die alte Datenbestände auch lesen können) stark

[65] tapeconversions.com Supported Media http://tapeconversions.com/supported_media.html

benachteiligt. Die etablierten Anbieter haben ein Interesse daran, erstens neue Mitbewerber am Markteintritt zu hindern, und zweitens die Käufer möglichst an ihre eigenen Produkte zu binden:

> Der Gedanke, daß die schlechthin technische Kommunikation das noch mit ihr Kommunizierbare in Richtung auf Betreiberinteressen hin instrumentalisiere, ist eine Anwendung eines aus der Technikphilosophie und ihrer Kontroversen bekannten Arguments, wonach bei gleichen ökonomischen Bedingungen Entscheidungen über technische Weiterentwicklungen immer in der Richtung gefällt würden, die dem Betreiber der Technik eine bessere Kontrolle über den Benutzer erlaubten.
>
> [Korn93, S. 60]

Ein weiteres Problem ist, daß Firmen manchmal in Konkurs gehen und dadurch das Wissen über ihre Produkte, zusammen mit der technischen Unterstützung, Ersatzteilen und neuen Gerätetreibern, verschwindet. Ohne Zugriff auf die Original-Pläne der Geräte und alle weiteren notwendigen Informationen (etwa: Quellcode der Treiberprogramme und der eventuell ins Gerät selbst einprogrammierten Anweisungen) ist es heute praktisch unmöglich, ein modernes Gerät nachzubauen oder auch nur instandzuhalten.

Die CD (-ROM) und ihre Nachfolgetechnologien (z. B. DVD) scheinen bei den Abspielgeräten eine Sonderstellung zu besitzen. Wegen der enormen, mit keinem anderen Datenträger vergleichbaren Verbreitung sind hier die Hersteller stark daran interessiert, daß ihre Geräte abwärtskompatibel sind, d. h. auch die alten Datenträger lesen können. Dadurch scheinen aus heutiger Sicht die CD- und DVD-basierten Datenträger zumindest für die mittelfristige Speicherung geeigneter als andere Datenträgerarten.

4.2.2 Verbesserung der Situation durch das Internet

Die Lebensdauer der Datenträger und der Abspielgeräte war früher ein viel größeres Problem als heute, erstens weil der Markt viel unübersichtlicher, fragmentierter und weniger auf Standards basierend war als heute, und zweitens weil wir heute (seit ungefähr 10 Jahren) eine von praktisch allen Computern und computerartigen Geräten unterstützte Technologie haben: das Internet.

Während früher der Datenaustausch zwischen verschiedenen Computerplattformen (z. B. IBM Mainframe, DOS-PC, Apple, Unix-Workstation) schwierig und problematisch war, beherrschen heute alle relevanten aktuellen Systeme die im Internet verwendeten Protokolle wie IP (Internet Protocol), TCP/IP (Transmission Control Protocol/Internet Protocol) und FTP (File Transfer Protocol). Diese Technologie hat es

sogar geschafft, frühere fast-Monopole wie von Novell und Microsoft im Netzwerkbereich abzulösen, die früheren Marktführer mußten auf Druck des Marktes ihre Produkte so umstellen, daß sie jetzt auch auf Internet-Technologien basieren.

Die Internet-Unterstützung der heutigen Computer ist eine wichtige Funktion, deswegen sind die entsprechenden Programmteile gut dokumentiert und zuverlässig. Das bedeutet, daß heute praktisch jede moderne Computerplattform mit praktisch jeder anderen in einer standardisierten, für die AnwenderInnen gewohnten Art Dateien und andere Daten austauschen kann. Deswegen ist es heute viel leichter, die gefährdeten Daten von einem alternden, aber noch funktionierenden Computer zu einem der neuesten Generation zu kopieren.

Soweit also Daten auf einer heute üblichen Computerplattform zur Verfügung stehen, ist es kein Problem mehr, ihre Bits und Bytes jeweils weiter zu übertragen. Die fehlerfreie Übertragung wird bereits durch die verwendeten Internet-Protokolle gewährleistet, und es ist auch leicht, nachträglich zu überprüfen, ob die Kopien identisch sind, zum Beispiel mit Hilfe von Prüfsummen[66] (vgl. [Rana91, S. 97]).

Ein Problem können solche Computerplattformen bedeuten, die „zusammengesetzte“ Dateien[67] speichern. Sie bieten aber in jeden Fall die Möglichkeit, diese zusammengesetzten Dateien in „flache“ Dateien zu exportieren, um diese dann problemlos übertragen zu können, wodurch das Problem in eines mit Dateiformaten umgewandelt wird.

[66] Sogenannte Prüfsummen können für jede Datei oder beliebige andere Byte-Folge gebildet werden. Sie haben die Eigenschaft, daß schon eine kleine Änderung im Eingabe-Datenstrom zu großen Änderungen der Prüfsumme führt. Auf diese Weise können auch geringfügige Änderungen (aber natürlich auch große) leicht entdeckt werden.

Ein Beispiel: das „md5sum“-Programm bildet die Prüfsumme nach dem weitverbreiteten MD5-Algorithmus von zwei Texten:

Text	MD5-Prüfsumme
Langzeitverfügbarkeit	83ce0d9a50d1c28c26fbc4a5c8b35df4
LangzeitVerfügbarkeit	9889d9eb7a1c9cf0411f78592f69aed9

Die Prüfsummen unterscheiden sich stark, obwohl im zweiten Text nur das „v“ groß geschrieben wurde.

Eine weitere Eigenschaft von guten Prüfsummen-Algorithmen (z. B. MD5 oder SHA) ist, daß es sehr schwierig ist, die *Ausgangsdaten* so zu konzipieren, daß eine gewisse Prüfsumme entsteht. Solche sog. *kryptographisch sicheren* Prüfsummen können daher verwendet werden, die Echtheit eines Datenstroms nachzuweisen (natürlich muß dazu die Prüfsumme auf eine gesicherte Weise gebildet und übertragen worden sein). Dies kann für digitale Archive wichtig sein, um Manipulationen an Daten und Dateien zu erkennen.

[67] Z. B. sogenannte „Streams“ bei Microsoft Windows NT, 2000 und XP, oder „Resource Forks“ bei Apple MacOS.

4.3 Lebensdauer der Dateisysteme

Es gibt selten Probleme mit Dateisystemen, da sie in der Regel Teil eines Betriebssystems und deswegen meist gut dokumentiert sind. Sie sind auch selten besonders komplex. Da sie nur in Verbindung mit einem Betriebssystem Sinn machen, ist ihre Anzahl – im Gegensatz zu Dateiformaten – in der Praxis begrenzt.

Unbekannte Dateisysteme kommen insbesondere auf Datenträgern vor, die von längst veralteten Computerplattformen erstellt wurden. Bei der Rettung der Daten von solchen Systemen ist selten das Dateisystem das einzige oder das größte Problem; häufiger sind Probleme mit dem abweichenden technischen Format der Datenträger sowie dem Fehlen von Schnittstellen zu anderen Plattformen und insbesondere den Dateiformaten.

Es ist durchaus vorstellbar, daß noch Datenträger auftauchen, die die einzigen Kopien von wichtigen Informationen enthalten, und deren Dateisystem unbekannt oder beschädigt ist. In diesem Fall können die Daten mit mehr oder weniger aufwendigen Methoden[68] manchmal gerettet werden.

Für die Zukunft ist das Problem weniger bedeutend. Dadurch, daß alle mittel- oder weitverbreiteten Dateisysteme von Open-Source-Betriebssystemen wie Linux gelesen werden können, steht nicht nur die Dokumentation des Dateisystems, sondern auch gleich ein fertiger Mechanismus zum Lesen der Daten zur Verfügung.

4.4 Lebensdauer der Dateiformate

Wie wir gesehen haben, sind Datenträger und die auf ihnen gespeicherten Daten auch von Überalterung und Defekten bedroht. Im Falle von Daten, die in den letzten 5-8 Jahren entstanden sind, ist dies sehr selten ein Problem, und bei älteren Daten meistens auch nicht das größte. Heute können wir davon ausgehen, daß wir die Bitfolgen von Dateien, die auf einem funktionierenden System digital zur Verfügung stehen, durch Umkopieren für einen unbestimmten, langen Zeitraum verfügbar halten können.

Leider gilt das nicht automatisch für die in den Dateien enthaltene, für Menschen relevante *Information*. Eine Datei, deren Format wir nicht feststellen können, ist kaum nützlicher als wenn sie verloren wäre, so als ob wir ein in einer unbekannten fremden Sprache oder einem fremden Alphabet geschriebenes Buch hätten[69].

[68] Zuerst müssen die Bits vom Datenträger in ein heutiges System übertragen werden. Wenn das gelungen ist (wofür im schlimmsten Fall eine neue Schnittstelle zum Lesegerät oder überhaupt ein neues Lesegerät gebaut werden muß), kann halbautomatisch nach Dateien mit bekannter Struktur gesucht werden, oder das Format des Dateisystems wird herausgefunden, was manchmal einfach sein kann, da simplere Dateisysteme keine komplexen Datenstrukturen brauchen.

[69] Daraus folgt natürlich nicht, daß es klug wäre, das Buch wegzuwerfen oder die Datei zu löschen. Es könnten ja irgendwann neue Informationen auftauchen, die über die unbekannte Sprache oder das Format der Datei Auskunft geben.

Es gibt einige allgemeine Kriterien mit (positiven oder negativen) Auswirkungen auf die Langzeitverfügbarkeit von Dateiformaten.

Je „offener“ ein Format ist, desto länger ist wahrscheinlich seine Verfügbarkeit. „Offen“ kann ein Format oder ein Verfahren sein, wenn allgemein zugängliche Implementierungen im Quellcode (Open Source) existieren, oder wenn die Spezifikation offen vorliegt (am besten ist natürlich die Situation, wenn beides zutrifft). Das bedeutet nämlich, daß bei Bedarf auch in Zukunft Ansichts- oder Konvertierprogramme für Dateien in dem Format übernommen (bei Open Source) oder geschrieben (bei Verfügbarkeit der Spezifikation) werden können. Patente können die Offenheit eines Formats einschränken, aber das ist mehr ein Hindernis für die aktuelle Verbreitung als für die Langzeitverfügbarkeit (Patente laufen ja nach spätestens 20 Jahren ab).

Wenn keine guten offenen Formate zur Verfügung stehen, ist es zweckmäßig, ein möglichst weit verbreitetes Format zu wählen (vgl. [JoBe01, S. 134]). Die große Verbreitung eines Formats schafft nämlich einen Bedarf und somit auch einen Markt für konkurrierende Lösungen, die mit dem Format umgehen können (es sei denn es ist mit Patenten belegt, aber große Anbieter können selbst in diesem Fall alternative Implementierungen schaffen). Wenn ein Format hinreichend verbreitet ist, gibt es für Software-Anbieter, die einmal die Unterstützung dafür implementiert haben, keinen Grund, diese Unterstützung zu entfernen. Ein gutes Beispiel dafür sind Video-Codecs: während die frühen Videocodecs wie Cinepak (mittlerweile fast 15 Jahre alt) für neue Aufgaben keine Rolle mehr spielen, ist die Unterstützung für sie in Abspiel- und Konvertierprogrammen nach wie vor vorhanden.

Im Folgenden beschreibe ich die Langzeitverfügbarkeits-Aspekte verschiedener Dateiformate.

4.4.1 Unstrukturierte (Freiform-) Textdateien

Unstrukturierte Textdateien sind wahrscheinlich die leichteste Kategorie bei der Sicherung der Langzeitverfügbarkeit. Es gibt zwar die genannten Kodierungsprobleme mit ihnen, aber diese existieren bei fast allen anderen Arten von Computerdateien auch. Die Problematik ist auch recht bekannt, die Lösungswege und Werkzeuge seit Jahren bis Jahrzehnten verfügbar.

Aufgrund dieser Überlegungen hat sich das 1971 gegründete US-Amerikanische Gutenberg-Projekt[70] entschlossen, nach Möglichkeit alle Texte als reine ASCII-Textdateien anzubieten:

[70] Das Gutenberg-Projekt digitalisiert Bücher und andere Texte, die nicht mehr unter den Schutz des Urheberrechts fallen. Derzeit sind mehr als 10.000 Texte vom Projekt verfügbar.
Project Gutenberg http://www.gutenberg.net/

> We stress the inclusion of plain text because of its longevity: Project Gutenberg includes numerous text files that are 20-30 years old. In that time, dozens of widely used file formats have come and gone. Text is accessible on all computers, and is also insurance against future obsolescence.[71]

Das deutsche Gutenberg-Projekt arbeitet erst seit 1994, und setzt – auch weil Deutsch wegen der Umlaute nicht so einfach in reinem ASCII verbreitet werden kann – auf das damals bereits verfügbare HTML-Format, das Formatierungsmöglichkeiten und eine eindeutige Abbildung der deutschen Umlaute bietet.

4.4.2 Strukturierte Textdateien

4.4.2.1 Programmcode Programmcode allein macht selten Probleme für die Langzeitverfügbarkeit, da die Form meist sehr genau festgelegt ist und die Spezifikationen für Programmiersprachen selten und nur vorsichtig geändert werden. Programme in Quellcode-Form sind aber für die meisten Menschen nicht sehr interessant, wichtiger ist die Fähigkeit, sie auszuführen. Das beinhaltet aber eine Menge neue, nicht dateiformatbezogene Probleme, die in Kap. 4.6 auf Seite 79 behandelt werden.

4.4.2.2 Konfigurationsdateien Konfigurationsdateien machen selten Probleme bei der Langzeitverfügbarkeit, da sie nur zusammen mit der dazugehörigen Software Sinn machen, und wenn die Software konserviert werden soll, ergeben sich dabei viel schwierigere Probleme; wenn die gelöst sind, ist die Konfigurationsdatei auch meist ohne Probleme verwendbar. Außerdem sind Konfigurationsdateien meistens schon von ihrem Zweck her so angelegt, daß sie nicht übermäßig komplex und für Menschen möglichst verständlich sind.

4.4.2.3 Separierte Textdateien Durch ihre bekannte und einfache Struktur, und weil sie häufig extra für den Datenaustausch gedacht sind, können separierte Textdateien auch von Menschen mit wenig Erfahrung identifiziert werden. Es ist meistens schnell klar, welche Einstellungen für den Import (am wichtigsten: Trennzeichen und Kodierung) sinnvoll sind. Da Zahlen in Textform (mit Ziffern) gespeichert sind, gibt es kaum Mehrdeutigkeiten und Unklarheiten wie bei der binären Speicherung, höchstens ob das „Komma“ in Dezimalzahlen mit einem Punkt (wie im englischen Sprachraum üblich) markiert ist oder nicht.

Die *Bedeutung* der Datenfelder/Spalten kann jedoch unklar sein, insbesondere wenn es sich um Zahlen oder andere nicht-Text-Informationen handelt. Es hat sich daher als

[71] Project Gutenberg - Public Domain eBook Submission HOWTO http://www.gutenberg.net/howto/spd-howto

Konvention durchgesetzt, daß die Namen der Datenfelder entweder als erste Zeile der Datei oder in einer separaten Beschreibungsdatei angegeben sind, die natürlich auch mehr Angaben zu den Datenfeldern enthalten kann.

Separierte Textdateien werden wahrscheinlich noch lange Zeit als gemeinsames Format für Datenaustausch dienen. Wenn nicht, sind sie einfach in Nachfolge-Formate (z. B. XML, falls sich das durchsetzen sollte) konvertierbar. Es ist daher damit zu rechnen, daß auch die Menschen, die in Zukunft mit der Nutzung und Archivierung von Datenbeständen aus unterschiedlichen Quellen zu tun haben werden, das Format und die geeigneten Werkzeuge kennen werden, um die Daten nutzbar zu machen und, wenn nötig, zu konvertieren.

4.4.2.4 Escape-markierte Textdateien Escape-markierte Textdateien sind für die Langzeitarchivierung (mit den genannten Einschränkungen für Textdateien) relativ unproblematisch, da auch Menschen das Format einfach verstehen können. ProgrammiererInnen mit etwas Erfahrung können dadurch leicht ein Anzeige- oder Umwandlungsprogramm schreiben, oder wenn das zu viel Aufwand wäre, sind die Escape-Sequenzen leicht entfernbar, wodurch eine reine Textdatei entsteht.

4.4.2.5 Markup-basierte Textdateien Diese Dateiformate wurden bewußt mit zwei Zielsetzungen konzipiert: einerseits flexibel genug, um (fast) beliebige Daten aufnehmen zu können, andererseits möglichst einfach strukturiert und für Menschen lesbar und verständlich.

Text-Informationen in XML-Dokumenten sind praktisch immer erkennbar, wenn die Tags für die Anzeige herausgenommen, ausgeblendet oder farbig markiert werden. So zeigen z. B. Web-Browser XML-Dateien, deren Struktur sie nicht kennen, an. Das heißt, daß die grundlegende Information auch ohne Kenntnisse über das erstellende Programm, das Dateiformat (bis auf daß es z. B. XML oder SGML ist) usw. erkennbar ist.

Das XML-Format, das sich unter den tag-basierten Formaten weitgehend durchgesetzt hat, ist hinreichend genau spezifiziert (und die Einhaltung der Spezifikation ist leicht verifizierbar), sodaß es damit sehr wenige mögliche Unklarheiten mit Textkodierung[72], Zeilenenden[73] und anderen für Textdateien typischen Eigenschaften gibt. Die tag-Namen sind automatisch auch ein Hinweis auf ihren Inhalt (allerdings werden häufig verwendete Elemente manchmal abgekürzt, z. B. steht `<p>` in XHTML für „paragraph“). Die Dokumenthierarchie (und damit die Trennung unterschiedlicher

[72] Standardmäßig UTF-8, Abweichungen müssen in der Deklaration der Datei angegeben werden.
[73] Diese spielen bei XML keine Rolle.

Dateneinheiten voneinander) ist eindeutig. Es gibt frei verfügbare automatische Werkzeuge, um die Dokumentstruktur zu bestimmen und die Daten mit wenig Aufwand in andere Formate zu konvertieren.

Aus all diesen Gründen erscheinen markup-basierte Dateiformate (SGML und noch mehr XML) als sehr geeignet für die langfristige Speicherung komplexer Daten, weil es am wahrscheinlichsten sein dürfte, daß sie auch in vielen Jahren noch gelesen, interpretiert und verarbeitet werden können (vgl. [Bor+03, S. 138]).

4.4.3 Binäre Dateien

Anwendungsspezifische binäre Dateien sind die größte Herausforderung für die Langzeitverfügbarkeit, wenn ihr Format nicht genau dokumentiert ist. Wie Jeff Rothenberg beschreibt, ist es in vielen Fällen nicht möglich, herauszufinden, wie eine solche nicht dokumentierte Datei aufgebaut ist:

> Man wird eine Datei durch schlichtes Probieren entziffern können, wenn sie im wesentlichen nur aus einer einfachen Folge von Zeichen besteht; in komplizierteren Fällen wird man damit jedoch kaum zum Ziel kommen. Die Bedeutung einer Datei liegt sowenig in ihren Bits wie die Bedeutung dieses Satzes in seinen einzelnen Wörtern. Um ein Dokument zu verstehen, müssen wir seine Bedeutung in der Sprache des intendierten Empfängers kennen; das ist jedoch in der Regel ein Programm. Eine Multimedia-Präsentation ohne die entsprechende Software zu verstehen ist schlicht unmöglich.
>
> [Roth95b, S. 68]

Wenn eine Spezifikation des Dateiformats zur Verfügung steht, aber kein verwendbarer Programmcode, muß ermittelt werden, ob es Sinn macht, ein Programm zu implementieren, das die Daten in ein verwendbares Format konvertiert oder darstellt. Wenn es nur um wenige kleine Dateien geht, kann es schneller sein, sie durch einen Menschen „händisch“ in ein verwertbares Format bringen zu lassen. Hierbei handelt es sich um eine monotone und fehlerträchtige Arbeit. Wenn das Konvertierprogramm implementiert werden muß, kann daraus je nach Komplexität der Spezifikation und den Anforderungen an das Programm ein größeres Entwicklungsprojekt werden.

Relativ unproblematisch sind nur solche binäre Dateien, die nach einer öffentlich zugänglichen Spezifikation erstellt wurden. Wenn die Spezifikation veröffentlicht ist, gibt es nämlich typischerweise mehrere verschiedene Programme, die diese implementieren, und das hat positive Effekte auf die genaue Einhaltung der Spezifikation sowie die Chancen, daß die Daten auch in Jahren noch verwendbar sind.

4.4.3.1 Textdokument-Formate Wegen der Vielfalt der Textverarbeitungsprogramme, ihrer unterschiedlichen Ansätze und des unterschiedlichen Funktionsumfangs existiert eine große Anzahl von Formaten. Die meisten verbreiteten Programme können einige fremde Formate (und meistens ihre eigenen älteren Formate) importieren, und es gibt Konvertierprogramme für die verbreiteteren Formate. Beim Import und bei der Konversion treten jedoch häufig kleine, subtile oder auch größere Ungenauigkeiten oder Fehler und Informationsverluste auf.

Die textverarbeitungs-eigenen Formate sind (vielleicht mit Ausnahme der XML-basierten Formate, aber es gibt noch zu wenig Erfahrung mit ihnen) für die langfristige Speicherung in der Regel nicht geeignet. Es ist von den kommerziellen Überlegungen der Hersteller abhängig, wie sie mit alten (eigenen und fremden) Formaten umgehen; zum Beispiel unterstützt Microsoft Word eine Menge ältere Formate, aber keine von bestimmten wichtigen Mitbewerbern, wodurch die BenutzerInnen dieser Programme (z. B. OpenOffice.org, StarOffice usw.) gezwungen sind, ihre Dokumente in fremde Formate zu konvertieren.

Leider gibt es einige Textverarbeitungsformate, die mit Patentansprüchen belegt sind[74]; wie sich das auswirkt, muß die Zukunft zeigen. Generell besteht bei der Verwendung solcher Formate ein gewisses Risiko gegenüber solchen, die nicht patentiert sind, weil die Patentinhaber einige Nutzungsarten verbieten können.

Textverarbeitungsdokumente können unterschiedlichen Zielen dienen. Wenn es nur auf die Information im Text ankommt, ist auch ein Textformat ausreichend. In der Praxis wünschen die meisten Leute etwas mehr Formatierung, dann kann das Dokument für langfristige Nutzung in HTML oder RTF konvertiert werden, da diese textbasierten Formate wegen ihrer großer Verbreitung höchstwahrscheinlich noch sehr lange unterstützt werden. Kommt es auf ganz exakte Formatierung (inkl. Originalschriftarten) und auf die Aufteilung in Seiten an, kann das Exportieren in PDF eine Lösung sein.

Komplexe Dokumente mit aktiven Inhalten (z. B. eingebettete Animationen, Makros, die etwas berechnen etc.) sind meistens nicht ohne Informationsverlust in ein anderes Format exportierbar. Für diese Dokumente (von denen es aber relativ wenige geben dürfte) kommt daher nur die aufwendige Aufbewahrung zusammen mit der Originalsoftware in Frage.

Es ist noch nicht abzusehen, wie groß die Probleme, die durch die sogenannten Information Rights Management-Technologien entstehen, werden. Jedenfalls bedeutet der Einsatz einer solchen Technologie, daß die Voraussetzungen für das Lesen eines Dokuments drastisch zunehmen, da nicht mehr nur das Dokument selbst und eine geeignete

[74] z. B.: European Patent Office: Word-processing document stored in a single XML file http://v3.espacenet.com/textdoc?DB=EPODOC&IDX=EP1376387&QPN=EP1376387

Anzeigesoftware vorhanden sein müssen, sondern dazu auch noch ein kompatibler IRM-Server, die Originalschlüssel zur Entschlüsselung des Dokuments und eine Methode, die Person, die das Dokument lesen will, gegenüber dem System zu identifizieren. Es kann argumentiert werden, daß Dokumente, die mit einer solchen Technologie geschützt sind, gar nicht für die Archivierung bestimmt sind; es gibt allerdings rechtliche Vorschriften über die Mindestaufbewahrungsdauer von Firmendokumenten und jenen in der staatlichen Verwaltung. Es kann durchaus passieren, daß die IRM-Technologie bei unüberlegtem Gebrauch zur Unbrauchbarmachung der damit „geschützten“ Dokumente führen wird.

4.4.3.2 Seitenbeschreibungsformate DVI und PostScript erscheinen wegen ihrer mäßigen Verbreitung, der Ausrichtung auf den Druck statt auf Dokument-Austausch und der weniger verbreiteten Software-Unterstützung für die langfristige Speicherung weniger empfehlenswert als das Portable Document Format. PDF-Dokumente aller Versionen sind enorm verbreitet, Adobe und die anderen Hersteller von Anzeigeprogrammen sind daher stark daran interessiert, daß ihre Programme möglichst alle PDF-Dateien anzeigen können. Das Format wurde auch speziell für den Datenaustausch und erweiterbar konzipiert, sodaß grundlegende Änderungen daran in der Zukunft unwahrscheinlich erscheinen. Die Bedenken wegen der Patente auf einzelne Aspekte des Formats sollten langfristig kein Problem darstellen (da Patente ja nach einiger Zeit auslaufen), in der Zwischenzeit könnten sie jedoch die Versorgung mit PDF-Werkzeugen etwas beschränken oder verteuern. Neue PDF-Versionen haben auch Elemente, die nicht in der öffentlich verfügbaren Spezifikation beschrieben sind, und die nur die Original-Werkzeuge von Adobe beherrschen (vgl. [ScTr04, S. 198]).

Trotz der (eher theoretischen) Bedenken ist das Portable Document Format so weit verbreitet, daß es in der Langzeitverfügbarkeit auf jeden Fall besonders beachtet werden muß. Durch die große und noch steigende Verbreitung dürfte es auch langfristig Interesse an der (Weiter-)Entwicklung von Werkzeugen zur PDF-Erstellung, -Verarbeitung und -Anzeige geben, die Offenheit der Spezifikation und das Vorhandensein von Open-Source-Komponenten ermöglicht das auch. Alternativen zu PDF dürften sich nur dann durchsetzen, wenn sie wirklich dramatische Vorteile bieten, die PDF nicht nachmachen kann – aber das erscheint derzeit eher unwahrscheinlich.

4.4.3.3 Bildformate für Rasterbilder Für die Archivierung erscheinen verlustbehaftete Formate wegen des Prinzips der Archivierung als „Konservierung des Originalzustands“ als nicht geeignet. Diese Bedenken können jedoch in den Hintergrund treten, wenn die Bilder in den verlustbehafteten Formaten (ohne sichtbare Verschlechterung) nur ein Zehntel oder noch weniger Speicherplatz brauchen, oder wenn sie gar nicht in

einem verlustlosen Format zur Verfügung stehen (z. B. Fotos aus digitalen Fotoapparaten). Auf jeden Fall sollte eine Konvertierung, falls eine notwendig wird, nur in ein verlustloses Format durchgeführt werden; wegen der ständig wachsenden Speicher- und Übertragungskapazitäten erscheint das machbar.

PNG und TIFF scheinen wegen ihrer großen Verbreitung und Flexibilität für die langfristige Speicherung geeignet zu sein, bei TIFF sollte nur beachtet werden, daß keine Reduktion (z. B. mit dem JPEG-Verfahren) stattfindet. Da die Problematik der Speicherung von reinen Bitmap-Daten seit Jahrzehnten bekannt und gelöst ist, erscheint es unwahrscheinlich, daß sich in diesem Bereich kurz- oder mittelfristig neue verlustfreie Formate durchsetzen und PNG und TIFF komplett verdrängen. Für beide Formate gibt es lizenzfrei einsetzbare, nicht von Patenten behinderte Open-Source-Programmmodule, zukünftige Programme können daher auch leicht diese Formate unterstützen.

4.4.3.4 Vektorformate Vektorbilder werden häufig in eher speziellen Anwendungen benutzt, aber für die allgemeine Weitergabe meistens in ein anderes, weit verbreitetes Format gebracht (z. B. in Bitmap-Bilder umgewandelt oder in PDF-Dokumente eingefügt). Mehrere gute Formate (z. B. EPS und SVG) sind offen spezifiziert, soweit bekannt frei von Patentansprüchen, und es gibt Open-Source-Softwaremodule für ihre Erzeugung, Verarbeitung, Anzeige und Konvertiertung.

Dateien in weniger verbreiteten Formaten können an ihre erzeugenden Programme gebunden sein; in diesem Fall sind Konvertierung (wenn ohne Informationsverlust möglich) oder die Aufbewahrung der Originalsoftware notwendig.

4.4.3.5 Audio- und Videoformate Hier besteht die Vielfalt nicht so sehr in den Formaten, sondern in den Codecs. Das Format einer Datei ist üblicherweise relativ leicht feststellbar, und der verwendete Codec ist dann im Beschreibungs-Teil der Datei angegeben. Allerdings sind die meisten Codecs so komplex, daß sie ohne eine Spezifikation praktisch nicht verwendbar sind, und die Implementierung einer Spezifikation kann auch sehr aufwendig sein und tiefgehende Kenntnisse über Audio- und Videokodierung voraussetzen. In der Praxis ist es daher sehr schwer, Multimediadateien abzuspielen, deren Codec nicht installiert und einfach auffindbar ist.

Konvertierung von Videodaten und verlustbehafteten Audiodaten sollte in der Regel vermieden werden, da durch die unterschiedlichen Reduktionsmethoden die Original-Daten mit unterschiedlichem Auswirkungen geändert werden; die Verluste summieren sich. Es ist also erforderlich, mehrere Abspielprogramme vorrätig zu haben, um alle möglichen Formate abzuspielen. Leider gibt es nicht jedes Programm für jedes Betriebssystem, weswegen sinnvollerweise mehrere Computer vorhanden sein müßten, um

wirklich alles abspielen zu können (oder andere Betriebssysteme müßten emuliert werden, siehe Kap. 5.7 auf Seite 92).

Video am Computer ist noch zu jung, es gibt noch nicht wirklich Erfahrungen mit seiner Langzeitverfügbarkeit. Die Entwicklung eines neuen Codecs ist aber ein größeres Unterfangen; aus diesem Grund ist die Anzahl der Codecs limitiert. Es ist also durchaus vorstellbar, daß Videoabspielprogramme, die die meisten Formate und Codecs kennen, auch in Zukunft – so wie heute – erhältlich sein werden.

Viel größer erscheint das Problem mit DRM-Technologien, siehe Kap. 5.10.1 auf Seite 107.

4.4.4 Datenbanken

Die Information in einer Datenbank ist im Allgemeinen nur zugänglich, wenn die dazugehörige Datenbanksoftware zur Verfügung steht, da die eigentlichen Daten auf dem Datenträger meist in hochkomplexen binären Formaten gespeichert sind. Die Zugänglichkeit der Information einer komplexen Datenbank in Tabellenform bedeutet jedoch noch nicht, daß sie sinnvoll verwendbar ist; dafür ist meist auch die zur Datenbank gehörende Datenbankanwendung notwendig. Bei jeder solchen Anwendung stellen sich auch alle mit der Langzeitverfügbarkeit von Software verbundenen Probleme. Aus diesem Grund gehören Datenbanken zu der Kategorie von Daten, deren Verfügbarkeit am schwierigsten zu sichern ist.

Da die Datenbanksoftware für den Zugriff auf die Daten fast zwingend notwendig ist, sind für die Langzeitverfügbarkeit solche Datenbanksoftware-Produkte geeigneter, die als Open Source verfügbar sind. Diese werden nämlich mit höherer Wahrscheinlichkeit (insbesondere wenn sie populär sind) auf aktuelle Plattformen und Betriebssysteme portiert als proprietäre Datenbanksysteme. Und wenn die Software selbst nicht lauffähig sein sollte, ist zumindest im Quellcode eine genaue Dokumentation der Dateistrukturen vorhanden.

Bei der Umstellung von einer Datenbanksoftware auf die andere ist zu beachten, daß die unterschiedlichen Produkte aus technischen und Wettbewerbsgründen teilweise unterschiedliche Funktionen bieten. Diese Funktionen können die Programmierung vereinfachen, aber sie binden gleichzeitig an die Software des jeweiligen Herstellers. Es erscheint daher zweckmäßig, bei der Entwicklung nur standardisierte Funktionen (z. B. die in der SQL'99-Spezifikation) zu verwenden, wenn jemals die Änderung der Datenbanksoftware notwendig werden kann.

4.5 Lebensdauer der Verweise

Wenn bestimmte Inhalte nicht direkt in eine Datei eingefügt werden sollen oder können, wird oft ein *Verweis* zum eigentlichen Ort des gewünschten Inhalts gesetzt. Solche Verweise (Verknüpfungen, Links) sind aus dem Internet wohl bekannt, es gibt aber viele andere Arten von Verweisen. Zum Beispiel speichern Datenbanken, die auch binäre Dateien wie Bilder enthalten müssen, diese häufig nicht direkt in der Datenbank, sondern irgendwo im Dateisystem des Computers; in der Datenbank ist dann nur ein Verweis auf den Speicherort der Datei enthalten. Auch innerhalb einer Datei können Verweise auf andere Teile der Datei enthalten sein.

Speicherorte von Dateien können sich aus vielen Gründen ändern: organisatorische Umstellungen, Verbesserungen an Programmen, Umstellungen auf andere Softwaresysteme usw. können dazu führen. Im Internet kommt noch die Problematik hinzu, daß Namen und andere Bezeichnungen alles andere als stabil sind[75]; wie in Kap. 1.6 auf Seite 13 zitiert, beträgt die durchschnittliche Lebensdauer von Hyperlinks im World Wide Web nur 44 Tage.

Es gibt einige Ansätze im Internet, sogenannte *persistent URL*s (bleibende Adressen, URL = Uniform Resource Locator) einzuführen[76]. PURL-Register speichern eine eindeutige, unveränderbare (persistente) Adresse und auch die aktuelle physische Adresse der Ressource. Die Anbieter der Ressourcen müssen sich darum kümmern, bei Änderungen der physischen Adresse auch den Verweis im PURL-Register zu ändern. Nach außen wird dann nicht mehr die physische, sondern die persistente Adresse kommuniziert, die Internet-Software kümmert sich um die automatische Weiterleitung an die physische Adresse (vgl. [Bor+03, S. 105]).

Ein möglicher Schwachpunkt dieses Systems ist die Verfügbarkeit des PURL-Registers selbst. Es erscheint zweckmäßig, daß Nationalbibliotheken und andere, langfristig notwendige offizielle Einrichtungen diese Aufgabe übernehmen. Doch selbst dann ist die ewige Verfügbarkeit nicht gewährleistet: Sowohl Namen können sich ändern, wie im Fall der ehemaligen Sowjetunion, als auch die Technologie. Wir wissen z. B. heute nicht, wie lange noch das im World Wide Web übliche Hypertext Transfer Protocol verwendet wird. Für dieses Problem gibt es jedoch eine Lösung: Nationalbibliotheken unterschiedlicher Länder können mit den PURL-Registern zusammenarbeiten und eine gemeinsame Datenbasis nutzen. NutzerInnen könnten dann jederzeit bei ihrer eigenen Nationalbibliothek (oder einem anderem ihnen bekannten PURL-Register) mit den

[75] Früher gab es das Länderkennzeichen `.su` für Computer in der Sowjetunion und ihren Nachfolgestaaten. Nach der Auflösung der Sowjetunion galten die Adressen für eine Übergangszeit von einigen Jahren, aber heute nicht mehr.

[76] Beispielsweise das PURL-Projekt des Online Computer Library Center: http://purl.oclc.org/

jeweils üblichen Methoden auf die Daten zugreifen.

4.6 Lebensdauer von Software

Software ist verwendbar, solange ihr Bitstrom vorhanden ist und die Annahmen, die sie über die Umgebung trifft, gültig sind. Zu diesen Annahmen gehört in erster Linie der Prozessortyp, dann das Vorhandensein bestimmter Software-Schnittstellen, aber auch implizite Annahmen über die Hardware wie Geschwindigkeit oder Speicherausstattung[77].

Die Prozessor-Umgebung kann von einem Original-Computer, einem abwärtskompatiblen Nachfolger oder durch *Emulation* (siehe Kap. 5.7 auf Seite 92) geboten werden. Die Lebenszeit der Original-Computer dürfte sich im Allgemeinen eher in Jahren als in Jahrzehnten messen lassen, die sich ergebenden Probleme sind in Kap. 5.3 auf Seite 85 beschrieben.

Auf dem Prozessor muß eine kompatible Version eines Betriebssystems laufen. Die verbreiteten Betriebssysteme bleiben teilweise über ein Jahrzehnt abwärtskompatibel, das heißt, ältere Programme laufen auch auf neuen Betriebssystemversionen. Diese Kompatibilität kann jedoch nicht immer aufrechterhalten werden, oder der Hersteller ist nicht daran interessiert. Als große, komplexe Projekte sind am Markt weniger erfolgreiche Betriebssysteme teilweise von der Einstellung bedroht (Beispiele aus den 1990-er-Jahren: IBM OS/2 und BeOS), wenn es sich finanziell nicht mehr lohnt, sie weiterzuentwickeln. Ein solches Betriebssystem läuft zwar theoretisch noch auf einem neueren Prozessor, aber andere Teile der Architektur (z. B. Grafikkarten, für die es keine Treiber mehr gibt) können Probleme verursachen, die den Einsatz des Betriebssystems auf neuer Hardware unmöglich machen. Für die Langzeitverfügbarkeit erscheinen daher Open-Source-Betriebssysteme sowie solche, deren Hersteller finanziell sehr erfolgreich ist, geeignet.

Die Software kann auch auf zusätzliche Programmierschnittstellen, die nicht im Betriebssystem enthalten sind, angewiesen sein. Beispiele sind Datenbankschnittstellen und Programmierbibliotheken mit erweiterten (z. B. mathematischen oder grafischen) Funktionen. Diese Schnittstellen können für ein Betriebssystem, eine einzige Betriebs-

[77] Manche Programme, die auf früheren, langsameren Computern entwickelt wurden, funktionieren nicht auf den heutigen Prozessoren, z. B. weil sie mit der Messung der Dauer bestimmter Berechnungen die Geschwindigkeit des Computers feststellen wollen. Wenn der Prozessor die Berechnung z. B. in „0 Millisekunden“ durchführt, kommt es zu einer unzulässigen Division durch Null und das Programm stürzt ab.

Andere Programme wurden verwirrt, als Festplatten mit mehr als 2 Gigabytes Kapazität eingeführt wurden. Die Programme speicherten die Kapazität in Variablen, die nur bis zu dieser Anzahl Zahlen enthalten konnten; wenn das Betriebssystem einen größeren Wert zurückgab, zeigte das Programm eventuell eine negative Festplattenkapazität oder völlig unsinnige Zahlen an.

systemversion oder auch für mehrere Betriebssysteme geschrieben sein; die universellen sind natürlich besser für die Sicherung der Langzeitverfügbarkeit geeignet.

In Einzelfällen kann sogar eine bestimmte Hardware für die Lauffähigkeit der Software erforderlich sein. Vor allem Software für Spezialaufgaben wie Meßdatenerfassung, medizinische Anwendungen, Fertigungssteuerung etc. kann in diese Kategorie fallen, wenn sie die Geräte nicht über allgemeine Schnittstellen, sondern durch die Software selbst ansteuert. Eine andere Kategorie bilden sogenannte „Dongles“ für den Kopierschutz; das sind kleine Hardwarekomponenten, die zusammen mit dem Programm verkauft werden, das nicht läuft, wenn es die Hardware nicht findet (siehe Kap. 5.10.1 auf Seite 107). Da Hardware und Schnittstellen relativ schnell veralten, ist es bei dieser Art von Software sehr wahrscheinlich, daß sie nach einigen Jahren überhaupt nicht mehr verwendbar ist.

In den letzten Jahren haben einige Hersteller ebenfalls aus Kopierschutzgründen die sogenannte „Produktaktivierung“ eingeführt: Die Software arbeitet nur einige Tage oder Wochen lang, wenn sie nicht online oder telefonisch registriert wurde. Der Einsatz solcher Software geschieht also rein nach Ermessen der Herstellerfirma. Wenn sie in Konkurs geht, aufgekauft wird oder nicht mehr an der Software interessiert ist, wird diese für die AnwenderInnen komplett wertlos, da sie nicht mehr (oder nur mit teilweise illegalen Tricks) einsetzbar ist.

4.7 Information aus ökonomischer Sicht

Durch die technische Entwicklung werden Produktion und Verarbeitung von Information immer einfacher, daher auch immer billiger (vgl. [Shen97, S. 29]). Die Schwelle, bestimmte Informationen überhaupt zu produzieren, nimmt ab (z. B. mit den Digitalkameras, die keine Kosten für die Fotoentwicklung nach sich ziehen, wird wahrscheinlich mehr fotografiert). Die Menge der produzierten, verarbeiteten und publizierten oder für eigenen Bedarf gespeicherten digitalen Information nimmt jedes Jahr um ca. 50 % zu (vgl. [Zimm01, S. 57]). Was sich nicht im gleichen Ausmaß weiterentwickelt, sind Zeit und Aufmerksamkeit der Menschen. (Menschen, die zu viel Information ausgesetzt sind, können laut [Shen97, S. 37-39] sogar psychische Störungen davontragen.)

Ein signifikanter Teil der produzierten Informationsmenge dient unserer Unterhaltung (z. B. Filme, Musik und Romane) oder Unterrichtung (Nachrichten, Fachliteratur). Diese Inhalte haben die Eigenschaft, daß ihr Nutzen für uns abnimmt, wenn wir sie schon kennen, oder – in unterschiedlichem Ausmaß – wenn sie alt sind. Aus diesem Grund sind wir auch nicht bereit, den selben Preis für sie zu zahlen wie früher. Der in Geld ausgedrückte Marktwert der Information nimmt also während ihres Lebenszyklus ab. (Dieser Effekt kann sich in der Verringerung eines „globalen Preises“ – z. B. Fern-

sehausstrahlungsrechte eines Films – ausdrücken oder so, daß zwar die Preise – wie bei Musik-CDs, auf die ihr Produzent ein Monopol hat – im Laufe der Zeit konstant bleiben, aber ihre Verfügbarkeit nimmt ab, weil es für die Geschäfte irgendwann teurer wird, die CD zu lagern als der erwartete Erlös aus dem Verkauf. Dies führt auch zu verringerten Einnahmen.)

Die Leichtigkeit der Informationserstellung, -verarbeitung und -weitergabe ermöglicht auch kostenlos (legal oder illegal) zugängliche Inhalte. Die Film- und Musikindustrie berichten von großen Verlusten durch illegale Kopien ihrer Werke, die im Internet gratis verbreitet werden; es ist aber auch eine Tatsache, daß Musik, Bücher, Poesie, Nachrichten, Fachliteratur zu allen erdenklichen Themen, Spiele und noch viele andere Inhalte legal mit Wissen der Inhaber der Urheberrechte (oder nach Ablauf der Dauer des Urheberrechts) im Internet verfügbar sind. All diese Inhalte konkurrieren mit der immer größeren Menge der kommerziellen Inhalte um eine mehr oder weniger konstante Größe, nämlich die Aufmerksamkeit der Menschen in den entwickelten Ländern. Dies kann zu einer weiteren Verringerung des Marktwertes der Information führen.

Firmen und Einzelpersonen, die mit der Produktion von Unterhaltungsinhalten Geld verdienen, sehen also Konkurrenz einerseits von den kostenlosen Inhalten, andererseits von *ihren eigenen* früheren Produkten. Wer sich im Fernsehen einen alten Film anschaut, kann am selben Abend nicht ins Kino gehen und dort für mehr Geld einen neuen ansehen, selbst wenn das Geld vorhanden wäre. Die Anzahl der Abende in einer Zeitperiode ist limitiert. Aus diesem Grund sind Inhaltsproduzenten gar nicht daran interessiert, daß Unterhaltungsinhalte am Markt langlebig sind[78], da sie jeweils nur beim Verkauf, nicht aber bei der Nutzung der Inhalte Geld bekommen[79]. Wenn Unterhaltungsinhalte in großem Maßstab verloren gehen (z. B. durch den Umstieg auf die Compact Disc, wodurch Schallplattensammlungen durch die schlechte Verfügbarkeit von Plattenspielern und Ersatzteilen uninteressanter werden – es ist leichter, eine gewünschte Aufnahme auf CD neu zu kaufen als sich um den Plattenspieler zu kümmern), steigt also die Nachfrage nach neuen Produkten oder neuen Ausgaben der alten Produkte.

4.8 „Soziale“ Lebensdauer

Gesellschaftliche Veränderungen oder gewaltsame Umstellungen haben schon oft in der Geschichte der Menschheit dazu geführt, daß bestimmte Informationen unerwünscht wurden. Sehr bekannte Beispiele sind die Bücherverbrennungen in Nazi-Deutschland

[78] Sie können natürlich daran interessiert sein, ihre eigenen Inhalte gut aufzubewahren, da die populären Stücke auch nach Jahrzehnten noch verwertbar sein könnten.

[79] Mit der digitalen Technologie wären aber solche pay-per-use-Modelle möglich und sie werden mittlerweile auch getestet.

oder das Verbrennen der Bücher der Bibliothek von Alexandria im Jahr 641 auf Befehl des Kalifen Omar mit der Begründung, daß sie entweder dem Koran widersprächen oder nur das wiederholten, was im Koran bereits enthalten sei und sie deswegen in beiden Fällen unnötig wären (vgl. [Canf98, S. 103]).

Je zentraler und leichter zugänglich Informationen gespeichert sind, desto effizienter können sie auch vernichtet werden. Oder sie werden unterdrückt, indem sie zwar nicht vernichtet, aber in den elektronischen Katalogen unauffindbar gemacht werden. Die Löschung digitaler Daten produziert auch keinen auffälligen Rauch wie das Verbrennen vieler Bücher auf der Straße. Unter bestimmten Umständen ist es möglich, die Löschungen so durchzuführen, daß sie gar nicht gleich entdeckt werden.

In unserer Zeit werden auch Gesetze benutzt, um Informationen zu unterdrücken. Es ist z. B. bekannt, daß die Church of Scientology mit großem Eifer unter dem Vorwand des Schutzes von Urheberrechten kritische Materialien im Internet zu verhindern versucht. Die Volksrepublik China und viele arabische Staaten haben ausschließlich überwachte und zensierte Internet-Verbindungen und blockieren Medien, die den offiziellen Meinungen der dortigen Regierungen nicht entsprechen.

5 Lösungsansätze

In einzelnen Fällen sind Probleme mit verlorenen oder verlustgefährdeten Daten bereits aufgetreten. Wenn die Daten noch gerettet werden konnten, wurden die geeigneten Maßnahmen mit EDV-Methoden, in der Regel von der selben Organisation, die die Daten produziert hat und sie weiterhin benötigte, gesetzt. Die heutige Situation ist jedoch anders: einerseits bekommen Archive und Bibliotheken zunehmend digitale Information zur Aufbewahrung, andererseits speichern auch Privatpersonen immer mehr Information digital. Beide brauchen daher standardisierte, gut untersuchte Methoden mit fertigen Lösungen, die am besten auch ohne Programmierkenntnisse anwendbar sind.

5.1 Metadaten als Voraussetzung

Bei großen Informationsmengen ist das erste Problem das Auffinden der gewünschten Informationen. Dafür ist eine standardisierte Erfassung der relavanten Daten (z. B. Dokumenttitel, AutorIn, Erstellungszeitpunkt; für digitalisierte Information der Speicherort, das Format, eventuell Zugriffsrechte u. v. m.) notwendig. Bibliotheken haben z. B. schon immer Listen oder Kataloge ihrer Bücher geführt (siehe [Cass02, S. 19] – es hat allerdings bis ins Mittelalter gedauert, bis die Listen in eine standardisierte, z. B. nach dem Alphabet geordnete Form gebracht wurden).

Die beschreibenden Daten über Objekte (wirkliche und digitale) werden *Metadaten* oder „Daten über Daten“ genannt. Sie sind die Grundlage jeder systematischen Informationsarchivierung.

Es gibt in der elektronischen Datenverarbeitung viele Ansätze zur Speicherung von Metadaten. Fast jedes Computersystem bietet die Möglichkeit, Dateien in Verzeichnissen und Unterverzeichnissen beliebiger Tiefe zu speichern und mehr oder weniger beliebig zu benennen. Häufig gibt eine Dateinamenerweiterung (z. B. „.html“) den Typ der Datei an. Mit diesen Mitteln ist es schon – bei geeigneter Standardisierung innerhalb einer Organisation – möglich, eine gewisse Systematik aufzubauen, zum Beispiel indem Dateien nach folgendem Benennungsschema gespeichert werden: „daten/<Jahr>/<AutorIn>/<Titel>.<Typ>“. Doch erfüllt dieses Schema nicht alle Anforderungen, die bei Informationen auftauchen können: zum Beispiel kann ein Dokument durchaus von mehreren oder unbekannten Leuten geschrieben worden sein. Der Titel kann nur in Verbindung mit dem Untertitel aussagekräftig sein, der Typ ist vielleicht nicht eindeutig (z. B. haben alle Dokumente, die jemals von Microsoft Word gespeichert wurden, die Erweiterung „.doc“, obwohl sich das Format über die Versionen stark geändert hat). Diese Art der Speicherung mag also für private Zwecke ausreichen,

ist aber für die Anforderungen von Bibliotheken, Archiven oder größeren Firmen nicht geeignet.

Viele Dateitypen erlauben es, fix vorgegebene oder auch selbst benennbare Metadaten in die Datei einzufügen. In vielen Büroprogrammen ist diese Funktion unter Datei/Dokumenteigenschaften oder an ähnlicher Stelle erreichbar. Wenn das nicht geht – z. B. bei manchen verbreiteten Bildformaten –, müssen die Metadaten separat, etwa in einer eigenen Datei pro Bilddatei oder für ein ganzes Verzeichnis gespeichert werden. Hierbei stellt sich allerdings das Problem, welches Format diese Metadatendatei haben soll.

Alternativ dazu können unter manchen Betriebssystemen (MacOS, Windows XP) sogenannte „erweitere Attribute“ von Dateien gespeichert werden. Diese Attribute wären auch für die Speicherung der Metadaten verwendbar, aber das ist wegen der Bindung an das Datei- und Betriebssystem nicht empfehlenswert.

Die dateibasierten Ansätze haben allein genommen den Nachteil, daß für eine Suche immer sämtliche Dateien oder Metadatendateien durchsucht werden müssen. Häufig ist sogar die Kenntnis des Dateiformats notwendig, um die Metadaten überhaupt lesen zu können. Deswegen ist eine separate Aufbewahrung der Metadaten vorteilhaft (vgl. [Bor+03, S. 11]). In der Praxis werden daher die Metadaten automatisch oder manuell auch in einer eigenen Datenbankanwendung erfaßt, die dann als Katalog dienen kann. Wenn aber die Metadaten in einer Datenbank gespeichert sind, ist ihre separate Speicherung in der Datei oder in Metadatendateien überflüssig (redundant). Viele Archivierungssysteme verzichten daher auf datei- und dateinamenbasierte Metadatenspeicherung: jedes Dokument bekommt eine eindeutige Nummer zugewiesen, über die es dem System bekannt ist; alle Metadaten stehen in der Datenbank.

Die Speicherung von Metadaten in Datenbanksystemen führt wieder zu den Problemen mit der Langzeitverfügbarkeit von Datenbanken, weswegen hier erhöhte Vorsicht geboten ist[80].

Für die Langzeitverfügbarkeit sind zusätzliche Metadaten (engl. „*preservation metadata*“) erforderlich, die für den täglichen Betrieb überflüssig erscheinen, etwa über die Version der Software und des Betriebssystems, die das Dokument erstellt haben, die ver-

[80] Die Bibliothek der Eötvös Lóránt-Universität in Budapest hat zwischen 1985 und 1995 in einer Katalogsoftware ca. 30.000 bis 40.000 Bücher erfaßt. Die Software wurde damals vom Hersteller nicht mehr weiterentwickelt und entsprach nicht den modernen Anforderungen (z. B. Katalogsuche übers World Wide Web). Es wurde daher durch ein neues System ersetzt, aber die Daten konnten nicht übernommen werden. Wer im Katalog nach Büchern, die zwischen 1985 und 1995 angeschafft wurden, suchen wollte, mußte dort in der Bibliothek an den speziellen Arbeitsplätzen mit dem alten System arbeiten. Mittlerweile sind alle Daten noch einmal manuell eingegeben worden und das alte System besteht nicht mehr.

Quelle: Interview mit Petrovics Mária, damals die Leiterin der Universitätsbibliothek, am 15. 8. 2002; aktuelle Situation: Gespräch am 24. 9. 2004.

wendete Hardware, rechtliche Vorschriften (z. B. Ablauf der Dauer des Urheberrechts oder der Aufbewahrungspflicht), eventuell Checksummen und digitale Signaturen usw.

Es empfiehlt sich, für die Speicherung der Metadaten ein möglichst langlebiges Format zu wählen. Jeff Rothenberg hielt 1995 noch simple Textformate für die geeignetsten (vgl. [Roth95a, S. 29]), neuere Initiativen wie VERS (Victorian Electronic Record Strategy, [Wau+00, S. 180]) oder XMetaDiss[81] bevorzugen XML-basierte Formate.

5.2 Überblick der vorgeschlagenen Ansätze

Viele Probleme mit Daten, deren Verlust droht, können in ad-hoc-Projekten behoben werden. Die beteiligten Personen erkennen vielleicht gar nicht, daß es sich um ein generelles Problem mit der Langzeitverfügbarkeit handelt; das Projekt wird als normale EDV-Aufgabenstellung angesehen.

Im Folgenden stelle ich Methoden vor, die allesamt beanspruchen, allgemeine, für mehrere gleichartige Probleme geeignete Vorgehensweisen zu bieten. Sie alle sind in unterschiedlichem Ausmaß auf EDV-Kenntnisse und Erfahrung angewiesen, die aber heute teilweise nicht einmal bei Bibliotheken und Archiven vorhanden sind (vgl. [JoBe01, S. 28]).

5.3 Hardware-Museum

Ein ziemlich intuitiver Ansatz ist das Aufheben alter Computer und Hardware-Peripherie zusammen mit Unterlagen zur Bedienung und Wartung und der Software. Diese Methode wird in der Literatur als „Hardware-Museum“ bezeichnet (vgl. [Bor+03, S. 16]), wobei wegen der kürzeren Aufbewahrungszeit (höchstens einige Jahrzehnte) und der Nutzung der Computer (im Gegensatz zur reinen Ausstellung), und weil auch Software aufbewahrt wird, dieser Begriff etwas ungenau ist („Computer-Archiv“ wäre wahrscheinlich besser).

In einer kleineren Organisation mit nicht sehr komplexen Daten kann das Aufbewahren der alten Systeme für den Zugriff auf alte Daten für einen gewissen Zeitraum sehr nützlich sein. Für wirklich langfristige Sicherung sowie für komplexere Aufgaben ist die Methode aber nicht geeignet.

Das erste Problem ist die Lebenszeit der Hardware. Insbesondere die beweglichen Komponenten und solche, die eine eigene Wärme produzieren und deswegen wiederholten Temperaturschwankungen ausgesetzt sind, gehen nach begrenzter Zeit kaputt, aber selbst ohne Nutzung vollzieht sich eine gewisse Verschlechterung (vgl. [Roth99, S.

[81] Format des Metadatensatzes Der Deutschen Bibliothek für Online-Hochschulschriften XMetaDiss Referenz http://www.ddb.de/standards/xmetadiss/

13]). Da Computerkomponenten aus hochintegrierten elektronischen Bauteilen bestehen, ist eine Reparatur in der Regel schwierig, daher ist meist nur der Austausch der betroffenen Komponente möglich. Die Hersteller sind in der Regel nicht daran interessiert, Ersatzteile für ihre ziemlich alten Geräte anzubieten, da der erzielbare Umsatz wegen der kleinen Stückzahlen gering ist. (Dies hängt natürlich mit dem Marktanteil zusammen; für verbreitetere Geräte kann die Versorgung länger gewährleistet sein, weil mehr Nachfrage besteht.)

Die nächste Herausforderung ist das häufige Fehlen von Schnittstellen zu anderen, heutigen Computersystemen. In der Welt der Personal Computer der 1980-er-Jahre war es etwa sehr selten, daß ein Computer irgendeine Art von Netzwerk unterstützte, und manche Systeme verwendeten auch eigene Diskettenformate oder Dateisysteme, die von keinem anderen System lesbar waren. Die Daten sind in ein solches System „eingesperrt“, selbst wenn alles funktioniert, können sie nur mit der Original-Software am Original-System vor Ort betrachtet werden, eine Nutzung übers Internet etwa ist nicht möglich.

Die Probleme mit der Lebensdauer der Datenträger wurden bereits detailliert beschrieben. Wenn die Original-Datenträger nur im Original-System lesbar sind und keine Ersatzdatenträger zur Verfügung stehen, ist auch kein Umkopieren möglich; dann führt der Verlust der Daten auf dem Datenträger zu ihrem permanenten Verlust, wenn keine aufwendigen Rettungsmaßnahmen, die sich in den meisten Fällen nicht lohnen dürften, eingeleitet werden. (Vgl. [Roth99, S. 12])

Die Bedienung älterer Computersysteme kann für BenutzerInnen, die noch nie mit ihnen gearbeitet haben, sehr verwirrend sein. Grafische Benutzeroberflächen setzten sich erst vor ca. 20 Jahren, in manchen Bereichen noch später durch, vorher war die Befehlszeilen-Bedienung vorherrschend. Hierbei setzten viele Hersteller auf die Entwicklung eigener Befehle, die nur mit den dazugehörigen Handbüchern verständlich waren. Wir können heute davon ausgehen, daß für die Bedienung von Unix- und DOS-ähnlichen Systemen noch relativ leicht Leute gefunden werden können, andere Systeme können eine längere Suche nach geeigneten Personen oder die Einarbeitung in ein fremdes System notwendig machen.

Hardware zu entwickeln ist viel aufwendiger als Software zu schreiben. Für die meisten Systeme steht daher weitaus mehr Software zur Verfügung als ein Exemplar der Hardware verträgt. Es kann sich als unmöglich herausstellen, z. B. alle gängigen Textverarbeitungsprogramme auf einem einzigen Computer zu installieren. Jede Installation steigert auch die Komplexität eines Systems und bedeutet daher ein Risiko.

Es ist nicht klar, welche Computersysteme wie lang am Markt verfügbar sind und ob es kompatible Nachfolgesysteme geben wird (und ob diese auch wirklich hundertprozen-

tig kompatibel sind). Ein vorsichtiges Computer-Archiv muß also eine größere Anzahl von Systemen anschaffen und warten. Die Kosten und das notwendige Know-How dafür können schnell ansteigen.

Zusammenfassend kann gesagt werden, daß das Aufheben alter Systeme als Vorstufe zu anderen Verfahren recht nützlich bis absolut notwendig sein kann, als Mittel der langfristigen Sicherung der Information ist es aber nicht geeignet (vgl. [Roth99, S. 13] und [Bor+03, S. 18]).

5.4 Umkopieren

Eine naive Vorgangsweise ist das simple Umkopieren[82] der Daten auf aktuelle Datenträger, ohne zu prüfen, ob sie in der neuen Umgebung auch noch verwendbar sind. Häufig sind insbesondere für verbreitete Dateiformate noch auf ein bis zwei Generationen von Systemen geeignete Programme vorhanden, deswegen ist manchmal nicht sofort erkennbar, daß es ein Problem gibt.

Diese Methode ignoriert eine wesentliche Eigenschaft der digitalen Daten, nämlich daß sie häufig an ihre Umgebung gebunden sind (vgl. Kap. 4.4 auf Seite 69). Sie ist daher für die langfristige Verfügbarkeit der Information allein nicht geeignet. (Natürlich ist Umkopieren ein wesentlicher Teil aller besseren Methoden.)

5.5 Verwendung standardisierter Dateiformate

Es ist eine verbreitete und gut argumentierbare Annahme, daß Standards für Dateiformate durch ihre im Normalfall öffentlich zugängliche Dokumentation und in der Regel größere Verbreitung eine längere Lebenszeit haben können. Rothenberg nennt in [Roth99, S. 10] verschiedene Probleme mit der Betrachtung von Standards als Allheilmittel, wobei ich der Meinung bin, daß er in einigen Punkten aus heutiger Sicht (fünf Jahre später) Unrecht hat. Zwei Faktoren, die 1999 anscheinend noch nicht so gut sichtbar waren, führen meines Erachtens zu einer besseren Situation als von Rothenberg dargestellt.

Erstens übt der Markt durch das Internet einen Druck *zur* Standardisierung aus. Früher war ein Datenaustausch zwischen verschiedenen Computerplattformen weniger üblich, mit dem Internet wurde aber der Web-Browser mit der relativ genau umrissenen Funktionalität zum Standard, und die Bereitschaft, für das Ansehen von Inhalten neue Software zu installieren, nahm ab. Die Anbieter der Information mußten sich an die

[82] In der Literatur häufig als „refreshing“, Auffrischung bezeichnet, da die Bits sozusagen frisch, „wie neu“, auf den Datenträger gelangen, wodurch sich ihre Haltbarkeit verlängern kann (vgl. [Bor+03, S. 45]).

Erwartungen des Publikums anpassen, das heißt, jene Formate verwenden, die möglichst universell zugänglich sind. Das sind in der Regel die standardisierten Formate wie HTML, GIF, JPEG, PNG und PDF. Es wurde viel Aufwand in die Verbesserung genau diese Formate gesteckt, der früher wahrscheinlich zur Entwicklung komplett neuer Formate geführt hätte. Die standardisierten Formate sind also heute deutlich mehr als noch in den 1990-er-Jahren für die wirklichen Anforderungen (selbst wenn es sich um recht spezielle Anforderungen handelt) geeignet. Das PNG-Format ist ein gutes Beispiel für eine praktisch optimale, kaum noch zu verbessernde Lösung für ein wohl verstandenes und verbreitetes Problem. Es ist heute einfach nicht mehr interessant, dasselbe Problem noch besser zu lösen, da die einzige Optimierungsmöglichkeit eine gegenüber PNG noch kleinere Dateigröße wäre, aber der Bedarf daran nimmt mit der Verbreitung immer größerer Speicher- und Übertragungskapazitäten ständig ab.

Der zweite wesentliche Faktor, der die Langzeitverfügbarkeit standardisierter Dateiformate heute besser erscheinen läßt als 1999 ist der Aufstieg der Open-Source-Software. Natürlich hat OSS auch 1999 schon existiert, aber sie wird heute deutlich öfter eingesetzt und in Verbindung damit hat auch das Software-Angebot stark zugenommen. Für praktisch alle standardisierten Formate gibt es als Open Source zugängliche Software-Bibliotheken, die teilweise sogar den Markt dominieren. Das hat wiederum zwei Konsequenzen: erstens ist die Motivation seitens der Hersteller, Formate willkürlich „weiterzuentwickeln“ (also zu ändern), weniger vorhanden, da in einem von Open Source beeinflußten Markt die Interessen der Nachfrageseite deutlich größeren Einfluß haben. Zweitens steht der Quellcode fertiger Anzeige- und Bearbeitungsprogramme für die Dateiformate zur Verfügung, wodurch die relativ leichte Übertragung der Programme auf neue Plattformen gewährleistet ist und nicht vom Vorhandensein, Willen und den kommerziellen Erwartungen der Hersteller abhängt.

Heutige Standards werden in offeneren Prozessen geschaffen als früher. Die mit dem Internet möglich gewordenen Methoden der internationalen Zusammenarbeit erlauben die schnellere Entwicklung und die bessere Praxis-Orientierung von Standards als früher übliche kleine Standardisierungsgremien. Dadurch ist die Beeinflussung in Richtung herstellerabhängiger Technologien geringer und die Anwendung der Standards kann schon während der Entwicklung beginnen (viele Webseiten verwenden z. B. für den Datenaustausch das RSS-Format[83] in der Version 0.91, was darauf hindeutet, daß der Standard schon vor der offiziellen Verabschiedung (Versionsnummer 1.0) durchaus verwendbar war). Das alles fördert die Verbreitung standardisierter Formate und verringert die Chancen der nicht standardisierten, deren Bedeutung nimmt daher ab.

[83] RSS: Rich Site Summary oder Really Simple Syndication. Ein XML-basierter Standard für die Publikation von Neuigkeiten auf Webseiten.

Natürlich können Standards auch Nachteile haben. Viele Dokumente und Informationssammlungen lassen sich nicht ohne Reduktion ihrer Eigenschaften (z. B. eingebettete Funktionalität, Änderungsgeschichte, plattformspezifische Erweiterungen usw.) in standardisierten Formaten wiedergeben (siehe auch Kap. 3.5.8.6 auf Seite 47). In diesen Fällen ist also die Verwendung von Standards nachteilig oder sogar unmöglich. Allerdings kann häufig eine „Gebrauchs-Kopie“ in standardisierten Formaten angefertigt werden, die manche Arten der Nutzung erleichtern kann.

Die Verwendung von Standards reduziert die Vielfalt der zu archivierenden Dateiformate und damit die Komplexität. Dadurch sind weniger Kenntnisse (z. B. über Hunderte von Grafikformaten) und ggf. Werkzeuge (z. B. Software) erforderlich, was auch fortgeschrittenere Verfahren wie die Migration oder Emulation, falls sie später doch notwendig werden, einfacher machen kann (vgl. [JoBe01, S. 107]). (Es bedarf dann nicht so vieler Migrationsprogramme oder emulierter Umgebungen, und die Programme können durch das offen dokumentierte Format leichter und präziser geschrieben werden.)

Die Verwendung von Standards kann also eine längere Lebensdauer von Informationen, die sich in standardisierten Formaten speichern lassen, sichern helfen als die bisher behandelten anderen Methoden (vgl. [JoBe01, S. 57]). Wirklich langfristige Aussagen (z. B. über 40-50 Jahre) sind nicht möglich, aber die Verwendung geeigneter Standards versperrt nicht den Weg zu länger wirkenden Methoden wie Migration oder Emulation, im Gegenteil: sie können durch Standards sogar unterstützt werden.

5.6 Migration (Konversion)

Konversion bedeutet in diesem Zusammenhang die Umwandlung von einem Format in ein anderes. Migration ist ein Überbegriff für verschiedene Methoden der Konversion und inkludiert auch die automatische Konversion großer Datenmengen, die Anpassung an neue Umgebungen und die Überprüfung des Ergebnisses. Das Ziel der Migration ist, die Daten in aktuelle, zur jeweiligen Zeit übliche und mit aktueller Software betrachtbare Formate zu bringen. Das sichert den bestmöglichen Zugang zur Information auch in der Zukunft.

Der Migration als Methode liegt die Annahme zugrunde, daß neue Formate sich nur dann durchsetzen können, wenn sie alle notwendigen Eigenschaften der früheren Formate enthalten und ihnen gegenüber auch noch Vorteile bieten. Unter dieser Annahme erscheint es logisch, daß eine vollständige Konversion des früheren Dateiformats in das neuere möglich ist. Dadurch können die Dateien immer in den gerade üblichen Formaten vorliegen und es geht nichts verloren.

Diese Annahme ist aber nicht immer gültig. Es kommt durchaus vor, daß durch die

Verwendung eines neuen Formats die Eigenschaften der alten Daten verloren gehen. Zum Beispiel gibt es bei Internet-Veröffentlichungen im HTML-Format keine gute, allgemein verwendbare Methode, um Dokumentteile zu kennzeichen, was wir z. B. bei Büchern mit Hilfe der Seitenzahlen durchaus gewohnt sind.

Vieles, was bei der Verwendung standardisierter Dateiformate gilt, kann auch über die Migration gesagt werden. Mit der größeren Bedeutung öffentlich spezifierter Dateiformate und von Open-Source-Software finden Formatwechsel seltener statt, und bessere Dokumentation sowie das Vorhandensein leichter zugänglicher und anpaßbarer Werkzeuge erleichtern die seltenere Migration. Die Komplexität verringert sich, da weniger Dateiformate zu migrieren sind, wodurch weniger unterschiedliche Migrationswerkzeuge angeschafft, getestet, entwickelt und benutzt werden müssen (vgl. [Bor+03, S. 41]). Allerdings gelten auch die Einschränkungen bei der Übertragung komplexer Dokumente in Formate, die nicht alle Eigenschaften des Originals enthalten können (Reduktion).

Jeff Rothenberg nennt in [Roth99, S. 14ff] weitere Probleme, die sich ergeben, wenn Migration als einzige Methode der Langzeitsicherung angewendet wird: „the nearly universal experience has been that migration is labor-intensive, time-consuming, expensive, error-prone, and fraught with the danger of losing or corrupting information. Migration requires a unique new solution for each new format or paradigm and each type of document ...“. Jeder Migrationszyklus sei aufwendig und schwer automatisierbar. Außerdem seien die Migrationszyklen von außen (der Entwicklung des Marktes) vorgegeben, eine Vorausplanung unmöglich.

Ein weiterer Kritikpunkt von Rothenberg ist, daß die Migration nicht sicherstelle, daß Änderungen zwischen Original und migrierter Kopie detektierbar sind ([Roth00, S. 33]). Dies ist jedoch falsch. Wenn die Software nicht nur in eine Richtung (vom alten Format zum neuen) konvertieren kann, sondern auch zurück (was bei Konvertiersoftware für viele heutige Dateiformate gilt), kann das zurückkonvertierte Exemplar mit dem Original verglichen werden (dies ist sogar automatisierbar). Eine andere Möglichkeit ist, zwei verschiedene Programme unterschiedlicher Hersteller für die Konvertierung oder für die Prüfung des Ergebnisses der Konvertierung zu verwenden. Bei manchen Dateiformaten wird bei der Zurückkonvertierung eine komplett mit dem Original-Bitstrom identische Datei herauskommen, bei anderen eventuell nur eine „funktional identische“, d. h. eine, die in allen relevanten Aspekten dem Original entspricht[84]. Bei einem Bitmap-Bild wäre

[84] Daß die Dateien möglicherweise nicht identisch sind, spielt keine Rolle, solange sie der jeweiligen Formatspezifikation entsprechen. Z. B. sind bei PNG-Dateien mehrere Kompressionsstufen möglich, die nur den Berechnungsaufwand beim Speichern ändern, nicht aber das Bild. Wenn eine PNG-Datei auf einem alten Computer gespeichert wurde, wurde vielleicht aus Geschwindigkeitsgründen eine niedrigere Kompressionsstufe gewählt; bei der Migration kann auf dem üblicherweise viel schnelleren Computer der höchste Kompressionsgrad verwendet werden.

das ein Bild, das Bildpunkt für Bildpunkt mit dem Original verglichen keinerlei Unterschiede enthält. Diese Vergleichsmöglichkeit stellt also sicher, daß die Authentizität von Dokumenten auch bei der Migration gesichert werden kann[85] (siehe Migrationsexperiment auf Seite IV).

Bibliotheken und Archive mußten ja auch bereits die funktional identische Aufbewahrung von Information akzeptieren, als Bücher massenhaft begannen, wegen sauren Papiers auseinanderzufallen (vgl. [Smit99a, S. 6]). Das Kopieren auf Mikrofilm rettet den Informationgehalt des Objekts, auch wenn das Objekt selbst verlorengeht. Ähnlich ist das Ergebnis bei der funktional identischen Migration geeigneter Dateiformate.

Auch die Migration ist als einzige Methode der Langzeitsicherung ungeeignet, da nicht für alle Formate eine funktional identische (oder selbst eine weniger anspruchsvolle) Konversion möglich ist. Wie die Verwendung standardisierter Formate erleichtert sie jedoch auch die noch fortgeschrittenere Methode, die Emulation. Wenn die konvertierten Originale nach der Migration nicht gelöscht werden – wofür mit der ständig steigenden Speicherkapazität wirklich kein Grund besteht – verbaut die Migration auch nicht den Weg, die Originale mit Hilfe der Emulation in ihrer Originalumgebung zu betrachten; dafür sind leicht Ansichtskopien herstellbar, die sich mit den zur jeweiligen Zeit üblichen Programmen anschauen lassen. Die Nützlichkeit der archivierten Informationen für einen größeren Personenkreis steigt dadurch drastisch an. Der Zugang übers Internet (oder zukünftige Netzwerke) wird auch viel einfacher, wenn die Daten in zur jeweiligen Zeit üblichen Formaten vorliegen.

Es ist zweckmäßig, für jedes Format einzeln zu entscheiden, ob die Migration das geeignetste Verfahren der Langzeitsicherung ist. Meines Erachtens gilt das für ganze Klassen von Dateien, etwa Nur-Text-Dokumente, Druckbilder von Textdokumenten (auch mit Bildern und internen Hypertext-Verknüpfungen), Bitmap-Bilder, Ton- und

---

Bei tag-basierten Textdateien ist die Verwendung von „whitespace“ (Leerzeichen, Zeilenwechsel, Tabulator) komplett freigestellt, manche Programme schreiben etwa XML aus Platzgründen ohne „unnötige“ Leerzeichen, andere brechen die Zeilen im Interesse der Lesbarkeit wo es möglich ist um und rücken hierarchische Strukturen ein. Die Dokumente in beiden Darstellungen sind aber trotzdem funktional identisch.

[85] Sicherlich weicht das vom allgemeinen Konzept des „Originals“ ab. Wenn Authentizität gefordert wird, wie z. B. in Archiven, muß der Migrationsvorgang, der zur funktional identischen Kopie geführt hat, lückenlos dokumentiert und überprüfbar sein.

Unser Rechtssystem erkennt schon reduzierte Exemplare von Unterhaltungswerken (etwa von Audio-CDs kopierte und kodierte MP3-Dateien) als Kopien an und bestraft ihre unlizenzierte Verbreitung genauso wie die unlizenzierte Verbreitung der Originale. Dabei handelt es sich noch nicht einmal um funktional identische Kopien (auf Signalebene gibt es einen Unterschied, der für Menschen aber in der Regel nicht hörbar ist). Ich sehe also keinen Grund, warum funktional identische migrierte Kopien nicht als Stellvertreter für nicht mehr benutzbare Originale anerkannt werden könnten. (Natürlich müßten eventuelle Maßnahmen, die die Authentizität technisch sichern, auch angepaßt werden, etwa indem die digitale Signatur des Originals als eine Komponente der neuen Signatur der Kopie herangezogen wird.)

Videoaufzeichnungen und viele mehr. Wenn sich neue Formate ausbilden, müssen sie in der Regel alle Eigenschaften dieser verbreiteten Formate enthalten, da der Markt sie sonst nicht annimmt. Der Markt setzt wegen der bereits enormen Menge von Dateien in den genannten Formaten auch durch, daß gute, zuverlässige Konvertierungswerkzeuge erscheinen, die sich automatisiert betreiben lassen, wodurch die Kosten niedrig bleiben.

„Interaktive“ Dokumente und Anwendungen wie z. B. ein komplexeres und mit externen Daten gespeistes Tabellenkalkulationsdokument, ein auf CD oder DVD vertriebenes Multimedia-Lexikon oder ein Datenbanksystem lassen sich mit Hilfe der Migration – so wie mit keinem anderen der bisher vorgestellten Verfahren – nicht langfristig zugänglich halten (siehe auch Migrationsexperiment 7.2.3 auf Seite IX).

5.7 Emulation

In der Informatik bedeutet Emulation die „interpretative Implementierung des Operationsprinzips einer Rechnerarchitektur“ (Lexikon der Informatik und Datenverarbeitung, Oldenbourg, München, 1997), aber nicht nur Rechnerarchitekturen, sondern auch Software-Schnittstellen und Hardwarekomponenten können emuliert werden.

Die Emulation ist in der Informatik kein neues Konzept, und einzelne Anwendungen sind in verbreiteten Betriebssystemen enthalten: Apple MacOS X enthält einen Emulator für MacOS 9 (unter dem Namen „Classic“), damit (viele) alte Programme ausführbar bleiben; Microsoft Windows NT und seine Nachfolger emulieren das alte DOS und für DOS-basierte Programme sogar die früher verbreitete „Sound Blaster“-Hardware.

Jeff Rothenberg (in [Roth00] und anderen Veröffentlichungen) und viele andere empfehlen die Emulation oder Varianten davon (z. B. für eine virtuelle Maschine geschriebene Programme, vgl. [Lori01]) als Lösung der Probleme der Langzeitarchivierung. Da Computer mit Software emuliert werden können und der emulierte Computer auch ein anderes Emulatorprogramm ausführen kann, das dann wieder einen anderen Emulator ausführt usw., ist die Methode zumindest theoretisch bis zur Unendlichkeit (oder genauer bis zum ältesten zu emulierenden Computer) fortführbar:

> It may seem unwarranted to assume that future hardware will be able to emulate any previous (obsolete) hardware, but this assumption actually rests on quite firm ground. The logical operations performed by current digital computers are instances of a well-defined class of mathematical computations known (suggestively) as ‘computable functions’. The basic instructions executed by such computers are generally defined in terms of the simplest logical and arithmetic operations. Whatever kinds of unimagi-

> nable operations future computers perform (such as quantum, multi-state computations), it is almost inconceivable that they will be incapable of performing the simple logical operations that constitute the instruction sets of current (and past) computers. It therefore seems safe to assume that any conceivable future general-purpose computer will be able to perform the computations necessary to emulate any current computer.
>
> [Roth00, S. 27]

Am leichtesten[86] ist es, die Hardware einer Rechnerarchitektur zu emulieren, und die korrekte Implementierung der Emulation zu überprüfen. Jeder Computerprozessor hat einen genau dokumentierten, ziemlich eingeschränkten Befehlssatz. Zusätzlich zum Prozessor müssen in der Regel die auf der jeweiligen Architektur üblichen weiteren Hardwarekomponenten wie Festplatten, Grafikkarten, Ein- und Ausgabegeräte usw. emuliert werden, wobei es häufig reicht, *eine* verbreitete Komponente jeder Art (am besten eine, deren Spezifikation zur Verfügung steht) zu emulieren. Der Nachteil der Emulation von Hardware ist die geringe Geschwindigkeit des emulierten Systems: üblich ist eine Verlangsamung auf ein Hundertstel bis ein Zehntel der ursprünglichen Ausführungsgeschwindigkeit (bei sehr unterschiedlichen Prozessorarchitekturen eventuell sogar noch stärker). Das ist jedoch in der Langzeitarchivierung kein großes Problem, da zukünftige Computersysteme noch um ein Vielfaches schneller sein werden als heutige Computer.

Wenn derselbe Prozessor wie im physischen System emuliert werden soll, kann auf die komplette Emulation verzichtet werden; in diesem Fall werden nur einige Prozessorbefehle, vor allem die für die Ablaufsteuerung und den Speicherzugriff, emuliert, andere (z. B. Berechnungen) laufen auf dem Originalprozessor ab. Dieses Verfahren heißt „Virtualisierung“ und ist heute wegen der hohen Geschwindigkeit gegenüber der vollständigen Emulation in manchen Bereichen recht beliebt.

Schwieriger (und auch schwerer zu verifizieren) ist die Emulation eines Betriebssystems, da die Programmierschnittstellen heutiger Betriebssysteme sehr komplex sind[87] und sie sich auch häufiger ändern als eine Hardware-Architektur. Trotzdem wird auch das häufig versucht, weil die Ausführungsgeschwindigkeit auf diese Weise fast an das Original herankommen kann. Diese Vorgangsweise ist jedoch für die Langzeitarchivierung weniger interessant, da dort immer das Originalsystem erwünscht ist.

[86] Natürlich nur relativ gesehen. Ein Emulationsprogramm zu schreiben ist immer ein größeres Projekt, das sehr gute Kenntnisse sowohl des Ausgangssystems als auch der Zielumgebung erfordert.

[87] Vgl. [Roth00, S. 25]
Das WINE-Projekt (WINE Is Not an Emulator) arbeitet seit 1993 an der Emulation von Microsoft Windows auf PC-Unix-Betriebssystemen, und es gibt nach wie vor viele Windows-Programme, die nicht richtig oder gar nicht unter WINE funktionieren.

Im Folgenden möchte ich kurz, ohne Anspruch auf Vollständigkeit, einige typische Emulatoren vorstellen und ihre Eignung für die Langzeitarchivierung abschätzen.

5.7.1 Bochs[88]

Dieser Emulator läuft unter den meisten heute verwendeten Systemen (Windows, Mac-OS, GNU/Linux usw.) und emuliert einen vollständigen IBM-kompatiblen PC, auf dem alle verbreiteten PC-Betriebssysteme laufen. Da der Schwerpunkt auf der Portierbarkeit liegt, ist die Emulation unter Bochs relativ langsam. Bochs ist Open Source, kostenlos verfügbar und wird aktiv weiterentwickelt.

Bochs scheint ein geeigneter Kandidat für die zukünftige Emulation von IBM-kompatiblen PCs zu sein, da der portabel geschriebene offene Quellcode wahrscheinlich leicht an zukünftige Plattformen angepaßt werden kann.

5.7.2 QEMU[89]

Dies ist ein auf Geschwindigkeit optimierter Prozessor-Emulator, der die emulierten Befehle zur Laufzeit in den echten Befehlssatz des ausführenden physischen Systems übersetzt. Aus diesem Grund ist QEMU deutlich schneller als etwa Bochs (laut Homepage bis zu 65mal so schnell), aber schwieriger auf andere Systeme zu portieren. Es läuft auf vielen Architekturen und Betriebssystemen und kann IBM-PCs sowie (unvollständig) PowerPC-basierte Systeme emulieren, ist aber noch nicht so ausgereift und zuverlässig wie Bochs (für PC) und PearPC (für PowerPC).

Wenn QEMU weiterentwickelt wird und mehr Systeme zuverlässiger unterstützt als jetzt, kann es als wegen der größeren Geschwindigkeit auf verbreiteten Plattformen eine gute Alternative zu Bochs werden. Im jetzigen Zustand sollte es wegen seiner aus der Entwicklungsphase rührender gelegentlicher Instabilität nur mit gut getesteter emulierter Software eingesetzt werden.

5.7.3 PearPC[90]

PearPC ist ein ziemlich neues Open-Source-Projekt, es wurde erst 2003 begonnen. Es emuliert die PowerPC-Architektur unter Windows und GNU/Linux auf verschiedenen Computersystemen und ist mittlerweile in der Lage, das beliebte Betriebssystem MacOS X, das sonst nur auf Apple-Hardware läuft, auszuführen.

Durch das große Interesse an PearPC und wegen der (im Vergleich zu IBM-kompatiblen) geringen Variation der PowerPC-Hardware erscheint PearPC als sehr geeigneter

[88] bochs: The Open Source IA-32 Emulation Project http://bochs.sourceforge.net/
[89] QEMU CPU Emulator http://fabrice.bellard.free.fr/qemu/
[90] PearPC - PowerPC Architecture Emulator http://pearpc.sourceforge.net/

Kandidat für die zukünftige Emulation von PowerPC-basierten Computern.

5.7.4 Basilisk II[91]

Das auf Motorola 680x0-basierten Prozessoren laufende Apple-MacOS-Betriebssystem war in den 1990-er-Jahren ein häufig nachgefragtes Emulationsziel; es gab mehrere kommerzielle und Open-Source-Lösungen. Ein Beispiel ist der Open-Source-Emulator „Basilisk II“, der allerdings (so wie alle anderen alten Apple-Emulatoren) das Original-Basissystem (ROM) von Apple zum Funktionieren braucht.

Basilisk II läuft auf verschiedenen Plattformen und Betriebssystemen. Die Entwicklung scheint allerdings (auch wegen des Alters der emulierten Plattform) mehr oder weniger stehengeblieben zu sein.

Die zukünftige Verwendbarkeit aller Apple-680x0-Emulatoren erscheint etwas problematisch, da die ROM-Problematik urheberrechtliche Fragen aufwirft. Die Programme werden anscheinend wegen des geringen Interesses auch nicht mehr weiterentwickelt; allerdings ist das wegen der Möglichkeit, Emulatoren in anderen Emulatoren auszuführen auch kein so drastisches Problem.

5.7.5 SIMH[92]

Dieses Projekt entwickelt ein sehr portierbares, als Open Source verfügbares Emulationssystem, das diverse „historische“ (also mehr als 20 Jahre alte) Computerplattformen von zehn verschiedenen Herstellern emulieren kann (das SIMH-Projekt verwendet allerdings den Begriff „Simulation“ statt „Emulation“). Es läuft unter diversen Betriebssystemen und auf verschiedenen Plattformen. Die Ausführungsgeschwindigkeit spielt kaum eine Rolle, da die heutigen Computer um Größenordnungen schneller als die emulierten Systeme sind.

Ein interessantes Problem mit SIMH im Vergleich zu den anderen, hier vorgestellten Systemen ist die Tatsache, daß die emulierten Systeme so alt sind und damals von so wenigen Leuten benutzt wurden, daß das Wissen, sie zu bedienen, schwer aufzutreiben ist (das ist natürlich kein Fehler von SIMH, aber es erschwert die Arbeit mit den emulierten Systemen erheblich).

SIMH als beliebtes, aktives Open-Source-Projekt kann sicherlich eine wichtige Rolle in der zukünftigen Langzeitarchivierung spielen, wenn noch so historische Computer emuliert werden müssen.

[91] The Official Basilisk II Home Page http://www.uni-mainz.de/~bauec002/B2Main.html
[92] The Computer History Simulation Project http://simh.trailing-edge.com/

5.7.6 VMware[93]

Das kommerzielle VMware ist eine Virtualisierungslösung, keine komplette Emulation. Aus diesem Grund läuft es nur auf IBM-kompatiblen PCs unter GNU/Linux und Windows, und emuliert einen PC, auf dem wiederum die meisten PC-Betriebssysteme verwendbar sind. Für optimale Geschwindigkeit liefert VMware für die verbreitetsten Plattformen (GNU/Linux und Windows) eigene Treiber mit, die in den virtuellen Maschinen installiert werden können. Aber auch ohne diese Treiber ist VMware durch die Virtualisierung sehr schnell.

Kurz- und mittelfristig kann VMware oder ein vergleichbares Konkurrenzprodukt in vielen Bereichen (wenn PCs zur Verfügung stehen, auf denen PC-Software emuliert werden soll) eine geeignete Lösung für die Emulation sein. Wenn sich jedoch andere Prozessoren durchsetzen, wird es, da kein echter Emulator, wertlos.

5.7.7 Virtual PC[94]

Die Firma Connectix entwickelte lange Zeit einen beliebten PC-Emulator für Apple-MacOS-basierte Systeme. Microsoft kaufte 2003 Connectix und bietet jetzt zwei Nachfolgeprodukte an.

Virtual PC für Windows ist eine Virtualisierungslösung wie VMware. Der Nachteil gegenüber VMware ist, daß Virtual PC explizit nur Windows-Betriebssysteme unterstützt, Microsoft gibt keinerlei Garantien ab, daß andere Betriebssysteme funktionieren, und leistet keine Unterstützung dafür.

Virtual PC für Mac läuft auf MacOS und emuliert einen PC.

Beide Versionen sind kommerzielle Software. Wegen Microsofts Einstellung gegenüber konkurrierenden Betriebssystemen erscheint Virtual PC nicht als geeignete Lösung für die Zukunft.

5.7.8 Virtuelle Maschinen[95]

Neben echten Computerarchitekturen können auch „erfundene“ emuliert werden (vgl. [Bor+03, S. 64]). Es gibt einige Gründe, Architekturen zu erfinden; der legendäre Informatikprofessor Donald Knuth erschuf für Lehrzwecke einen eigenen, nur am Papier vorhandenen Prozessor (für den aber seitdem Emulatoren geschrieben wurden); au-

[93] VMware http://www.vmware.com/

[94] Virtual PC for Mac http://www.microsoft.com/mac/products/virtualpc/virtualpc.aspx
Microsoft Virtual PC 2004 http://www.microsoft.com/windows/virtualpc/default.mspx

[95] „Virtuelle Maschine“ wird in zwei Bedeutungen verwendet: Erstens heißen die Systeme, die in einem Emulator wie hier beschrieben ablaufen, virtuelle Maschinen; zweitens werden auch erfundene, nicht real existierende Computerarchitekturen so bezeichnet.

ßerdem gelten einige Beschränkungen der physischen Prozessortechnik nicht für die virtuellen Architekturen.

Virtuelle Maschinen wie die von Java und dem Microsoft .NET Runtime Environment dienen heute in erster Linie dazu, die Verteilung von Software (ohne mühsame Installation) zu vereinfachen und dieselbe Version der Software unter verschiedenen Betriebssystemen und auf verschiedenen Computerplattformen ablaufen zu lassen.

Um ein für die virtuelle Maschine geschriebenes Programm auszuführen, muß auf dem Zielsystem die Ablaufumgebung (virtuelle Maschine) installiert sein. In der Regel stellen die Hersteller ihre virtuelle Maschine im Interesse der größeren Verbreitung gratis zur Verfügung, allerdings sind die genannten Umgebungen Java und .NET nicht Open Source, es ist also nach wie vor vom Hersteller und seinen Interessen abhängig, für welche Plattformen die Ablaufumgebung erscheint. (Es gibt allerdings Open-Source-Projekte, die an virtuellen Maschinen für Java (Kaffe, GNU Classpath usw.) und .NET (Mono, DotGNU) arbeiten.)

Raymond A. Lorie schlägt in [Lori01] für Zwecke der Langzeitarchivierung bestimmter Dokumenttypen vor, eine neue virtuelle Maschine namens UVC (Universal Virtual Computer) zu entwerfen, die möglichst allgemein, aber gleichzeitig möglichst simpel ist. Dadurch könnte in Zukunft auf einer neuen Plattform „leicht“ eine virtuelle Maschine für den UVC programmiert werden. Allerdings müßten die Anzeigeprogramme für die Daten speziell für den UVC umgesetzt werden, um diese Methode in Zukunft verwenden zu können; dieser Ansatz benötigt daher die Kooperation der Software-Hersteller, oder er funktioniert nur bei hinreichend einfach gestalteten Dateien, für die ein UVC-Anzeigeprogramm geschrieben werden kann. Bei dieser Art von Dateien erscheint es mir aber sinnvoller, sie selbst zu migrieren als immer wieder die virtuelle Maschine für neue Systeme umzusetzen.

5.7.9 Emulation in der Langzeitverfügbarkeit

Heutige Emulatoren sind meistens so geschrieben, daß sie möglichst schnell ablaufen. Das bedingt eine gewisse Komplexität, die im simpelsten Fall, wenn es auf die Geschwindigkeit nicht ankäme, nicht notwendig wäre. Außerdem sind die Emulatoren dadurch mehr oder weniger an das System, auf dem sie laufen, gebunden.

Jeff Rothenberg schlägt in [Roth00, S. 40] einen anderen Ansatz vor: Statt Emulatoren für bestimmte Computer zu schreiben, sei es besser, die Computer in einer unabhängigen Sprache, einer „Emulatorspezifikation“ zu beschreiben. Wenn diese Sprache ausdrucksstark genug ist, um alle zu beschreibenden Computertypen abzudecken, kann mit der Spezifikation und einem passenden Interpreter-Programm genauso eine Emulation durchgeführt werden wie mit einem spezialisierten Emulator, nur die Ge-

schwindigkeit wird nicht so optimal sein. Aber wie wir gesehen haben, ist das für die Langzeitverfügbarkeit weniger relevant, solange die Computer jedes Jahr so viel schneller werden.

Es würde dann ausreichen, diese Emulatorspezifikationen aufzubewahren und auf neuen Computersystemen relativ einfach implementierbare Interpreter für sie zu programmieren. Wenn dann ein Glied der Kette nicht mehr praktikabel ist (weil z. B. die Emulatorspezifikationssprache obsolet wird), kann auf dem zukünftigen Computer immer noch ein herkömmlicher Emulator die alte Implementierung des Emulatorspezifikations-Interpreters und darin die spezifizierten alten Computer ausführen.[96]

Ein Emulator muß auch *modular* aufgebaut sein, um verschiedene Hardwarekonfigurationen abdecken zu können, wenn Programme diese brauchen. Manche Software ist nämlich hardware-abhängig geschrieben und funktioniert z. B. nur mit einer bestimmten Hardwarekomponente. Auf die heutigen Emulatoren trifft das kaum zu: sie implementieren meist nur eine Art der Hardware, z. B. einen Grafikkarten-Typ, ohne die Möglichkeit, eine andere Karte zu emulieren[97].

Eine mit der digitalen Langzeitarchivierung befaßte Institution mit einer auf Emulation basierenden Strategie müßte nach [Roth00, S. 50ff] mindestens folgende Schritte durchführen:

1. Wie bei jeder anderen Strategie auch natürlich die Bit-Ströme der Dateien und der Metadaten archivieren und zugänglich halten.

2. Die Metadaten ständig in Formaten aufbewahren und ggf. in neue Formate konvertieren, die zur jeweiligen Zeit mit wenig Aufwand verständlich bleiben. (Das ist also eine Art Migration, wenn auch nicht der Originaldokumente, sondern der Metadaten zur Langzeitverfügbarkeit und der Dokumentation des Emulationssystems.)

3. Für jedes Dokumentformat die notwendigen Umgebungen beschreiben und mit

[96] Rothenberg führt den Gedanken sogar noch eine Stufe weiter. Es wäre möglich, eine „virtuelle Maschine für Emulation“ (emulation virtual machine) zu schaffen und den Emulatorspezifikations-Interpreter für diese Maschine zu schreiben. Dann müßte nicht der Interpreter, sondern nur die virtuelle Maschine auf neue Computer portiert oder auf ihnen in einer herkömmlichen Emulation ausgeführt werden.

Es wäre dann möglich, die emulation virtual machine in der Spezifikationssprache zu beschreiben, wodurch sie automatisch in neueren Emulatoren ausführbar wäre.

Meiner Ansicht nach ist das durch die weitere Stufe ein unnötiger Zuwachs an Komplexität, da so und so etwas auf die neue Computerplattform übertragen werden muß. Der Aufwand, einen Interpreter zu übertragen, ist vergleichbar mit der Übertragung der virtuellen Maschine, bzw. meiner Meinung nach (basierend auf der Anzahl von Emulationsprojekten im Vergleich mit virtuellen Maschinen) sogar leichter.

[97] QEMU kann zwei verschiedene Grafikkartenmodelle emulieren; manche Programme funktionieren mit dem einen oder dem anderen besser. (Vgl. Experiment 7.3.3 auf Seite XIV)

den zur Verfügung stehenden Umgebungen (Software, -version, Betriebssystem, emulierte Plattform) verknüpfen.

4. Die Bitströme aller in der Organisation jemals verwendeten Emulationsspezifikations-Interpreter aufbewahren.

5. Wenn eine neue Emulatorspezifikationssprache erscheint, sicherstellen, daß ein neuer Interpreter für diese Sprache geschrieben wird, der auf aktuellen Systemen oder der virtuellen Maschine für Emulation läuft.

6. Wenn eine neue virtuelle Maschine für Emulation entwickelt wurde, sicherstellen, daß darauf ein Interpreter für die alten virtuellen Maschinen für Emulation geschrieben wird.

7. Sicherstellen, daß ein Interpreter für Emulatorspezifikationen oder eine virtuelle Maschine für Emulation auf jeweils aktuellen Rechnerplattformen läuft.

In der Theorie steht unter Befolgung all dieser Regeln zu jeder Zeit von jedem Dokument, das korrekt ins Archiv aufgenommen und mit den entsprechenden Metadaten zur notwendigen emulierten Umgebung versehen wurde, ein benutzbares Exemplar zur Verfügung.

In der Praxis gibt es noch nicht genug Erfahrung mit der Emulation, und es gibt einige Zweifel (vgl. etwa [Bor+03, S. 20] oder auch das Ergebnis meines Experiments mit Emulation alter Software, siehe Seite XIII), ob mit Hilfe der Emulation wirklich alles langfristig sicherbar ist. Allerdings hat auch noch niemand etwas Besseres vorgeschlagen, weswegen die Emulation wahrscheinlich als letztes Mittel der Langzeitarchivierung digitaler Daten übrigbleiben wird.

Emulation wie hier beschrieben ist eine komplexe Aufgabe, die erheblichen finanziellen und technischen Aufwand sowie gute Kenntnisse der aktuellen und veralteten Informationstechnologie voraussetzt. Dieser Aufwand ist höchstens einigen Archiven und Bibliotheken zumutbar, keinesfalls einem Privathaushalt[98]. Ein umfassender, auch in Privathaushalten und kleinen Firmen anwendbarer Ansatz muß wesentlich einfacher sein, um überhaupt eine Chance auf Akzeptanz zu haben.

[98] Oder den Enkelkindern, die Rothenberg selbst in [Roth95a] beschreibt:

> The year is 2045, and my grandchildren (as yet unborn) are exploring the attic of my house (...) They find a letter dated 1995 and a CD-ROM. (...) If I include all the relevant software on the disk, along with complete, easily decoded specifications for the required hardware, they should be able to generate an emulator to run the original software that will display my document. I wish them luck.

5.8 Ein kombinierter Ansatz für die Langzeitarchivierung

Wenn es nicht um institutionelle Archivierung mit gesetzlichen Vorschriften und der Verpflichtung, fremde Daten zur Aufbewahrung zu übernehmen, geht, kann eine pragmatischere Vorgangsweise verwendet werden, die selbst in Privathaushalten (eventuell unter gelegentlicher Einbindung einer erfahreneren Person) und kleinen Firmen anwendbar sein dürfte. In diesen Bereichen ist die Vielfalt der Dateiformate auch nicht übermäßig groß, und es ist abschätzbar, wie wichtig die jeweilige Information ist (es kann also durchaus auch entschieden werden, daß bestimmte Daten nicht mehr notwendig sind).

Der erste Schritt zur möglichst langfristigen Verfügbarkeit der Information ist die Verwendung von Standardformaten, wann immer es möglich ist. Wenn kein Standardformat zur Verfügung steht, sollte ein möglichst verbreitetes Format gewählt werden. Textverarbeitungsdokumente können etwa derzeit, wenn möglich, statt der proprietären Formate der Textverarbeitungen in einem standardisierten Format wie RTF oder XML gespeichert werden, und dazu wird am besten eine Kopie im PDF-Format mitgesichert. Durch die Verwendung von Standardformaten (und deren Verbreitung) steigt die Wahrscheinlichkeit, daß während der Zeit, in der das Dokument interessant ist, gar keine Migration oder Emulation notwendig wird, weil auch zukünftige Software das Format unterstützt.

Es muß natürlich sichergestellt werden, daß die Bitströme der Dateien verfügbar bleiben. Das ermöglichen Massenspeicher wie CD-R oder DVD-R, die regelmäßig (rechtzeitig vor Ablauf der angenommenen Lebensdauer) auf neuere Datenträger umkopiert werden. Natürlich muß das auch passieren, wenn es sich abzeichnet, daß die Datenträger selbst noch in Ordnung sind, aber die Lesegeräte vom Markt zu verschwinden drohen. Am besten werden von wichtigen Daten zwei Kopien an verschiedenen Orten aufbewahrt.

Ist ein Dateiformat so veraltet, daß gängige Programme es nicht mehr unterstützen, ist üblicherweise einige Jahre lang noch immer Software auffindbar, die das Format lesen und konvertieren kann. Im Idealfall kann der Konverter (oder ein anderer Konverter) das neue Format auch ins alte zurückkonvertieren, um die funktionale Identität der Dokumente vergleichen zu können. Wenn das nicht möglich ist, kann eventuell auf ein drittes, möglicherweise vereinfachtes Format zurückgegriffen werden (sowohl PostScript- als auch PDF-Dateien lassen sich z. B. in Bitmap-Bilder umwandeln, die dann verglichen werden können). In Firmen führen eventuell gesetzliche Aufbewahrungsfristen dazu, daß dieser Migrationsprozeß bestimmten Standards in Bezug auf Authentizität entspricht[99]. Im privaten Bereich kann auch die stichprobenweise opti-

[99] Da das Problem der digitalen Langzeitarchivierung heute ziemlich unbekannt ist, gibt es solche

sche Inspektion der konvertierten Inhalte genügen. Die Originaldaten sollten für eine eventuell später erforderliche emulationsbasierte Methode auch nach der Migration aufgehoben werden.

Nur wenn die Dokumente auch mit Hilfe der Migration nicht mehr zugänglich sind, muß auf Emulation ausgewichen werden, da diese Methode den meisten Aufwand verursacht. Dazu muß die ursprünglich verwendete Software auch aufgehoben (als Bitstrom bewahrt) werden. Bei Systemwechseln (die selten öfter als alle zehn Jahre vorkommen) sollte rechtzeitig ein Emulator für das alte System besorgt und getestet werden. Dieser Emulator muß nicht allen Empfehlungen von Rothenberg entsprechen, es genügt, wenn er ohne jede Emulatorspezifikation einfach nur das andere System emuliert. Da im beschriebenen einfachen Szenario die Bandbreite der verwendeten Software klein ist, sind nur wenige Emulatoren notwendig, und sie können immer in den neueren Emulatoren auf jeweils neueren Systemen ausgeführt werden. (Es ist natürlich kein Problem, wenn nur ein allgemeiner, emulatorspezifikation-basierter Emulator zur Verfügung steht, nur sind simplere Emulatoren wahrscheinlich einfacher bedienbar.) Es ist zweckmäßig, allgemeine Anleitungen zum Anzeigen der Information für jedes emulierte System auch aufzuheben, da die Bedienung der Betriebssysteme und Software sich über die Zeit ändern kann, und unsere heutigen Systeme sind dann in Zukunft vielleicht nicht mehr leicht verständlich.

5.9 Rechtliche Rahmenbedingungen

Die Langzeitverfügbarkeit digitaler Information wird aus mehreren Gründen stärker durch das Rechtssystem beeinflußt als bei herkömmlichen Informationsträgern. Das liegt u. A. an der schnelleren Alterung der Informationsstrukturen und -träger sowie daran, daß die fortgeschrittenen Methoden der Langzeitverfügbarkeit erheblich mehr rechtlich regulierte Handlungen (Kopieren, Verändern) beinhalten als selbst die aufwendigsten Verfahren der herkömmlichen Restaurierung.

5.9.1 Das Urheberrecht

Das Recht der Urheber „eigentümlicher geistiger Schöpfungen“ (§ 1 UrhG), über die Nutzung ihrer Werke zu bestimmen, heißt Urheberrecht[100]. Das Urheberrecht ist im

gesetzlich anerkannten Standards noch nicht, aber wenn das Problem akut wird, wird es sie geben müssen. Die Authentizität bei der Emulation ist nämlich auch nur gewährleistet, solange alle Komponenten der Emulationsumgebung und der emulierten Software authentisch sind, was ebenfalls entsprechende Standards und Kontrollmöglichkeiten voraussetzt.

[100] In angelsächsischen Ländern „copyright“ genannt.

Es gibt feine Unterschiede zwischen dem kontinentaleuropäischen und dem angelsächsischen Verständnis von Urheberrecht; in Europa zählen die Persönlichkeitsrecht-Aspekte etwas mehr, in

Laufe der Entwicklung neuer Technologien und Nutzungsarten ständig angepaßt worden, es ist heute ziemlich komplex und erfaßt auch neue Bereiche, für die es früher nicht zuständig war.

Das Urheberrecht wird heute in der Regel nicht von den Urhebern selbst umgesetzt, sondern sie treten alle oder bestimmte Rechte im Austausch für einen Anteil an den finanziellen Erlösen an spezialisierte Firmen wie Verlage o. Ä. ab, die dann am Markt die Interessen der Urheber vertreten. Bestimmte Arten der künstlerischen Arbeit wie Filme oder Musikaufnahmen sind wegen der Anzahl der mitarbeitenden Leute so komplex, daß die Rechte am entstandenen Werk überhaupt nur gemeinsam sinnvoll vertreten werden können (vgl. [Wand02, S. 3]).

Das Urheberrecht hat traditionell vor allem die Vervielfältigung, Veränderung und die öffentliche Darbietung von Werken geregelt. Diese Rechte (und je nach Art des Werks manche andere) waren (und sind nach wie vor) exklusiv dem Rechteinhaber (Urheber oder dessen bevollmächtigtem Vertreter) vorbehalten, der sie unter ihm genehmen Bedingungen an Andere weitergeben konnte. Die Dauer des Schutzes beträgt heute in den meisten Industrieländern 70 Jahre nach dem Tod des Urhebers (§ 60 UrhG) für die meisten Werkarten (früher war die Frist kürzer). Nach Ablauf dieser Periode werden die Werke „gemeinfrei“ (*public domain*), es bestehen keine Exklusivrechte mehr an ihnen[101].

Alle vorgestellten Verfahren der Langzeitarchivierung basieren darauf, daß die Daten regelmäßig, deutlich häufiger als es der Ablauf des Urheberrechts ermöglichen würde, umkopiert werden. Genau das Kopieren ist jedoch im Urheberrecht das am strengsten und weitesten geregelte Verfahren. „Der Öffentlichkeit zugängliche Einrichtungen, die Werkstücke sammeln“ (also z. B. Archive und Bibliotheken) dürfen unter bestimmten Voraussetzungen eine einzige oder einige wenige Kopien herstellen (§ 42 (7) UrhG), wenn das auf Papier „oder einem ähnlichen Träger“ geschieht. Für digitale Datenträger gelten weitere Einschränkungen.

Die Migration geht notwendigerweise mit einer Veränderung (Bearbeitung) der Werke einher. Diese Handlungen sind vom Urheberrecht in der Regel nur mit Zustimmung der Rechteinhaber erlaubt.

---

den USA die Verwertungsrechte (vgl. [Wand02, S. 4]). Wegen internationaler Vereinbarungen sind aber die wichtigsten und für den Alltag relevantesten Bestimmungen gleich oder sehr ähnlich.

[101] In der Realität kann das bei manchen Arten von Werken komplizierter sein. Z. B. beinhaltet eine Musik-Aufnahme Rechte des Komponisten/der Komponistin, der aufführenden KünstlerInnen und am technischen Aufnahmevorgang. Die Aufnahme ist erst dann nicht mehr von Exklusivrechten belegt (also z. B. frei kopierbar), wenn all diese (unterschiedlich langen) Schutzfristen abgelaufen sind.

Wie in den vorangegangenen Kapiteln gezeigt ist es relativ unwahrscheinlich, daß eine CD noch benutzbar ist, wenn alle an ihr und der enthaltenen Musik bestehenden Schutzrechte abgelaufen sind.

Für die Emulation ist das Aufbewahren der Originalsoftware notwendig. In der Praxis werden virtuelle Abbilder (*images*) von Festplatten (oder anderen, zur jeweiligen Zeit üblichen Speichermedien) angefertigt und diese dem Emulator gegeben. Es wird häufig notwendig sein, dasselbe Softwarepaket (z. B. Betriebssysteme und andere Standardsoftware) in mehreren Images zu installieren. Für Software ist das Urheberrecht jedoch strenger als für andere Inhalte, ein Exemplar darf nicht mehrmals installiert werden (außer wenn das explizit erlaubt wurde, wie z. B. bei Open-Source-Software). Dadurch steigen die Kosten für die Archive und Bibliotheken an, weil sie mehrere Lizenzen kaufen müssen, oder die Komplexität, wenn die Images dynamisch zusammengefügt werden, um dem Buchstaben des Gesetzes zu entsprechen (was grundsätzlich möglich, aber ziemlich aufwendig ist, vgl. [Bor+03, S. 69]).

Eine neue Richtung hat das Urheberrecht genommen, als mit der Entwicklung und weiten Verbreitung digitaler Aufnahme- und Speichertechnologien das verstärkte Kopieren in Originalqualität möglich wurde. (Die Qualität von Digitalkopien nimmt im Gegensatz zu Analogkopien nicht ab, siehe Kap. 3.1 auf Seite 18.) Die Rechteinhaber reagierten mit massivem Lobbying für neue, verschärfte Urheberrechtsgesetze und der Einführung sogenannter Kopierschutz-Technologien (siehe Kap. 5.10.1 auf Seite 107). Da diese Technologien aber in der Regel auf herkömmlichen Computersystemen nicht sehr wirksam sind (um die geschützten Inhalte nutzen zu können, müssen sie schließlich irgendwie in einen ungeschützten, für die Menschen zugänglichen Zustand gebracht werden), sind sie in vielen Fällen recht schnell „aufgebrochen“ worden. Aus diesem Grund wurde auf Betreiben der Rechteinhaber das Urheberrechtsgesetz erweitert[102]; es enthält jetzt ein Verbot von Technologien, die geeignet sind, den wirksamen Schutz des Urheberrechts von Werken zu entfernen. Solche Technologien, früher komplett legal, dürfen heute nicht mehr entwickelt, benutzt oder in Verkehr gebracht werden (§ 90c UrhG). Dieser gesetzliche Schutz kennt keinerlei Ablauffristen, die sonst im Urheberrecht üblich sind. Für die Umgehung von Kopierschutzmaßnahmen und die Herstellung geeigneter Werkzeuge sind auch Gefängnisstrafen vorgesehen (§ 91 UrhG).

Das führt dazu, daß selbst wenn Kopieren und Verändern (im Zuge der Migration) durch gesetzliche Ausnahmen im Interesse der Langzeitverfügbarkeit erlaubt wären, die Daten trotzdem nicht legal kopiert werden könnten, weil die Werkzeuge dafür fehlen, nicht legal beschafft werden können und/oder ihre Anwendung (unabhängig vom Zweck) strafbar ist.

[102] In den USA: Digital Millenium Copyright Act, 1998.
In der EU: Richtlinie 2001/29/EG des Europäischen Parlaments und des Rates zur Harmonisierung bestimmter Aspekte des Urheberrechts und der verwandten Schutzrechte in der Informationsgesellschaft.
In Österreich wurde die EU-Richtlinie Ende 2003 umgesetzt.

5.9.2 Das Patentrecht

Patente sind Exklusivrechte an Erfindungen; sie sollen die Kreativität und die Entwicklung neuer Methoden und Produkte und die Veröffentlichung der Innovation fördern, indem sie für eine beschränkte Zeit (meistens 20 Jahre) dem Patentinhaber (im Austausch für die Zahlung von Patentgebühren und die detaillierte, veröffentlichte Beschreibung der Erfindung) alleinige Verwertungsrechte an der kommerziellen Nutzung der Erfindung geben.

Lange Zeit wurden mathematische Algorithmen und reine Ideen ohne technische Umsetzung nicht als patentierfähig angesehen, da mathematische Zusammenhänge eher nur „entdeckt“ als „erfunden“ werden können. Außerdem ist Software bereits durch das Urheberrecht geschützt (als „Sprachwerk“, aber mit einigen Ausnahmeregelungen, die strenger sind als z. B. die Regelungen für Bücher). In den USA werden jedoch schon seit mehr als einem Jahrzehnt auch Patente auf Algorithmen und Geschäftsprozesse vergeben. Das EU-Parlament hat sich im Herbst 2003 gegen die Vergabe von Softwarepatenten ausgesprochen, die EU-Kommission hat diese Entscheidung aber im Mai 2004 rückgängig gemacht. Es ist daher noch unklar, ob in der Europäischen Union auch Softwarepatente möglich werden oder nicht. Einige tausend solcher Patente sind aber auch in der EU bereits eingetragen. Angeblich wird jedoch in der EU die neue Patentrichtlinie so gestaltet sein, daß sie die „Interoperabilität“ (Zusammenarbeit zwischen verschiedenen Produkten) nicht gefährdet.

Patente auf Algorithmen oder Dateiformate können bestimmte Verarbeitungsschritte so für andere blockieren, daß diese nur mit Zustimmung des Patentinhabers durchführbar sind. Auf diese Weise wurde bereits die Verbreitung von Konvertiersoftware für das ASF-Dateiformat unterbunden (siehe Fußnote auf Seite 50).

Softwarepatente können die unabhängige Entwicklung von Konvertier- und Migrationssoftware verhindern, wenn Interoperabilität nicht speziell von ihrer Wirkung ausgenommen ist. Patente laufen zwar nach 20 Jahren ab, aber diese Zeitspanne ist im Vergleich zur Entwicklung der Technologie so lang, daß es fraglich erscheint, ob der Markt nach 20 Jahren noch an Konvertiersoftware interessiert ist. Dieses Problem dürfte vor allem die Migration als Methode der Langzeitverfügbarkeit treffen; die Emulation, die ja die Verwendung der Originalsoftware in ihrer ursprünglichen Umgebung ermöglicht, sollte von Patenten weniger betroffen sein.

5.9.3 Lizenzvereinbarungen

Bei der Installation mancher Software müssen Lizenzvereinbarungen akzeptiert werden, das Programm läßt sich sonst nicht installieren (vgl. [Miel04, S. 217]). Nach dem Willen der Software-Anbieter ist die Nutzung von Software kein Kauf, sondern wird

nur mit zusätzlichen Einschränkungen erlaubt (lizenziert), dadurch kommt ein Nutzungsvertrag zustande. Die Lizenzvereinbarungen schließen manche Handlungen der NutzerInnen, die vom Gesetz her erlaubt wären, aus, z. B. das Recht auf Zurückentwicklung (*reverse engineering*, die Untersuchung der Arbeitsweise der Software). Die Lizenzvereinbarungen werden in der Regel auf unbeschränkte Zeit abgeschlossen, die normale Ablaufdauer des Urheberrechts greift nicht.

In manchen Lizenzvereinbarungen werden Dinge gefordert, die nichts mit dem Erwerb der Software zu tun haben, z. B. daß eine gratis verbreitete Zusatzsoftware nur mit einer bestimmten Version einer kostenpflichtigen Software zusammen eingesetzt werden darf, oder daß bestimmte Nutzungsarten der Software ausgeschlossen sind (vgl. [Miel04, S. 217]).

Einige Klauseln von Lizenzvereinbarungen wurden bereits von Gerichten für ungültig erklärt oder sind im Urheberrechtsgesetz explizit ausgeschlossen[103], aber die Vereinbarungen können durchaus noch Klauseln enthalten, die (bewußt oder unbewußt) die Migration oder (wahrscheinlicher) die Emulation unmöglich machen können.

Die wichtigste, häufige Einschränkung der Nutzung dürfte das Verbot der gleichzeitigen Ausführung derselben Software sein. Wenn sich die Institution daran halten will, muß sie bei der Emulation sicherstellen, daß jedes emulierte Exemplar der Software nur so häufig ausgeführt wird wie Lizenzen vorhanden sind. Das kann dazu führen, daß bei gleichzeitiger Nutzung manche BenutzerInnen warten müssen, bis eine Lizenz frei wird.

5.9.4 Pflichtexemplar u. dgl.

Bestimmte Bibliotheken in jedem Land oder Region (in Österreich: die Nationalbibliothek) haben ein gesetzlich vorgeschriebenes Recht auf die Ablieferung eines oder mehrerer Exemplare aller in ihrem Bereich veröffentlichten Werke. Dies wird als Pflichtexemplar bezeichnet und soll die komplette Dokumentation der publizistischen Tätigkeit des Landes oder der Region erleichtern.

Bei elektronischen Publikationen ist es unklar, wie so ein Pflichtexemplar aussehen soll. Bei Publikationen auf Datenträgern kann ja der Datenträger abgeliefert werden, aber selbst in diesem Fall ergeben sich eventuell Probleme mit dem Fehlen der üblichen Merkmale von Veröffentlichungen (z. B. ISBN-Nummer, Autorenangaben usw.).

Wenn jedoch gar kein Datenträger vorhanden ist, wie im Falle von Publikationen im World Wide Web, ist die Situation noch komplizierter. Wie soll die Pflichtexemplarbibliothek überhaupt von der Veröffentlichung erfahren? Ab wann gilt das Werk

[103] Z. B. ist reverse engineering für Interoperabilität erlaubt, und auf das Recht kann gar nicht wirksam verzichtet werden (§ 40e UrhG).

als veröffentlicht? (Es ist häufig leichter, elektronische Publikationen nach und nach aufzubauen, in diesem Fall steht das Werk zuerst nur unvollständig im Web.) Was fällt überhaupt unter das Pflichtexemplargesetz? (Soll also z. B. jede ins Web gestellte Hausübung, Kinderzeichnung und „Das-ist-meine-Katze“-Homepage in die Bibliothek aufgenommen werden?) Was passiert mit dynamischen Webseiten, etwa Online-Diskussionsforen, deren Inhalt sich mehrmals am Tag ändern kann?

Eine Pflichtexemplarbibliothek muß sich auf jeden Fall auf den Empfang digitaler „Pflichtexemplare“ vorbereiten. Sie muß für Datenträger entsprechende Lesegeräte anschaffen und die Software-Umgebung für die Anzeige der Daten installieren. Wenn die „Ablieferung“ des Werks nur aus der Bekanntgabe einer Internet-Adresse besteht, muß die Bibliothek selbst feststellen, welche Dateiformate verwendet werden und welche Umgebung daher für das Ansehen notwendig ist. Sie muß dann das Werk auch mit Metadaten versehen und bei sich abspeichern, um es in Zukunft selbst mit geeigneten Methoden der Langzeitverfügbarkeit zugänglich zu halten. Dies kann jedoch bei dynamischen Webseiten sehr schwierig oder unmöglich sein. Wegen der Menge der Daten muß der ganze Prozeß möglichst automatisch ablaufen. Es gibt bereits erste Ansätze für die Komplett-Archivierung des World Wide Web[104] oder von Ausschnitten[105], sie können jedoch aus den genannten Gründen noch nicht so vollständig und geordnet sein wie die Bestände einer Pflichtexemplarbibliothek.

Kopierschutzmaßnahmen und andere Nutzungsbeschränkungen können Probleme für die Langzeitarchivierung verursachen. Solange es einer Pflichtexemplarbibliothek nicht explizit erlaubt wird, solche Maßnahmen zu umgehen und lizenzrechtliche Einschränkungen zu ignorieren, muß sie bei der Abgabe der Pflichtexemplare darauf bestehen, Exemplare ohne Zugangshindernisse zu erhalten. Dies kann für die publizierenden Stellen erhebliche Mehrkosten verursachen, sie werden sich daher ohne entsprechende gesetzliche Verpflichtungen wahrscheinlich eher weigern. (Außerdem sind die meisten Inhaltsanbieter stark daran interessiert, daß gar keine „ungeschützten“ Exemplare ihrer Werke in Umlauf kommen.) Gesetzlich ist eine Zurverfügungstellung von ungeschützten Exemplaren jedenfalls nicht vorgesehen, im Gegenteil: laut § 44 MedienG „genügt die Ablieferung oder Übermittlung von Stücken der vom Hersteller ausgelieferten Art“.

[104] Internet Archive http://www.archive.org/
[105] z. B. für Österreich: Austrian On-Line Archive http://www.ifs.tuwien.ac.at/~aola/

5.10 Probleme mit den Methoden der Langzeitarchivierung

5.10.1 Kopierschutztechnologien

Kopierschutztechnologien (auch *DRM*, *digital rights management* oder *digital restrictions management* genannt) und ähnliche Nutzungsbeschränkungen könnten sich zukünftig als großes Problem der Langzeitarchivierung erweisen.

Frühere Kopierschutzmaßnahmen arbeiteten häufig unter Ausnutzung von unvorhergesehenen Aspekten der Technologie. Zum Beispiel wurden auf Originaldisketten unlesbare Sektoren erzeugt, und die Programme prüften, ob der Datenträger an dieser Stelle lesbar ist. Wenn ja, mußte es sich um eine Kopie handeln. Audio-CDs verstoßen heute häufig ein bißchen gegen die Spezifikation, um das Auslesen in Computern zu verhindern oder zu erschweren (siehe Kap. 3.3.3 auf Seite 24). Bei solchen Verfahren ist es schwer bis unmöglich, an den Datenstrom überhaupt heranzukommen, und Manipulationen, die dazu notwendig wären, können seit 2003 mit Freiheitsstrafen belegt sein. Ein Archiv oder eine Bibliothek kann/darf also die Informationen von solchen Datenträgern in der Regel nicht kopieren.

Es ist auch seit längerer Zeit üblich, die Nutzung einer Software an eine bestimmte Hardware (häufig als „Dongle“ bezeichnet) zu binden. Die Hardware wird in der Regel mit der Software ausgeliefert und muß an eine der Schnittstellen des Computers angesteckt werden (z. B. an die Druckerschnittstelle oder in neueren Zeiten an USB). Die Software läuft nur, wenn sie die Hardware findet. Eine solche Hardware zu emulieren dürfte technisch schwierig sein und es verstößt auch mit ziemlicher Sicherheit gegen das Verbot der Umgehung technischer Maßnahmen in den aktuellen Urheberrechtsgesetzen. Dongles werden vor allem bei teurer Spezialsoftware verwendet, die enthaltenen Daten sind deswegen manchmal ziemlich wertvoll; sie sind jedoch häufig in speziellen Dateiformaten gespeichert, was auch die Migration erschweren dürfte. In meinem Experiment mit einer Dongle-geschützten Software waren alle fünf Emulatorprogramme nicht in der Lage, den Zugriff auf das Original-Dongle so zu ermöglichen, daß das Programm es erkennt und richtig abläuft (siehe Experiment 7.3 auf Seite XIII).

Online legal vertriebene Dateien, die Unterhaltungsinhalte wie Musik und Filme enthalten, sind heute in der Regel verschlüsselt. Der notwendige Entschlüsselungscode ist in einem dazugehörigen Abspielprogramm enthalten. Das Abspielprogramm bekommt oder generiert während der Installation eine weltweit eindeutige Identifizierungsnummer o. Ä.; die Dateien werden für diese Nummer verschlüsselt. Das heißt, daß die Daten ausschließlich mit diesem Programm auf diesem Computer abspielbar sind, auf einem anderen Computer oder für andere Programme sind sie nur sinnlose Anhäufungen von Bits. Die Abspielsoftware kann auch zusätzliche Nutzungsregeln, wie eine Einschrän-

kung der Nutzungshäufigkeit oder Nutzungszeit durchsetzen, die Regeln sind kodiert in den Dateien enthalten. Auch hier ist die Erstellung oder Nutzung von Werkzeugen, die den Kopierschutz entfernen könnten, verboten. Die Migration der Daten oder auch nur die Konvertierung in offene Formate ist daher unmöglich, die Emulation wird verkompliziert oder unmöglich gemacht.

Wie in Kap. 4.4.3.1 auf Seite 74 beschrieben, kann Verschlüsselung in Verbindung mit einem zentralen Server in Zukunft sogar für eigene Dokumente verwendet werden. Für das Öffnen dieser Dokumente ist dann nicht mehr nur die geeignete Software, sondern auch eine korrekte Benutzer-Identifizierung und auch die Verfügbarkeit des zentralen Verschlüsselungsservers notwendig. So „geschützte“ Dokumente sind in Zukunft wahrscheinlich wertlos, wenn die Server nicht mehr in der selben Form vorhanden sind, die Technologie sich ändert oder notwendige Login- und Paßwortdaten verloren gehen.

Bei einigen Produkten wie dem Betriebssystem Windows XP oder sog. e-books (spezielle Computer zum Lesen digital verbreiteter Bücher) ist die Nutzung mit einer Online-Registrierung gekoppelt. Die Software oder die Inhalte lassen sich nur nach einer automatischen Freigabe des Herstellers über das Internet oder durch Eingabe eines telefonisch erfragten Codes nutzen. Dies kann für die Langzeitarchivierung fatal sein, da keineswegs gesichert ist, daß ein Hersteller in einigen Jahrzehnten noch existiert, das Produkt betreut oder daß überhaupt eine Form des Internet, wie wir es kennen, nutzbar ist. Bei solchen Produkten dürfte es sehr schwierig werden, sie in irgendeiner Form langfristig zu benutzen. (Einige e-book-Anbieter sind auch bereits in Konkurs gegangen oder haben sich andere Betätigungsfelder gesucht, da der Markt für digitale Bücher sich – vielleicht gerade wegen der drakonischen Kopierschutzmaßnahmen – nicht so entwickelte wie sie es gehofft hatten; vgl. [Rink03]. Die BenutzerInnen solcher Systeme sind nun mit den in ihren Lesegeräten gespeicherten Inhalten allein geblieben, ohne die Möglichkeit, die gekauften Inhalte bei Veralten oder Schäden des Geräts anderswo zu nutzen.)

Die Inhaltsanbieter (wie die Musik- und Filmindustrie), zu denen auch Teile der Software-Industrie zählen, drängen darauf, daß Kopierschutz möglichst umfassend in die grundlegende Architektur von Computern eingebaut wird (vgl. etwa [Himm04]). Sie bezeichnen das als „*trusted computing*“. Nach den Vorstellungen würde jede Komponente der Computer sich „ausweisen“ müssen, um mit urheberrechtlich geschützten Inhalten in Berührung kommen zu dürfen, und müßte bestimmte Kopierschutzmaßnahmen durchsetzen. Die Soundkarte würde sich etwa weigern, Musikstücke ohne gültige digitale Unterschrift eines Rechteinhabers abzuspielen. In einem solchen Konzept ist Emulation nicht nur nicht vorgesehen, sie wird sogar im Interesse des Schutzes der Urheberrechte explizit unmöglich gemacht. Migration dürfte auch stark erschwert

oder unmöglich werden, da die Hersteller der „trusted“ Abspielprogramme nicht daran interessiert sind, daß die Dateien in andere Formate umgewandelt werden können.

Ein interessanter Spezialfall ist der der wissenschaftlichen Journale. Da in diesem Bereich die Preise ständig stark steigen, während die Etats der wissenschaftlichen Bibliotheken stagnieren oder abnehmen (sog. *Zeitschriftenkrise*, vgl. etwa [Weyh00, S. 14ff]), können die Bibliotheken bestimmte Periodika nur mehr in elektronischer Form „kaufen“. Sie kaufen dann jedoch nicht die Zeitschrift selbst, sondern nur Zugang dazu (vgl. [Weyh00, S. 59]), den der Verlag, der das Journal herausgibt, z. B. mit Zugangs- und Nutzungsbeschränkungen (z. B.: Drucken und Kopieren sind nicht möglich) auf einem Server im World Wide Web realisiert. Wenn der Vertrag abläuft oder die Bibliothek die Zeitschrift abbestellen muß, kann es passieren, daß der Zugang zu den *alten*, bereits bezahlten Ausgaben der Zeitschrift verloren geht. (Dies läßt sich jedoch durch eine geeignete Vertragsgestaltung vermeiden.) Im Vergleich zu Papier-Ausgaben der Journale ist der Zugang trotzdem minderwertig: wenn der Verlag in Konkurs geht oder aus einem anderen Grund den Zugriff nicht mehr anbietet, oder wenn die verwendeten Zugangsmethoden und Dateiformate veralten, hat die Bibliothek nicht die gleichen Reaktionsmöglichkeiten wie bei physisch vorhandenen, im Eigentum der Einrichtung befindlichen, auf relativ stabiles Papier gedruckten echten Zeitschriften (vgl. [Weyh00, S. 58]).

5.10.2 Notwendiges Wissen und veränderte Bedienung

Um offene, standardisierte Formate verwenden zu können, ist es zuerst einmal notwendig, über die Vor- und Nachteile der zur Auswahl stehenden Formate Bescheid zu wissen. Dies dürfte die meisten Gelegenheits-AnwenderInnen schon überfordern, da sie gewohnt sind, das Format zu akzeptieren, das die verwendete Software vorschlägt, und sich höchstens dann mit der Problematik beschäftigen, wenn sie Schwierigkeiten beim Datenaustausch haben.

Für die Migration ist die Kenntnis über viele verschiedene, historische und aktuelle Dateiformate und geeignete Konvertierungswerkzeuge unerläßlich. Die im Alltag verwendeten Programme sind in der Regel nicht auf die automatische, überprüfbare Konvertierung großer Datei- und Datenmengen vorbereitet, dies ist aber genau die Vorgehensweise bei der Migration, wenn sie mit akzeptablem Aufwand geschehen soll.

Da bei der Emulation die häufig längst obsolet gewordene Original-Hardware emuliert und die Original-Software benutzt werden muß, ergeben sich Probleme bei der Bedienung. Die befehlsbasierten Benutzerschnittstellen der früheren Betriebssysteme und Programme sind alles andere als intuitiv, und häufig ist keine online verfügbare Einführung enthalten. Moderne grafische Benutzeroberflächen kommen uns intuitiv

vor, aber auch sie enthalten eine Menge Elemente unseres *heutigen* Verständnisses der Computerbedienung. Es kann also durchaus sein, daß viele unserer heutigen Konzepte der Maus- und Tastaturbedienung zweidimensionaler „Fenster“ in fünfzig Jahren nicht mehr allgemein bekannt sein werden. Auch die Anleitungen und Handbücher der früheren und heutigen Software benutzen die jeweils übliche Terminologie und entsprechen den jeweils üblichen Erwartungen; sie können sich als ähnlich nutzlos erweisen wie die Handbücher der ersten Computer (die ausschließlich mit Schaltern programmiert wurden) für uns wären. Wenn im Laufe der Zeit mehrere Emulationsschichten notwendig werden, muß zumindest zum Starten der jeweiligen Emulation und des Anzeigeprogramms in der „innersten“ Emulation das Wissen der geeigneten Vorgehensweise vorhanden sein.

Die für die Bedienung notwendige Hardware kann sich langsam ändern. Z. B. wurde die in den 1980-er-Jahren sehr beliebte Heimcomputerplattform Commodore 64 üblicherweise mit Tastatur und Steuerknüppel (*joystick*) bedient. Heute sind Mäuse weitaus üblicher als Joysticks, sie sind jedoch kein wirklicher Ersatz. In Zukunft könnte Spracheingabe die Tastaturbedienung ersetzen.

Die angesprochenen Kenntnisse lassen sich natürlich durchaus aneignen. Die Thematik hat jedoch eine gewisse Komplexität (allein die Bedienung des früher viel verwendeten Betriebssystems DOS füllt Bücher), und sie spielen außer für die Nutzung alter Informationen in keinem Bereich des täglichen Lebens eine Rolle; die Bereitschaft, sie zu lernen, dürfte bei PrivatanwenderInnen also eher gering sein. Wenn Bibliotheken und Archive in großem Maßstab digitale Daten anbieten werden, müssen sie große Mengen von Computern für die Benutzung (samt geeigneter Emulations- und Anzeigesoftware) kaufen und die MitarbeiterInnen werden mit der Einschulung und Unterstützung der BenutzerInnen ziemlich beschäftigt sein.

5.10.3 Lösungsansätze für die Probleme

In erster Linie müßte ein Problembewußtsein geschaffen werden. Da das Thema der Langzeitarchivierung digitaler Daten als Schnittmenge von Archiv- und Computerwissenschaft derzeit als eher langweilig gilt, müssen wohl größere Mengen von Informationen ziemlich spektakulär und von den Medien beachtet verloren gehen, bis ein öffentliches Bewußtsein für die Problematik entsteht.

Mit öffentlichem Bewußtsein, daß die Menschheit ihr kulturelles Erbe verlieren könnte, würden keine Gesetze wie das Verbot der Umgehung von Kopierschutzmaßnahmen entstehen, oder ihre Auswirkungen würden zumindest für legitime Zwecke der Archivierung gemildert. Das Bewußtsein würde auch auf der Nachfrageseite des Marktes helfen, zu kurzlebige und geschlossene Datenträger und Dateiformate zu vermeiden.

Hersteller kopiergeschützter Programme und Inhalte müßten gesetzlich verpflichtet werden, bei legitimem Bedarf (auch blinde Menschen sind z. B. daran angewiesen, daß sie Texte mit Braille-Terminals lesen können; dazu müssen die Dateien im Klartext vorliegen) ungeschützte Kopien anzubieten und bei geeigneten Archiven und Pflichtexemplarbibliotheken zu deponieren. Die Pflichtexemplarregelung sollte auf Software ausgebreitet werden (derzeit sind nur mit Medieninhalten vergleichbare elektronische Produkte erfaßt, reine Anwendungssoftware nicht). Die Software müßte den Archiven ohne Kopierschutz, einschränkende Lizenzverträge und Laufzeitbeschränkungen (z. B. zeitbezogen oder auf eine bestimmte Hardware angewiesen) zur Verfügung gestellt werden. Wenn die Inhalte und die Software nicht in geeigneter Form zugänglich sind, müßten Archive und Bibliotheken im Interesse der Langzeitsicherung auch ungestraft den Kopierschutz umgehen können. Herstellung und Verbreitung der erforderlichen Werkzeuge müßten erlaubt sein, sie könnten ähnlich wie Schußwaffen und Medikamente reguliert werden.

Softwarepatente sollten, wenn möglich, gar nicht erlaubt sein. Falls sie in Europa eingeführt werden, müßten sie (und verwandte EDV-Patente) im Interesse der Interoperabilität möglichst eindeutige und weite Ausnahmeregelungen haben.

Bei einer gewissen Verbreitung eines Dateiformats liegt es im öffentlichen Interesse, daß das Format nicht zur Einbahnstraße für Daten wird und nur die Monopolbestrebungen des Herstellers stärkt. Es erscheint daher sinnvoll, wenn gesetzliche Regelungen eingeführt werden, die die Hersteller verbreiteter Dateiformate verpflichten, kostenlose Anzeige- und Konvertiersoftware ohne Zeit- und anderweitige Beschränkungen anzubieten.

Im EDV-Unterricht in der Schule sollte nicht nur mit einem einzigen Betriebssystem und einer Art von Software gearbeitet werden, sondern wenn möglich mit unterschiedlichen Bedienungskonzepten. Der Unterricht wäre auch geeignet, den in Zukunft wahrscheinlich wichtiger werdenden Umgang mit Emulatoren und Migrationsprogrammen zu demonstrieren und die digitale Langzeitverfügbarkeit als Beitrag zur Bewahrung der Kultur der Menschheit zu darzustellen.

5.11 Zukunftsaussichten

5.11.1 Die Verbreitung von Open-Source-Software

Über die letzten zehn Jahre hat Open-Source-Software eine merkbare und stärker werdende Marktpräsenz erreicht. Gleichzeitig hat das Internet offene Standards und gemeinsam erarbeitete Kommunikationsprotokolle gefördert. Die beiden Trends zusammen dürften eine gewisse Entwicklung hin zu offenen, besser dokumentierten und da-

durch von mehr Software unterstützten Dateiformaten ergeben. Dadurch könnte in Zukunft die Migration leichter werden als heute, und Emulation sollte seltener notwendig werden, weil praxiserprobte, offene Formate mit großer Verbreitung weniger schnell „veralten“. Die Voraussetzung dafür, daß diese Trends halten, ist jedoch, daß gewisse rechtskräftig verurteilte Quasi-Monopolisten unter den Software-Anbietern es nicht schaffen, zentrale Internet-Technologien in den eigenen Einflußbereich zu bringen.

Open Source bedeutet auch, daß die Software selbst langlebiger wird. Da die NutzerInnen der Software sich an der Weiterentwicklung oder Portierung auf neue Plattformen beteiligen können (Open-Source-Programme gehören heute schon zu den meistportierten) und dabei nicht von den Entscheidungen der Hersteller abhängig sind, wird es viel seltener notwendig, Anwendungsprogramme zu wechseln.

5.11.2 Der Einfluß der Kopierschutz-Technologien

Ein Trend, der in die gegenteilige Richtung wirkt, ist die zunehmende Verbreitung verschlüsselter Dateiformate im Interesse des Kopierschutzes oder der Vertraulichkeit von Daten. Das sind natürlich legitime Zwecke, nur behindern sie eben die Langzeitverfügbarkeit. Solange diese Verfahren gesetzlich so streng geschützt sind wie heute, ist ihre Umgehung nicht möglich, und der Zugang zu den Informationen wird selbst für legale Zwecke unterbunden.

5.11.3 Die Bedeutung der Emulation

In den letzten Jahren ist durch die schnelle Entwicklung der Hardware und die damit verbundene Entstehung großer Leistungsreserven die Emulation (allerdings nicht für die Zwecke der Langzeitverfügbarkeit) in den Vordergrund des Interesses gerückt. Es ist damit zu rechnen, daß auch in Zukunft gute Emulatorsoftware für alle verbreiteten Computerplattformen zur Verfügung stehen wird, in vielen Fällen sogar kostenlos als Open Source. Aus diesem Grund dürfte die Emulation in Zukunft eine größere Rolle spielen als heute, und die notwendigen Kenntnisse sollten dadurch weiter gestreut sein.

6 Schlußfolgerungen

6.1 Wie groß ist das Problem?

Hypothese: Alle Informationen, die ohne besondere Berücksichtigung der Langzeitverfügbarkeit digital geschaffen oder digitalisiert und digital gespeichert wurden, sind innerhalb von Jahren vom Verfall bedroht. Selbst die Beachtung der erarbeiteten Empfehlungen etwa von Jeff Rothenberg kann die Langzeitverfügbarkeit nicht in jedem Fall sichern, und es gibt Arten von Daten, auf die die Empfehlungen nicht anwendbar sind.

Wie in Kap. 4.1 auf Seite 54 beschrieben, halten die heute verbreiteten Datenträger nur einige Jahre oder wenige Jahrzehnte lang ihre Daten zugänglich. Die Abspielgeräte werden häufig in noch kürzerer Zeit kaputt oder sie lassen sich nicht an aktuelle Computer anschließen. Die reinen Daten lassen sich zwar mit Hilfe von Netzwerken häufig von veralteten Computern auf neuere kopieren, aber ohne geeignete Software nicht darstellen oder interpretieren (vgl. Kap. 4.4 auf Seite 69). Software ist jedoch an eine gewisse Umgebung angewiesen, die auf späteren Computern nicht einfach herstellbar ist. Die Methoden, all diese Probleme zu lösen, sind ziemlich aufwendig, und führen nicht einmal in allen Fällen zum Ziel.

6.2 Was sind die Ursachen des Problems?

Hypothese: Der Großteil der Computer-Industrie ist wegen des mangelnden Interesses auf der Nachfrageseite nicht oder nur marginal daran interessiert, Langzeitverfügbarkeit in ihre Produkte einzubauen.

Wie in Kap. 4.2.1 auf Seite 64 beschrieben, hat es für die Hersteller von Hard- und Software kaum Vorteile, wenn sie Langzeitverfügbarkeit in ihre Produkte einbauen, im Gegenteil, wenn sie alte Formate und Standards „mitschleppen“ müssen, kann das ihre Kosten erhöhen und sie im sehr intensiven Wettbewerb am Markt behindern.

Hypothese: Die „inhaltsproduzierende Industrie“ ist nicht oder nur marginal daran interessiert, die Langzeitverfügbarkeit ihrer Produkte zu sichern.

Die Aufnahmekapazität des Marktes für Unterhaltungs- und Bildungsinhalte ist mit der Aufnahmekapazität und Freizeit der Bevölkerung beschränkt (vgl. Kap. 4.7 auf Seite 80). Alte Inhalte führen am Markt nicht zu ähnlichen Erlösen wie neue, ihre Pflege verursacht aber Kosten. Die Inhaltsproduzenten sind daher eher daran interessiert, daß die Mehrheit alter Inhalte (bis auf gewisse langfristige Bestseller) verlorengeht.

6.3 Was sind aktuelle Trends?

Hypothese: Große Teile der Computerindustrie und der Unterhaltungsbranche arbeiten an Wegen, die die Sicherung der Langzeitverfügbarkeit noch stärker als bisher behindern.

Wie in den Kapiteln 4.2.1 auf Seite 64 und 5.10.1 auf Seite 107 beschrieben, werden für kommerziell verbreitete Inhalte und mittlerweile auch für eigene Dokumente Technologien in den Markt gedrückt, die durch Verschlüsselung und ähnliche Verfahren den „unberechtigten“ Zugang zur Information unterbinden. Die Überprüfung der Berechtigung setzt komplexe Systeme voraus, deren langfristige Verfügbarkeit sehr fraglich erscheint, und deren Umgehung auf Betreiben der genannten Gruppen strafbar wurde.

6.4 Sind die in der Literatur vorgeschlagenen Verfahren in der Praxis umsetzbar und lösen sie das Problem?

Hypothese: Weder Migration noch Emulation sind in der Lage, alle auftretenden Probleme zu lösen. Beide Verfahren haben ihre Stärken und Schwächen; für die Praxis kann eine Mischung aus beiden die beste Lösung sein. Es gibt Informationen, die mit keinem der genannten Verfahren langfristig zugänglich gehalten werden können; für diese müssen speziellere Verfahren gefunden werden oder die Information geht verloren.

Die Migration eignet sich nicht für alle Arten von Daten (siehe Kap. 5.6 auf Seite 91 und das Experiment 7.2.3 auf Seite IX). Komplexe sowie zusammengesetzte Dateiformate sind auf ihre ursprüngliche Software-Umgebung angewiesen; diese kann in vielen Fällen mit Hilfe der Emulation geschaffen werden. Einige Elemente der Software-Umgebung können jedoch weitere, externe Anforderungen haben, z. B. eine bestimmte Hardware-Komponente (z. B. Dongle) oder einen Freigabe-Server. In diesen Fällen hilft auch die Emulation nicht immer weiter. Manche Lösungsansätze, die technisch möglich wären, fallen unter das verschärfte Urheberrecht und werden mit Gefängnisstrafen bedroht.

6.4.1 Sind die vorgeschlagenen Verfahren im privaten Bereich anwendbar?

Hypothese: Im privaten Bereich sind die Mittel und Kenntnisse, die für die Anwendung der vorgeschlagenen Verfahren der Langzeitverfügbarkeit nötig wären, derzeit kaum vorhanden. Es besteht ein Bedarf an vereinfachten Verfahren und einfach nachvollziehbaren Anleitungen, um die Langzeitverfügbarkeit zu sichern.

Die Migration erfordert gute Kenntnisse über viele Dateiformate, ihre Eigenschaften und Vor- und Nachteile sowie über Konvertiersoftware und deren Automatisierung (siehe Kap. 5.10.2 auf Seite 109). MitarbeiterInnen von Archiven und Bibliotheken können dieses Wissen im Zuge ihrer Arbeit erwerben. Es gibt zumindest für die Konvertierung weit verbreiteter Dateiformate leicht anwendbare Programme. In Privathaushalten dürfte also die Migration ein gangbarer Weg für manche Dateitypen sein (wenn sie sich überhaupt für die Migration eignen). Allerdings sind eine gewisse Erfahrung und ständige Marktbeobachtung erforderlich, um zu erkennen, wann die Migration alter Dateiformate in aktuelle Formate notwendig wird. Außerdem bevorzugen PrivatanwenderInnen selten bewußt offene und standardisierte Dateiformate, was die Migration auch erschweren kann.

Für die Emulation müssen Emulationsprogramme und alte Betriebssysteme sowie Software angeschafft werden. Je älter die Software, desto unwahrscheinlicher ist es, daß sie überhaupt erhältlich ist. Auch die Information, welches Dateiformat welche Software und diese wiederum welches Betriebssystem erfordert, ist ohne eine genaue Dokumentation der Formate und ihrer Voraussetzungen schwer zu beschaffen. Deswegen dürfte Emulation im privaten Bereich höchstens bei technisch orientierten BenutzerInnen durchführbar sein, die DurchschnittsanwenderInnen sind damit wahrscheinlich überfordert.

6.4.2 Unterstützt das Rechtssystem die Langzeitverfügbarkeit digitaler Information?

Hypothese: Alle Industrieländer haben bereits Gesetzgebung, oder sie sind dabei, Gesetze zu verabschieden, die dazu führen, daß die wichtigsten Verfahren der Langzeitverfügbarkeit, nämlich Migration und Emulation, in manchen Fällen illegal werden. Das wird zu Informationsverlust führen, wenn das Problem nicht in spezieller, neuer Gesetzgebung anerkannt und gelöst wird.

Wie in Kap. 5.10.1 auf Seite 107 ausgeführt, behindern Kopierschutz- und verwandte Technologien die Migration der Daten und die Emulation der notwendigen Systeme. Viele technische Hürden können beseitigt werden, allerdings ist die Gesetzgebung bei der Umsetzung der Forderungen der Inhaltsindustrie übers Ziel hinausgeschossen und bestraft auch die Anwendung der Umgehungstechnologien für legitime Zwecke.

Die in Kap. 5.10.3 auf Seite 110 vorgeschlagenen Änderungen des Urheberrechts wären für die langfristige Sicherung des „geschützten“ Teiles der Information absolut unerläßlich. Heute fallen solche Inhalte (Bücher, Musik, Filme etc.) unter die Pflichtexemplarregelungen; wenn sie nur mehr in elektronischer Form vertrieben werden, kann

es ohne gesetzliche Änderungen passieren, daß die Pflichtexemplarbibliotheken ihrer gesetzlichen Aufgabe nicht mehr nachkommen können.

7 Experimente

7.1 Experiment: Analogkopien zwischen VHS-Videokassetten

7.1.1 Versuchsanordnung

Folgende Geräte wurden verwendet:

- S3 Savage/IX-MV Grafikkarte mit TV-Ausgang (S-Video) eines IBM T22 Laptops für die Generierung des Testsignals
- TV-Karte mit Brooktree Bt878 Digitalisier-Chip
- Videorecorder Sharp VC-S2000
- Videorecorder Grundig GV 440

Kassetten:

- TDK TV 240
- BASF EMTEC EQ 260

Diese Versuchsanordnung entspricht bewußt nicht Laborbedingungen: der weitaus größte Anteil von Videoaufnahmen befindet sich schließlich auf VHS-Kassetten in Haushalten.

Das Testbild wurde komplett digital am Computer generiert, um jede Verfälschung durch eine vorhergehende Aufnahme (etwa mit einer Kamera) zu vermeiden.

Für die Generierung des Testbildes wurde der Demo-Modus des Programms XaoS[106] in der Version 3.0 unter Debian GNU/Linux verwendet. XaoS zeigt Fraktale (Visualisierungen komplexer mathematischer Funktionen) an und kann sie auch animieren. Die Fraktale sind teilweise sehr farbig und besitzen feine Strukturen, sie sind daher sehr gut für die Beurteilung der Qualität des Videobildes geeignet. Das Testvideo wurde auf folgende Weise erzeugt:

1. XaoS wurde mit dem Befehl „xaos“ auf der Befehlszeile aufgerufen
2. Durch zweimaliges Drücken von „h“, dann zweimal „2“, wurde das Tutorial „An Introduction to Fractals“ / „Introduction and the Mandelbrot set“ aufgerufen.
3. Diese automatisch ablaufende Demonstration wurde auf den Videorecorder ausgegeben.

[106] Unter diversen Betriebssystemen lauffähige Freie Software, herunterladbar unter http://xaos.theory.org/

Dazu wurde das Lied „Cell Block Tango“ vom Soundtrack des Films „Chicago“ als Ton gespielt. Dieses Lied besteht sowohl aus gesprochenen als auch gesungenen Teilen sowie Perioden nur mit Musik und erlaubt daher die Beurteilung der Qualität dieser drei unterschiedlichen Elemente.

Das Originalsignal (Computer-Bild und Ton) wurde in den Sharp Videorecorder eingespielt und auf die TDK-Videokassette aufgenommen. Von dieser Aufnahme (1. Generation) wurde über SCART-Kabel (da der andere Videorecorder, wie die meisten Consumer-Videorecorder, keinen S-Video-Ein- und Ausgang besitzt), auf die BASF-Kassette kopiert (2. Generation). Diese Aufnahme wurde dann in umgekehrter Richtung wieder auf die TDK-Kassette kopiert (3. Generation), und von dort wieder auf die BASF-Kassette (4. Generation). Das Bild der 4. Generation wurde dann für den Vergleich der Bildverschlechterung mit dem Composite-Eingang der TV-Karte digitalisiert.

7.1.2 Ergebnisse

Bereits die erste Generation zeigt eine geringfügige Verschlechterung der Bildqualität: Während bei der Darstellung des Originalsignals am Fernseher die ziemlich kleine Schrift noch halbwegs (wenn auch mühsam) lesbar ist, sind die Buchstaben auf der Kopie schon etwas verschwommener und ihre Bedeutung ist nur mehr zu erahnen.

Bei der zweiten Generation verschlechtert sich das Bild: die Texte sind bis auf das größere „XaoS“ in der Titelleiste des Fensters nicht mehr lesbar, und vor allem größere gleichfarbige Flächen flimmern sichtbar. Es gibt keine scharfen Abgrenzungen mehr zwischen Flächen unterschiedlicher Farbe. Vertikale gerade Linien sind nicht mehr gerade, sondern haben kleine „Zacken“. Der Ton ist nicht merkbar schlechter geworden.

Bei der dritten Generation ist der Ton noch immer in Ordnung. Das Bild verschlechtert sich weiter: auch die Überschrift „XaoS“ ist trotz der größeren Schrift nur mehr schlecht zu lesen. Alle scharfen Farbübergänge (z. B. Fensterrand, feine Fraktalstrukturen usw.) werfen einen „Schatten“ nach rechts. Das ganze Bild flimmert sichtbar.

Die Aufnahme der vierten Generation ist im Vergleich zum Original schon sehr gestört. Die im Original noch leuchtenden Farben sind hier viel schwächer. Das Flimmern ist noch stärker, der „Schatten“ ist nicht mehr nur rechts, sondern auch unter den Strukturen sichtbar. Beim Ton sind „s“-Laute etwas überbetont.

Diese vierte Aufnahme wurde nocheinmal abgespielt und mit der TV-Karte des Computers digitalisiert. Dieser Vorgang führt nicht mehr zu gleich großen Qualitätsverlusten wie eine ganze VHS-Generation, da weniger analoge Komponenten dazwischen sind. Das rechte Bild in Abb. 4 könnte daher als „viereinhalbte“ Generation bezeichnet werden.

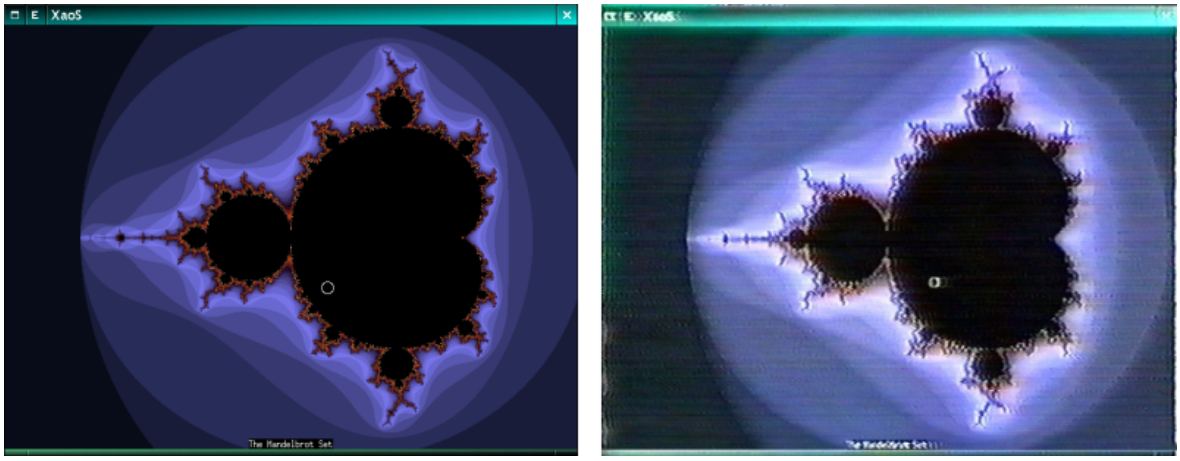


Abbildung 4: Das Originalbild im Vergleich mit der wieder digitalisierten vierten Generation der VHS-Kopie

7.1.3 Schlußfolgerung

Bereits einige wenige Kopiervorgänge mit analogen VHS-Videorecordern, wie sie im Heimbereich üblich sind, verschlechtern die Qualität sehr stark.

7.2 Experiment: Migration unterschiedlicher Dateitypen

In diesem Experiment ging es darum, die Eignung unterschiedlicher Dateiformate für die Migration festzustellen.

Gut für die Migration geeignet sind Dateiformate, deren Zweck und Struktur auf der ganzen Welt ähnlich sind; das Problem, das sie lösen, ist genau abgegrenzt und gut untersucht. Deswegen existieren gut automatisierbare Konvertierungswerkzeuge für sie. Ein Beispiel für diese Anwendung sind Rasterbilder.

Für die Migration nicht so gut geeignet sind Dateiformate, die weniger zum Informationsaustausch als für interne Zwecke eines Programms dienen. Es gibt manchmal keinen Konsens darüber, welche Informationen in der Datei überhaupt gespeichert werden sollen und keine „logischen“ Strukturen für die Daten. Ein Beispiel für diese Kategorie sind Adreßbücher.

7.2.1 Hypothesen für das Migrationsexperiment

Hypothese: Rasterbilder (Bitmaps) sind gut für die Migration geeignet. Es gibt automatische Werkzeuge für die Konvertierung, und es kann ebenfalls automatisiert festgestellt werden, ob die konvertierten Bilder funktional identisch mit dem Original sind.

Hypothese: Digitale Adreßbücher (z. B. von e-mail- und ähnlichen Programmen) sind

für die Migration schlecht geeignet. Zwischen manchen Formaten ist mit fertigen Programmen keine direkte Konvertierung möglich (nur durch Entwicklung eines eigenen Programms), und unterschiedliche Funktionalitäten der einzelnen Programme führen zur Reduktion der übernommenen Daten.

7.2.2 Migration von Rasterbildern

Für dieses Experiment habe ich diverse Bilddateien in unterschiedlichen Formaten von meinen Computern gesammelt (die Vielfalt der Formate und der Herkunftsquellen ist Absicht):

- JPEG (JFIF)-Dateien aus meinem eigenen Fotoalbum
- GIF (schwarzweiß) und farbige JFIF-Dateien von der online verfügbaren Dilbert-Comic-Serie[107] (einmal in der Woche erscheint ein längeres Comic in Farbe; die anderen sind schwarzweiß)
- PNG-Dateien von meiner Webseite
- TIFF- und XPM-Dateien, die vom Programm „WindowMaker“ als Programmsymbol (*icon*) verwendet werden
- BMP-Dateien von einer Windows-95-Installation sowie vom Programm „xmms“
- PCX-Dateien, die als Beispiel bei der Software „gocr“ dabei sind

Diese Bilder (72 Stück, zusammen 2.409.259 Bytes) habe ich der Einfachheit halber in ein neues Verzeichnis kopiert. Vom ganzen Verzeichnis machte ich vor Beginn der Migration eine Kopie, um den Ursprungszustand weiterhin verfügbar zu haben, wenn etwas Unvorhergesehenes passiert.

Wie in Kap. 5.6 auf Seite 90 beschrieben, läßt sich mit zwei Methoden feststellen, ob eine Konversion funktional identisch gelungen ist: einerseits, indem das Ergebnis ins Originalformat zurück-, andererseits, indem das Ergebnis und die Ursprungsdatei mit einem anderen Programm in ein „neutrales“ Format konvertiert werden. Da in dieser Dateisammlung JFIF-Dateien enthalten sind, die mit einer verlustbehafteten Reduktionsmethode gespeichert wurden, ist die erste Verifizierungsmethode nicht anwendbar.

Für die Konversion verwende ich die Software ImageMagick[108]. ImageMagick ist Freie Software, existiert für alle verbreiteten Plattformen und läßt sich sehr gut automatisieren. Es kennt laut Webseite „mindestens 89“ Bilddateiformate.

[107] Dilbert.com - The Official Dilbert Website by Scott Adams http://www.dilbert.com/
[108] ImageMagick http://www.imagemagick.org/

Das Ergebnis der Konversion überprüfe ich mit Netpbm[109] und mit GIMP[110]. Netpbm ist auch Freie Software, existiert auch für alle verbreiteten Plattformen und ist auch gut für die Automatisierung geeignet. Es kennt nicht so viele Formate wie ImageMagick, aber die meisten der hier verwendeten. GIMP ist eigentlich ein extrem leistungsfähiges Bildbearbeitungsprogramm, aber es läßt sich auch für die automatische Konvertierung von Bildern verwenden. Es kennt die meisten relevanten Bildformate, ist ebenfalls Open Source und steht auf den meisten Plattformen zur Verfügung.

Für den Vergleich der konvertierten Dateien mit den Originalen lasse ich beide mit Netpbm in dessen PNM- (portable anymap) Format konvertieren und vergleiche sie dann. Dateien, die von Netpbm nicht korrekt erkannt werden oder die es nicht richtig konvertiert, konvertiere ich für einen weiteren Vergleich mit GIMP. Wenn beide Vergleiche fehlschlagen, wird das gemeldet; das Bild muß in diesem Fall überprüft und eventuell mit einem anderen Programm oder händisch konvertiert werden, und wenn das alles nicht möglich ist, wäre es ein Fall für die Emulation.

Folgendes Skript migriert und überprüft automatisch den Inhalt eines Verzeichnisses mit Bilddateien:

```
#! /bin/sh
# Migration von Bilddateien in ein wählbares Format

# Balázs Bárány <balazs@tud.at> 2004-09-14
# Benutzung: im Verzeichnis mit den Bildern "migrate-images.sh [Format]"
# Format ist die Dateiendung des gewünschten Bildformats, z. B. tif oder png

# Konfiguration
# Welches Programm soll für die Konvertierung verwendet werden
convert=convert

# Überprüfen, ob ein Format angegeben wurde. Wenn nicht, aussteigen
if [ $# = 0 ]; then
  echo "Bitte Dateityp für die Konversion (z. B. png) angeben!"
  exit
fi

# Der erste Befehlszeilenparameter ist der Dateityp
typ="$1"

# Funktion für den Aufruf der GIMP-Konvertierung
gimpconvert () {
  gimp --no-data --no-interface --no-fonts --no-splash --batch \
    "(script-fu-formatconversion \"$1\" \"$2\")" '(gimp-quit 0)' \
    | grep -v 'batch command: executed successfully'
}

# Verzeichnis für die konvertierten Bilder anlegen
mkdir -p "migriert-$typ"
mkdir -p "konvertierfehler-$typ"
```

[109] Netpbm http://netpbm.sourceforge.net/
[110] GIMP - The GNU Image Manipulation Program http://www.gimp.org

```
# Schleife über alle Dateien im aktuellen Verzeichnis
for bild in *; do
  if [ -f "$bild" ]; then
    # Dateiname ohne Erweiterung
    basis=‘echo -n "$bild" | sed ’s/\.[^.]*$//’‘

    # Ziel im Unterverzeichnis mit der gewünschten Erweiterung
    ziel=‘echo -n "$basis" | sed "s/\\(.*\\)/migriert-$typ\/$basis.$typ/"‘

    # Vergleichsdateien
    verfmt=xpm
    ver1=‘echo -n "$basis" | sed "s/\\(.*\\)/konvertierfehler-$typ\/$basis.ori.pnm/"‘
    ver2=‘echo -n "$basis" | sed "s/\\(.*\\)/konvertierfehler-$typ\/$basis.knv.pnm/"‘
    ver3=‘echo -n "$basis" | sed "s/\\(.*\\)/konvertierfehler-$typ\/$basis.ori.$verfmt/"‘
    ver4=‘echo -n "$basis" | sed "s/\\(.*\\)/konvertierfehler-$typ\/$basis.knv.$verfmt/"‘

    # Jetzt mit dem gewählten convert-Befehl konvertieren
    $convert "$bild" "$ziel" 2> /dev/null

    # Das Original und das konvertierte Bild mit Netpbm in PNM konvertieren
    anytopnm "$bild" > $ver1 2> /dev/null
    anytopnm "$ziel" > $ver2 2> /dev/null

    if diff -q "$ver1" "$ver2" 2> /dev/null > /dev/null ; then
    # Konvertierung gelungen, die Dateien sind identisch
      # Die Vergleichsdateien löschen
      rm -f "$ver1" "$ver2"
      echo "$bild korrekt konvertiert"
    else

    # Laut Netpbm nicht korrekt konvertiert
      # Das Original und das konvertierte Bild mit GIMP ins Format für den
      # Vergleich konvertieren
      gimp-convert "$bild" "$ver4" 2> /dev/null
      mv "$ver4" "$ver3" # so sind die internen Namen identisch
      gimp-convert "$ziel" "$ver4" 2> /dev/null

      if diff -q "$ver3" "$ver4" 2> /dev/null > /dev/null ; then
      # Diesmal OK
        # Alle 4 Vergleichsdateien löschen
        rm -f "$ver1" "$ver2" "$ver3" "$ver4"
        echo "$bild korrekt konvertiert"
      else
        # Das konvertierte Bild zu den Konvertierfehlern verschieben und
        # den Mißerfolg melden
        mv "$ziel" "konvertierfehler-$typ"
        echo "$bild konnte nicht konvertiert oder verifiziert werden"
      fi
    fi
  fi
done
```

Weiters ist für die GIMP-Konversion ein GIMP-Befehlsskript erforderlich (die Datei z. B. im eigenen Home-Verzeichnis unter `.gimp-2.0/scripts/formatconversion.scm` speichern):

```
(define (script-fu-formatconversion infile outfile)
  (let* ((img (car (gimp-file-load 1 infile infile)))
  (drawable (car (gimp-image-active-drawable img))))

  (if (not (eqv? (car (gimp-drawable-is-rgb drawable)) TRUE))
    (gimp-image-convert-rgb img)
  )

  (gimp-file-save 1 img drawable outfile outfile)))

(script-fu-register "script-fu-formatconversion"
  _"<Toolbox>/Xtns/Script-Fu/Utils/Format conversion..."
  "Converts between image formats"
  "Balazs Barany"
  "Balazs Barany"
  "2004-09-14"
  ""
  SF-FILENAME "Infile" ""
  SF-FILENAME "Outfile" "" )
```

(Diese Skripts sollten auf allen Unix-ähnlichen Plattformen wie GNU/Linux, BSD, MacOS X usw. ohne oder mit geringen Modifikationen laufen. Für Microsoft Windows ist eine Unix-Umgebung wie Cygwin oder Services For Unix erforderlich. Die erwähnten Programme ImageMagick, GIMP und Netpbm müssen natürlich auch installiert sein.)

Es handelt sich natürlich nur um ein einfaches Skript zur Demonstration der grundlegenden Abläufe der Migration. In der Praxis würde ein richtiges Programm z. B. nach der erfolgreichen Verifikation die Metadaten der migrierten Dateien anpassen, eventuell eine digitale Signatur einfügen usw. Der Vergleich der funktionalen Identität kann auch effizienter gestaltet werden, indem ein Programm die Bilddaten einliest und im Speicher vergleicht. Es würde auch in der Regel nicht viel Sinn machen, Dateien, die bereits im Zielformat vorliegen, nochmal zu konvertieren (es sei denn das Format ist so unklar spezifiziert oder in Programmen so schlecht implementiert, daß es Variationen gibt, die z. B. von gängigen Anzeigeprogrammen nicht immer korrekt angezeigt werden).

Beispiel-Ausgabe des Scripts:

```
2004-09-13.jpg korrekt konvertiert
2004-09-14.gif korrekt konvertiert
alldates-dir.png korrekt konvertiert
alldates-dom.png korrekt konvertiert
font24.pcx korrekt konvertiert
Ftp.png korrekt konvertiert
GNUstep3D.xpm konnte nicht konvertiert oder verifiziert werden
GNUstepGlow.xpm korrekt konvertiert
...
```

Wenn der Vergleich gelungen ist, meldet das Skript den Erfolg und legt das konvertierte Bild im entsprechenden Verzeichnis ab. Bei nicht gelungener Konvertierung oder

anderen Fehlern werden das Bild und die fehlgeschlagenen Konvertierversuche in einem anderen Verzeichnis abgelegt.

Ich habe die Testbilder zuerst mit dem Befehl `migrate-images.sh tiff` ins TIFF-Format konvertiert. Alle bis auf eine Datei konnten konvertiert werden, das wurde durch visuellen Vergleich der Bilder bestätigt. Die problematische Datei liegt im XPM-Format (X PixMap) vor und enthält transparente (durchsichtige) Bereiche. Das Bild wurde konvertiert, aber ImageMagick hat die Transparenz in schwarzen Hintergrund umgewandelt, was die zwei Vergleichsoperationen nachher natürlich entdecken konnten. Es dürfte sich um einen Programmierfehler in ImageMagick handeln, da eine andere transparente XPM-Datei ohne Probleme umgewandelt wurde.

Die fehlerhaft konvertierte Datei läßt sich mit GIMP korrekt in TIFF konvertieren. Wenn sich Probleme mit ImageMagick häufen, wäre es in der Praxis eventuell sinnvoll, die Konvertierung in Zukunft mit GIMP durchzuführen und ImageMagick für die Überprüfung zu verwenden.

In der zweiten Runde konvertierte ich mit `migrate-images.sh png` die Bilder ins PNG-Format. Dabei sind gar keine Konvertierungsprobleme mehr aufgetreten, was auch die visuelle Überprüfung bestätigt.

Zwischen diesen beiden Formaten laufen auch alle weiteren Konvertierungsschritte ohne Probleme ab. Andere Formate kommen für dieses Experiment nicht wirklich in Frage, da sie nicht alle notwendigen Eigenschaften der kompletten Bandbreite der Testbilder (unterschiedliche Farbanzahl, Transparenz usw.) abdecken oder keine offenen Formate sind (z. B. Photoshop PSD).

Der Speicherbedarf der Bilder ist nach der Migration relativ stark angestiegen (TIFF: insg. 17.951.290 Bytes; PNG: 6.870.482 Bytes). Das ist leicht erklärbar: beide Formate verwenden verlustlose Algorithmen; die vorher mit JPEG gespeicherten Bilder sind deswegen deutlich größer geworden, dafür liegen sie jetzt in nicht reduzierten Formaten vor. (Natürlich wurden dadurch die irreversiblen Reduktionen, die der JPEG-Algorithmus früher durchgeführt hat, nicht korrigiert.)

7.2.2.1 Schlußfolgerung Automatische Migration von vielen Bitmap-Dateien ist mit kleinen Einschränkungen möglich. Fehler in der Konvertierung können automatisch entdeckt werden. Die korrekt migrierten Dateien sind mit den Originalen funktional identisch. Der Anteil der fehlgeschlagenen Dateien ist klein, sodaß sie manuell oder mit verbesserten automatischen Verfahren migrierbar sind.

7.2.3 Migration von Adreßbuch-Daten

Für dieses Experiment habe ich einige frei verfügbare Adreßbuch-Programme, Adreßbuch-Konvertierprogramme und komplette Anwendungen mit Adreßbuch-Komponente gesammelt und die von ihnen unterstützten Import- und Exportformate festgestellt.

Ziel des Experiments ist es, die Fallen, die beim Übertragen wichtiger Daten in andere Programme lauern, zu ergründen. Ich wollte im Rahmen der Möglichkeiten fair bleiben und wenn es mehrere Wege gab, den mit dem besten Ergebnis wählen, aber dabei die Funktionen der verwendeten Programme nicht selbst erweitern (also kein eigenes Konversionsprogramm entwickeln und die Import- und Export-Dateien nicht selbst korrigieren, weil DurchschnittsanwenderInnen dazu kaum in der Lage sind).

Programm	vCard[a]	CSV[b]	LDIF[c]	Andere (I: Import, E: Export)
Qtopia Desktop[d] 1.6.2	-	-	-	I: Palm Pilot I, E: Sharp Zaurus
Multisync[e] 0.82	-	-	-	I, E: Sharp Zaurus, Ximian Evolution, Palm Pilot usw.
Mozilla[f] Addressbook 1.7.2	-	I, E	I, E	
Mozilla Thunderbird 0.7.1	-	I, E	I, E	
Ximian Evolution[g] 1.4.6	I, E	-	I	
Sylpheed Claws[h] 0.9.12	I	-	I, E	
Kontact[i] 0.8.1	I, E	I, E	I, E	I: Opera, Eudora, Exchange
abook[j] 0.5.2	E	I, E	I, E	I, E: mutt, pine E: palm, elm

[a] Ein beliebtes Format für Adreßdaten-Austausch im Internet
[b] Mit Komma, Tab oder anderen Zeichen getrennte Textdateien
[c] LDAP Interchange Format
[d] Trolltech: Qtopia Desktop http://www.trolltech.com/download/qtopia/index.html?cid=22
[e] Multisync - A Synchronization Tool http://multisync.sourceforge.net/
[f] Mozilla Home Page http://www.mozilla.org/products/mozilla1.x/
[g] Evolution http://www.gnome.org/projects/evolution/
[h] Sylpheed Claws http://sylpheed-claws.sourceforge.net/
[i] Kontact Homepage http://kontact.kde.org/
[j] abook addressbook program http://abook.sourceforge.net/

Ich selbst speichere meine Adressen in einem Sharp Zaurus Handheld-Computer; diesen synchronisiere ich mit Qtopia Desktop. Das sind also die Quellen für die Daten für diesen Test.

In Qtopia Desktop und am Zaurus ist das Adreßbuch in einer XML-Datei mit ei-

nem speziellem, soweit mir bekannt nur in diesen Anwendungen verwendeten Schema gespeichert. Die Kodierung ist UTF-8, dadurch können theoretisch und auch praktisch alle deutschen und ungarischen Umlaute korrekt gespeichert werden.

Der einzige Weg meiner Adreßdaten zum Rest des Feldes führt über Multisync, das die Daten vom Zaurus in Evolution übertragen kann (1. Schritt). Hierbei treten schon die ersten Verluste auf. Mehrzeilige Anmerkungen faßt Multisync auf eine Zeile zusammen: wo ich längere Wegbeschreibungen, Ordinationszeiten o. Ä. eingegeben habe, sind diese dadurch etwas weniger übersichtlich. Die akademischen Grade der Personen (wie Mag. oder Dr.) übernimmt Multisync nicht.

Evolution speichert nur eine WWW-Adresse einer Kontaktperson, was bei den Leuten, von denen sowohl private als auch geschäftliche URLs erfaßt sind, dazu führt, daß nur die geschäftliche übrigbleibt. Der Rest der Daten wird korrekt übernommen. Evolution kann die Adressen in vCard exportieren, was dann mehrere Programme im Testfeld beherrschen (2. Schritt).

Diese vCard-Datei habe ich als nächstes in Kontact importiert. Hierbei sind schon inakzeptable Fehler passiert: Alle Umlaute in Vor- und Nachnamen wurden durch Kommas ersetzt, während die separat gespeicherten „Anzeigename“-Felder richtig erscheinen. Die beiden Felder sind in der VCF-Datei unterschiedlich kodiert; beide mit UTF-8, aber der Anzeigename zusätzlich mit der „Quoted Printable“-Methode. (Die Datei selbst dürfte in Ordnung sein, Evolution importiert sie auf einem anderen Computer richtig.) Derselbe Fehler trat auch beim Feld „Organisation“ auf. Das andere Programm, das VCF importieren kann, das Adreßbuch von Sylpheed, zeigt ein ähnliches Verhalten, es hat jedoch weit weniger Datenfelder und verliert daher mehr Information, also fuhr ich mit Kontact als geringerem Übel fort. Weitere Fehler: geschäftliche Adressen wurden unter dem Namen „Andere“ dupliziert, aber mit falscher Kodierung (die Kodierung derselben Daten in der Geschäfts-Adresse blieb aber korrekt). Manche Anmerkungen wurden nach ca. 50 Zeichen abgeschnitten, obwohl sie in der Datei richtig sind; andere nicht. Wenn Firma und Abteilung separat angegeben waren, fügte sie Kontact mit einem Strichpunkt zu einem Feld „Firma“ zusammen. Die Geburtstage wurden übernommen, Jahrestage jedoch nicht. Das liegt daran, daß das Feld „Jahrestag“ im vCard-Standard nicht vorgesehen ist; Evolution hat das Feld zwar exportiert, aber das konnte eben nicht in einer standardisierten Form geschehen. Das Gleiche gilt für die Information über Ehepartner.

Kontact kann in LDIF und CSV exportieren (3. Schritt). Ich habe versucht, beide Dateien in Mozilla Addressbook und Mozilla Thunderbird zu importieren, allerdings gelang das nur mit der LDIF-Datei; bei der CSV-Datei haben beide Programme nach einer zeitraubenden Zuordnung der einzelnen Felder nichts aus der Datei importiert

(oder genauer: die erste Zeile der Datei, in der die Feldnamen stehen). Da beide Programme vom selben Projekt stammen, Thunderbird nur eine Variante von Mozilla ist und beide sich im Test mit beiden Adreßbuchdateien identisch verhalten haben, betrachte ich nur mehr Mozilla näher.

Die aus der LDIF-Datei übernommenen Daten zeigen in Mozilla weitere Verluste. Die Vor- und Nachnamen, die Umlaute enthalten und von Kontact falsch gespeichert wurden, sind nun komplett verschwunden (die Eintragungen in der Datei entsprechen nicht den Regeln für UTF-8-Zeichenketten). Immerhin sind die Anzeigenamen nach wie vor korrekt, selbst die mit Umlauten. Bei Datensätzen, die nur eine geschäftliche Adresse hatten, ist dieselbe Adresse auch in die Felder für die private Adresse übernommen worden. Einzelne Felder (Anmerkungen, dienstl. Funktion) sind richtig kodiert, andere (Firmenname, Land usw.) haben die Umlaute verloren. Die Zuordnung von e-mail-Adressen zu „privat“ und „geschäftlich“ ist verlorengegangen, es gibt nur mehr die primäre und eine zusätzliche e-mail-Adresse. Bei Kontakten, die mehr als zwei e-mail-Adressen hatten, sind die über zwei hinausgehenden Adressen verlorengegangen. WWW-Adressen, die übernommen wurden, sind sowohl bei „privat“ als auch bei „geschäftlich“ eingetragen. Da nur mehr das Feld mit dem angezeigten Namen intakt ist, erfolgt die Sortierung nicht mehr nach Nachnamen, sondern störenderweise nach dem Vornamen.

Mozilla kann das Adreßbuch wieder in LDIF und CSV exportieren (4. Schritt). Bei CSV wird allerdings die sinnvolle Konvention, daß die erste Zeile die Feldnamen enthält, nicht eingehalten, das wird zukünftige Importaktionen ziemlich erschweren, da anhand der Daten abgeschätzt werden muß, welche Felder in welcher Reihenfolge in die Datei geschrieben wurden.

Die von Mozilla geschriebenen LDIF- und CSV-Dateien habe ich in abook eingelesen. Den LDIF-Import haben nur der angezeigte Name (immerhin noch mit korrekter Anzeige aller Umlaute), die jeweils erste e-mail-Adresse, die WWW-Adresse, die Anmerkung und der Spitzname überlebt, obwohl auch diverse Telefonnummer- und Adreßfelder in der LDIF-Datei enthalten sind und von abook unterstützt werden. Die Anmerkung wurde weiter gekürzt.

Der CSV-Import in abook ist komplett unnütz. Es gibt keine Möglichkeit, anzugeben, welche Felder in welcher Reihenfolge in der Datei sind; die von Mozilla exportierte Datei wird so eingelesen, daß der Vorname ins Namensfeld kommt, der Nachname ins e-mail-Feld, der ganze Name zur Telefonnummer und die e-mail-Adresse ins Feld für Spitznamen. 20 % der Adressen gehen überhaupt verloren, anscheinend diejenigen mit Umlauten.

Da die mit LDIF importierten Daten trotz der Verluste noch verwendbar waren,

exportierte ich sie ein letztes Mal aus abook (5. Schritt). Abook kennt viele Exportformate; ich habe LDIF und vCard gewählt, Sylpheed kann theoretisch beide importieren.

Praktisch wird beim LDIF-Import ein Fenster angezeigt, in dem die Attribute aus der LDIF-Datei mit denen von Sylpheeds Adreßbuch verbunden werden können, damit sie richtig zugeordnet sind. Ich habe es aber in mehreren Versuchen nicht erreichen können, daß mehr Datenfelder als die e-mail-Adresse eingelesen werden. Sogar die Namen und Anmerkungen fehlen.

Eine Spur besser sieht es mit dem Import der vCard-Datei aus. Ca. 10 % der Namen, jene mit zu vielen Umlauten, gehen verloren, dafür werden die anderen (u. a. einige, in denen Umlaute drinnen waren) zusammen mit der e-mail-Adresse eingelesen. In den Daten, die eingelesen wurden, fehlen jedoch alle Umlaute.

7.2.3.1 Schlußfolgerung Am Ende der fünf Konversionen ist der Datenbestand sowohl qualitativ (Datenfelder) als auch quantitativ (Anzahl der übernommenen Datensätze) geschrumpft. Selbst wenn ganze Datensätze verlorengingen, haben die Programme nicht gewarnt. Dieses Ergebnis ist schon im privaten Bereich inakzeptabel. Offensichtlich ist die Migration mit den zur Verfügung stehenden Werkzeugen für diese Art von Daten kein gangbarer Weg.

Es besteht auch wenig Hoffnung, daß die Werkzeuge besser werden. Datenexport in offene Formate ist für jedes Programm (ob kommerziell oder Open Source) eine ungeliebte Funktion, weil dadurch die NutzerInnen leichter zu einem anderen Programm wechseln können. (Auch wenn ein Umstieg unter diesen Voraussetzungen nicht empfehlenswert erscheint.)

Die Formate sind offensichtlich auch zu wenig genau spezifiziert. vCard ist ein offener Internet-Standard, explizit für Adreßdatenaustausch gedacht, aber die Export- und Importfunktionen der verschiedenen Programme machen trotzdem Fehler (z. B. im Bereich der Kodierung) oder stoßen an die Grenzen des Formats (Jahrestag und Ehepartner).

LDIF ist ein sehr flexibles, nicht nur für Adreßdaten gedachtes Format; aus diesem Grund gibt es Mehrdeutigkeiten bei der Interpretation der Bedeutungen der Datenfelder (wie bei Kontact und Sylpheed), und nicht alle Programme können mit allen Kodierungsverfahren nichtenglischer Texte umgehen.

Überhaupt scheinen manche Programme so US-zentrisch, daß sie mit Umlauten große Probleme haben. Das Speichern der Umlaute im Programm selbst kann funktionieren, aber das gilt nicht in allen Fällen für den Import und Export.

7.3 Experiment: Emulation alter DOS-Programme

Dieses Experiment hat einen realen Hintergrund. Wohnpark-TV[111], der Offene Fernsehkanal im Wohnpark Alt-Erlaa (Wien 23), verwendet derzeit zwei spezielle DOS-Programme für die Produktion und das Senden von Teletext- und Infotext-Inhalten. Beide laufen nicht unter Windows 2000 oder XP, nur unter den DOS-basierten Windows-Versionen (95, 98, ME) und natürlich DOS selbst. Das ist ungünstig für die ProduzentInnen, da sie beide Betriebssysteme installiert haben und ihren Rechner nur für die Arbeit mit den zwei Programmen neu starten müssen. Es würde die Nutzbarkeit beider Programme stark verbessern, wenn sie mit Hilfe der Emulation unter modernen Betriebssystemen wie Windows 2000 oder GNU/Linux verwendbar wären.

Für den Teletext verwendet Wohnpark-TV den Teletext Editor der Firma tss. Das Programm läuft unter DOS und ist mit einem Dongle, der am Druckerport des Computers angesteckt sein muß, vor unlizenzierter Nutzung (also z. B. auf mehreren Computern gleichzeitig) geschützt. Es läuft auch ohne Dongle, aber dann läßt es das Speichern und auf Sendung Stellen von Teletext-Seiten nicht zu. Die schwierigste Aufgabe für den Emulator beim Teletext Editor ist das Durchschleifen des Informationsflusses zwischen Programm und Dongle.

Der Infotext entsteht im Programm Scala Multimedia. Scala stammt aus den letzten Jahren, in denen DOS noch eine Bedeutung hatte, und nutzt die Grafik des Computers sehr intensiv. Die Routinen dazu wurden anscheinend im Interesse maximaler Geschwindigkeit sehr hardwarenah programmiert; das Programm läuft auch nicht mit jeder Grafikkarte korrekt. Bei Scala ist die wichtigste Aufgabe des Emulators, eine hinreichend leistungsfähige Grafikhardware korrekt zu emulieren.

7.3.1 Verwendete Emulatoren

Für die Emulation in der Langzeitarchivierung kommen nur echte Emulatoren in Frage, die alle Elemente des Computers emulieren. Die Anforderungen von Wohnpark-TV sind etwas anders, da es hier nur darum geht, die Software auf der Architektur, für die sie gedacht ist, noch einige Jahre lang in Betrieb zu halten.

Für beide Aufgaben gleichzeitig geeignet scheinen die echten Emulatoren QEMU und Bochs. Während der Teletext Editor wegen des Dongles an echte PC-Hardware gebunden ist, könnte wenigstens Scala mit einem echten Emulator auch auf einer anderen Plattform (z. B. unter MacOS X) weiterbenutzt werden.

Es gibt einige Projekte (Dosemu[112] und VMware), die mit Virtualisierung arbeiten und deswegen schneller laufen, aber eben nur auf Intel-kompatiblen PCs. Wäh-

[111] Wohnpark-TV Homepage http://www.alterlaa.net/Klubs/wpmedia/
[112] DOSEMU http://dosemu.sourceforge.net/

rend sie für die Emulation langfristig nicht geeignet sind, würden sie das Problem von Wohnpark-TV auch lösen, wegen ihrer Geschwindigkeit vielleicht sogar besser als die echten Emulatoren.

Einen eigenen Weg geht das auf DOS-Spiele spezialiserte Projekt DOSBox[113]. Es emuliert auch den Prozessor, läuft deswegen auch auf anderen Architekturen, aber es ist kein allgemeiner PC-Emulator, sondern nur einer für DOS. Aus diesem Grund könnte DOSBox für diese spezielle Aufgabe sowohl in der Langzeitarchivierung als auch für Wohnpark-TV nützlich sein, wenn es Aufgaben löst, die die anderen Emulatoren nicht oder schlechter lösen.

7.3.2 Bochs 2.1.1

Um Bochs zum Laufen zu bekommen, muß zuerst seine Konfigurationsdatei bearbeitet werden, z. B. um Dateien als virtuelle Laufwerke zuzuordnen.

Der Teletext Editor startet korrekt in Bochs, wenn auch sehr langsam. Das angesteckte Dongle wird trotz entsprechender Konfiguration der parallelen Schnittstelle nicht erkannt, der Editor ist also nicht produktiv verwendbar. Häufig treten Grafikfehler auf, die die Bedienung des Programms schwer bis unmöglich machen.

Scala läßt sich unter Bochs korrekt installieren. Das Hardware-Diagnose-Programm erkennt eine unterstützte Grafik- und Soundkarte. Scala selbst läuft auch korrekt, aber sehr langsam. Die Audioausgabe funktioniert nur fallweise, ohne erkennbares Muster.

7.3.3 QEMU 0.6.0

QEMU braucht nicht konfiguriert zu werden, es reicht, beim Aufruf die Namen der zu benutzenden Laufwerk-Dateien anzugeben.

Unter QEMU startet der Teletext Editor etwas schneller als unter Bochs. Da QEMU jedoch die parallele Schnittstelle gar nicht unterstützt, wird das Dongle nicht erkannt, und der Editor läuft nur im Demo-Modus, ohne die Möglichkeit, Dateien zu speichern. Ähnliche Grafikfehler wie unter Bochs treten auf. (QEMU hat Teile der Grafikschnittstelle von Bochs übernommen, also ist das nicht verwunderlich.) Allerdings unterstützt QEMU auch die Emulation eines anderen Grafikkartentyps, mit dem die Anzeige zwar auch nicht ganz korrekt, aber verwendbar funktioniert.

Scala ist verwendbar. Das Diagnoseprogramm erkennt Grafik- und Soundkarte und gibt den Sound auch korrekt wieder. Scala selbst läuft mit akzeptabler Geschwindigkeit ab. Allerdings stürzt QEMU gelegentlich ab, einmal ist mir dabei der Mauszeiger stehengeblieben, was die weitere Arbeit am Computer etwas behindert hat, bis ich QEMU

[113] DOSBox, a x86 emulator with DOS http://dosbox.sourceforge.net/

wieder startete und beendete und es den Mauszeiger freigab. Bei den Abstürzen hat Scala jeweils Töne ausgegeben und auch die Fehlermeldung von QEMU spricht von einem Fehler mit dem Audio-System. Ohne Audio zu verwenden, was für die Nutzung bei Wohnpark-TV akzeptabel ist, zeigte sich QEMU aber in einem mehrere Minuten langen Test stabil und machte keine weiteren Probleme.

7.3.4 Dosemu 1.2.1

Dosemu ist ziemlich schwer zu konfigurieren und sehr Linux-zentrisch. Es verwendet keine Abbilder von Festplatten und Disketten, sondern legt die DOS-Dateien in normalen Verzeichnissen ab.

Das Teletext-Programm startet schnell, findet aber (trotz stundenlanger Internet-Recherche und Konfiguriererei) das Dongle nicht. Im Demo-Modus läuft es jedoch ohne Grafikfehler und mit angenehmer Geschwindigkeit.

Scala läßt sich mit dem mitgelieferten Installationprogramm nicht korrekt installieren, manche notwendige Verzeichnisse werden nicht kopiert. Nach händischer Korrektur dieser Fehler starten zwar das Diagnoseprogramm und Scala selbst, aber sie akzeptieren keinerlei Eingabe, weder von der Maus noch von der Tastatur.

7.3.5 VMware Workstation 4.0

VMware, das einzige Nicht-Open-Source-Programm im Test, ist recht einfach konfigurierbar, für alle relevanten Einstellungen gibt es Menüs und Einstellungsfenster.

Das Teletext-Programm findet sein Dongle auch unter VMware nicht. Im Demo-Modus ist es akzeptabel bedienbar.

Das Diagnoseprogramm von Scala läuft in VMware korrekt ab und findet Sound- und Grafikhardware. Scala selbst läuft jedoch nicht, es steigt mit einer grafik-bezogenen Fehlermeldung aus.

7.3.6 DOSBox 0.61

DOSBox hat nicht viele Einstellungsmöglichkeiten und läuft ohne aufwendige Konfiguration.

Da DOSBox keine parallelen Schnittstellen unterstützt, erkennt der Teletext Editor darin das Dongle auch nicht. Im Demo-Modus läuft er mit akzeptabler Geschwindigkeit.

Das Diagnoseprogramm von Scala startet, aber beim Testen der Grafik-Einstellungen bleibt der Emulator mit einem schwarzen Bildschirm hängen und tut nichts mehr. Dasselbe geschieht beim Laden von Scala Multimedia.

7.3.7 Zusammenfassung

Programm	Teletext Editor	Scala Multimedia
Bochs	- -	+ langsam
QEMU	- -	+ OK
Dosemu	-	-
VMware	-	- -
DOSBox	-	- -

Der Teletext Editor läuft mit keinem Emulator komplett richtig. Drei der verwendeten Emulatoren unterstützen theoretisch die parallele Schnittstelle, sollten also die Daten des Dongles zum Programm durchreichen, das gelingt jedoch in der Praxis nicht. Da das Dongle zum Kopierschutz dient, wurde dessen Abfrage wahrscheinlich so programmiert, daß möglichst keine Umgehungsmöglichkeiten – zu denen auch die Emulation gehören kann – vorhanden sind. Dies läßt auch für zukünftige Emulatoren nicht viel Gutes erwarten.

Scala als ziemlich hardware-nah programmierte Software läuft nur in den zwei echten Emulatoren Bochs und QEMU richtig. Bochs ist von seinem Design her selbst auf meinem schnellsten Computer in der Emulation etwas langsam für Scala, das sollte jedoch auf zukünftigen Computersystemen kein Problem mehr sein. Die Emulationsgeschwindigkeit in QEMU ist ganz akzeptabel, der Emulator hat sich nur etwas instabil verhalten. Für die Zwecke von Wohnpark-TV erscheint QEMU derzeit trotzdem als verwendbar.

Index

Abbildungsverzeichnis

Literatur

[Abid98] Abid, A.: *Memory of the World: Preserving the Documentary Heritage*. In: Whiffin, J. I.; Havermans, J. (Hrsg.): *Library Preservation and Conservation in the '90s*, Saur, München, 1998, S. 122–132 14

[Ado03] Adobe Systems Incorporated: *PDF Reference*. 4. August 2003. WWW: http://partners.adobe.com/asn/tech/pdf/specifications.jsp 42

[Arps93] Arps, M.: *CD-ROM: Archival Considerations*. In: Mohlhenrich, J. (Hrsg.): *Preservation of Electronic Formats & Electronic Formats for Preservation*. Highsmith Press, Fort Atkinson, Wisconsin, 1993, S. 83–107 61

[Bor+03] Borghoff, U. M.; Rödig, P.; Scheffczyk, J.; Schmitz, L.: *Langzeitarchivierung*. dpunkt.verlag, Heidelberg, 2003 44, 73, 78, 84, 85, 87, 90, 96, 99, 103

[Bra+04] Bray, T.; Paoli, J.; Sperberg-McQueen, C. M.; Maler, E.; Yergeau, F.: *Extensible Markup Language (XML) 1.0*. 3. World Wide Web Consortium, 4. Februar 2004. WWW: http://www.w3.org/TR/2004/REC-xml-20040204/ 36

[Bred95] Bredereck, K.: Gefährdung, Restaurierung und Konservierung von Schriftgut. In: *Spektrum der Wissenschaft*, 1995, September, S. 96–107 56

[Brem03] Bremer, L.: Platten-Karussell. In: *c't*, 2003/14, 30. Juni, S. 136–138 59

[Bror03] Brors, D.: Gruppendynamik – Microsofts Strategie hinter Office 2003. In: *c't*, 2003/21, Oktober, S. 126 40

[Byer03] Byers, F. R.: *Care and Handling of CDs and DVDs*. National Institute of Standards and Technology, Gaithersburg, Maryland, Oktober 2003 62

[Canf98] Canfora, L.: *Die verschwundene Bibliothek – Das Wissen der Welt und der Brand von Alexandria*. Rotbuch-Verlag, Hamburg, 1998 14, 54, 82

[Cass02] Casson, L.: *Bibliotheken in der Antike*. Artemis & Winkler, Düsseldorf, 2002 83

[Clau04] Clausen, L. R.: Handling file formats / The State and University Library, Arhus / The Royal Library, Copenhagen. 2004. – Forschungsbericht. WWW: http://www.netarchive.dk/website/publications/FileFormats-2004.pdf 28

[Cony90] CONYERS, J. J.: Taking a byte out of history: the archival preservation of federal computer records / U.S. House Representatives Committee on Government Operations. Washington, 6. November 1990 (House Report 101-978) – Forschungsbericht 13

[Day01] DAY, M.: *Metadata for Digital Preservation: A Review of Recent Developments*. In: CONSTANTOPOULOS, P.; SOLVBERG, I. T. (Hrsg.): *Research and Advanced Technology for Digital Libraries, Proceedings of the 5th European Conference (ECDL 2001)*, Springer, Berlin Heidelberg, September 2001, S. 161–170 9

[Dech00] DECHTJAREW, B.: Microsoft zwingt Grau raus. In: *c't*, 2000/9, 25. April, S. 66 53

[Del+03] DELLAVALLE, R. P.; HESTER, E. J.; HEILIG, L. F.; DRAKE, A. L.; KUNTZMAN, J. W.; GRABER, M.; SCHILLING, L. M.: Going, Going, Gone: Lost Internet References. In: *Science*, 2003, 31. Oktober, S. 787 14

[DuBo03] DUCE, D.; BOUTELL, T.: *Portable Network Graphics (PNG) Specification*. 2. World Wide Web Consortium, 10. November 2003. WWW: http://www.w3.org/TR/PNG/ 45

[Duch88] DUCHEIN, M.; WALNE, P. (Hrsg.): *Archive buildings and equipment*. 2. Saur, München, 1988 58

[Embe02] EMBERTON, D.: The digital dark age. In: *Shift magazine*, 2002, 5. Juli. WWW: http://www.shift.com/content/web/385/1.html 4

[Gar+01] GARCIA-GUINEA, J.; CÁRDENES, V.; MARTÍNEZ, A. T.; MARTÍNEZ, M.: Fungal bioturbation paths in a compact disk. In: *Naturwissenschaften*, 2001/8, August, S. 351–354 61

[Gies04] GIESELMANN, H.: Auf Kleben und Tod. In: *c't*, 2004/9, 19. April, S. 134–139 61

[Hans03] HANSEN, S.: Un-CDs, nein danke! In: *c't*, 2003/9, April, S. 112 24

[Henz99] HENZE, V.: Langzeitarchivierung von Disketten. In: *Dialog mit Bibliotheken*, 1999/3, S. 15–17 11

[Himm04] HIMMELEIN, G.: Baustelle Sicherheit. In: *c't*, 2004/12, 1. Juni, S. 42 108

[JoBe01] JONES, M.; BEAGRIE, N.: *Preservation Management of Digital Materials*. The British Library, London, 2001. WWW: http://www.dpconline.org/graphics/handbook/ 70, 85, 89

[Kahl97] KAHLE, B.: Preserving the Internet. In: *Scientific American*, 1997, März, S. 72 14

[Kasd98] KASDORF, B.: SGML and PDF – Why we need both. In: *Journal of Electronic Publishing*, 1998/4, Juni. WWW: http://www.press.umich.edu/jep/03-04/kasdorf.html 34

[Klin59] KLINCKOWSTROEM, C. G. v.: *Knaurs Geschichte der Technik*. Droemersche Verlangsanstalt Th. Knaur Nachf., München Zürich, 1959 17, 54

[Korn93] KORNWACHS, K.: *Information und Kommunikation*. SEL-Stiftung. Springer-Verlag, Berlin Heidelberg, 1993 19, 21, 66, 67

[Lind03] LINDAU, E. E.: Gesamtlösung statt Fleckerlteppich: Klare E-Government-Strategie fehlt. In: *Computerwelt*, 2003/8, 14. Februar, S. 1 7

[Lori01] LORIE, R. A.: Preserving Digital Information – An Alternative to Full Emulation. In: *Zeitschrift für Bibliothekswesen und Bibliographie*, 2001/3-4, S. 205–209 92, 97

[Lü02] LÜKE, H. D.: Zur Frühgeschichte der Digitalisierung. In: *Frequenz*, 2002/5-6, S. 117–122 21

[Miel04] MIELKE, K.: Lizenzgestrüpp: Rechtsunsicherheiten beim Umgang mit Standardsoftware. In: *c't*, 2004/22, Oktober, S. 210–217 104, 105

[PeHa03] PENN, W. A.; HANSON, M. J.: The Syracuse University Library Radius Project: Development of a non-destructive playback system for cylinder recordings. In: *First Monday*, 2003/5, Mai. WWW: http://firstmonday.org/issues/issue8_5/penn/ 57

[Rana91] RANADE, S.: *Mass storage technologies*. Meckler Publishing, Westport, 1991 68

[Rink03] RINK, J.: Gemstar verkauft keine E-Books mehr. In: *c't*, 2003/14, 30. Juni, S. 33 108

[Roth95a] ROTHENBERG, J.: Ensuring the Longevity of Digital Documents. In: *Scientific American*, 1995/1, Januar, S. 24–29. WWW: http://www.clir.org/pubs/archives/ensuring.pdf 4, 8, 13, 38, 85, 99

[Roth95b] ROTHENBERG, J.: Die Konservierung digitaler Dokumente. In: *Spektrum der Wissenschaft*, 1995, September, S. 66–71 73

[Roth99] ROTHENBERG, J.: Avoiding Technological Quicksand: Finding a Viable Technical Foundation for Digital Preservation / Council on Library and Information Resources. 1999. – Forschungsbericht. WWW: http://www.clir.org/pubs/reports/rothenberg/contents.html 86, 87, 90

[Roth00] ROTHENBERG, J.: *Using Emulation to Preserve Digital Documents*. Koninklijke Bibliotheek, Juli 2000 8, 90, 92, 93, 97, 98

[Scha01] SCHAARSCHMIDT, R.: *Archivierung in Datenbanksystemen*. B. G. Teubner, Stuttgart/Leipzig/Wiesbaden, 2001 50

[Schn96] SCHNEIDER, W.: Langzeit-Archivierung von magnetischen Datenträgern. In: *KES – Zeitschrift für Kommunikations- und EDV-Sicherheit/BSI-Forum*, 1996/1, S. 61 59

[Schn97] SCHNEIDER, W.: Langzeit-Archivierung von optischen Datenträgern. In: *KES – Zeitschrift für Kommunikations- und EDV-Sicherheit/BSI-Forum*, 1997, Juni, S. 57–58 23

[ScTr04] SCHÜLER, P.; TRINKWALDER, A.: Ungedruckte Vordrucke. In: *c't*, 2004/20, 20. September, S. 196–200 75

[Shan93] SHANNON, C. E.: *Information theory*. In: SLOANE, N. J. A.; WYNER, A. D. (Hrsg.): *Collected Papers*. IEEE Press, Piscataway, 1993, S. 212–220 5

[Shen97] SHENK, D.: *Data Smog: Surviving the Information Glut*. HarperEdge, 1997 80

[Siet02] SIETMANN, R.: Digitales Alzheimer. In: *c't*, 2002/25, 2. Dezember, S. 52–53 4

[Smit99a] SMITH, A.: *The Future of the Past: Preservation in American Research Libraries*. Council on Library and Information Resources, Washington, D.C., April 1999 58, 91

[Smit99b] SMITH, A.: *Why Digitize?* Council on Library and Information Resources, Washington, D.C., Februar 1999 WWW: http://www.clir.org/pubs/reports/pub80-smith/pub80.html 18, 22

[Smit00] SMITH, M.: *Enigma entschlüsselt*. Heyne, München, 2000 22

[StBe97] STEINBRINK, B.; BEHR, B.: CD-R im Härtetest: Wie lange halten CD-Recordable-Medien? In: *c't*, 1997, September, S. 240–245 61, 62

[Step98] STEPANEK, M.: Data storage: From digits to dust. In: *BusinessWeek*, 1998, 20. April, S. 61 13

[Ston99] STONEBRAKER, M.: *Objektrelationale Datenbanken*. Hauser, München; Wien, 1999 50

[Vö96] VÖLZ, H.: *Informationsspeicher: Grundlagen – Funktionen – Geräte*. expert verlag, Renningen-Malmsheim, 1996 17, 20, 24, 25, 60

[Volp03] VOLPE, F. P.: Die Un-CDs – So arbeiten Abspielsperren für Audio-CDs. In: *c't*, 2003/7, 24. März, S. 144 24

[Wä95] WÄCHTER, W.: Strategien für die Konservierung und Restaurierung von Schriftgut. In: *Spektrum der Wissenschaft*, 1995, September, S. 105–107 55

[Wand02] WANDTKE, A.: Copyright und virtueller Markt in der Informationsgesellschaft. In: *Gewerblicher Rechtsschutz und Urheberrecht*, 2002, 8. Januar, S. 1–11 102

[Wau+00] WAUGH, A.; WILKINSON, R.; HILLS, B.; DELL'ORO, J.: *Preserving Digital Information Forever*. In: NÜRNBERG, P. J.; HICKS, D. L.; FURUTA, R. (Hrsg.): *Proceedings of the fifth ACM conference on Digital libraries*, ACM Press, San Antonio, Texas, 2000, S. 175–183 85

[Webe91] WEBERS, J.: *Handbuch der Film- und Videotechnik*. 3. Franzis, München, 1991 58

[Webe94] WEBERS, J.: *Handbuch der Studiotechnik*. Franzis, Poing, 1994 59

[Wett95] WETTENGEL, M.: Maschinenlesbare Datenträger: Zusammenstellung Archivrelevanter Normen und Standards elektronischer Speichermedien. In: *Der Archivar*, 1995/3, S. 461–471 57, 59

[Wett97] WETTENGEL, M.: Zur Rekonstruktion digitaler Datenbestände aus der DDR nach der Wiedervereinigung. In: *Der Archivar*, 1997/4, S. 735–747 37, 38, 59

[Weyh00] WEYHER, C.: *Electronic Publishing in der wissenschaftlichen Kommunikation*. Verlag für Berlin-Brandenburg, Potsdam, 2000 (Materialien zur Information und Dokumentation) 109

[Wor99] World Wide Web Consortium: *HTML 4.01 Specification*. 24. Dezember 1999. WWW: http://www.w3.org/TR/html4/ 36

[Wor02] World Wide Web Consortium: *XHTML 1.0 The Extensible HyperText Markup Language*. 2. 1. August 2002. WWW: http://www.w3.org/TR/xhtml1/ 36

[Zimm01] ZIMMER, D. E.: *Die Bibliothek der Zukunft*. Ullstein, Potsdam, 2001 9, 55, 56, 80

Lebenslauf

Balázs Bárány

A. Baumgartnerstr. 44/A3/255, 1230 Wien
E-mail: balazs@tud.at – Homepage: http://tud.at

Geboren	1. 1. 1976, Budapest, Ungarn
Mutter	Éva Janni, Informatikerin
Vater	Sándor Bárány, Informatiker

Ausbildung

1982 – 1989	Besuch der (achtjährigen) Grundschule, Budapest, Ungarn
1989 – 1992	Besuch des Bundesgymnasiums Stubenbastei, Wien
1992 – 1994	Besuch der Schule für Datenverarbeitungskaufleute, Wien
1996	Externistenreifeprüfung BHAK Pernerstorfergasse, Wien
1996 –	Studium der Publizistik- und Kommunikationswissenschaft und gewählter Fächer (Statistik, Wirtschaftsinformatik) an der Universität Wien und Technischen Universität Wien
1997 – 2002	Tutoriumsleitung bei der Lehrveranstaltung „Information und Dokumentation“ am Institut für Publizistik- und Kommunikationswissenschaft der Universität Wien
1998	Teilnahme am Ausbildungsprogramm „Gruppenleitung und Kommunikationstraining für TutorInnen“ an der Universität Wien
1997 – 1998	Tutoriumsleitung bei weiteren Lehrveranstaltungen an der Universität Wien und Technischen Universität Wien

Berufliche Tätigkeit

1994 – 2000	Tätigkeit als Programmierer, Firma LB-data, Wien Entwicklung des medizinischen Informationssystems „MedInfo“
1997 –	Ehrenamtliche Tätigkeit als Techniker und Redakteur bei Wohnpark-TV, Wien
2000 –	Tätigkeit als Programmierer, Firma apc interactive, Wien Entwicklung von Nutzungsauswertungssystemen
2000 –	Entwicklung Freier Software als Projektleiter („fwanalog“: Sicherheitssoftware) oder Mitarbeiter („MoviX“: Mediensoftware)
2004 –	Tätigkeit als Tanzlehrer-Assistent, Tanzschule Immervoll, Wien