Techno-Logik

Apparaturen, Architekturen, Verfahren

Der Computer als Medientechnologie rekonfiguriert die Versammlung *von* und den Umgang *mit* Informationen. Im Übergang vom Analogen zum Digitalen gewinnen Informationen zunehmend an Autonomie. Sie zirkulieren durch die globalen Computernetzwerke und bieten unüberschaubare Möglichkeiten zur Selektion, Kombination und Rekombination. Die Gesamtheit des Weltwissens ist – so scheint es jedenfalls – in digitalen Datenbanken versammelt und kann aus diesen abgefragt werden. Und mehr noch, aus vorhandenen Informationen können neue geschöpft werden, sodass digitale Datenbanken nicht nur als enormer Informationsspeicher dienen, sondern auch als Ressource respektive Basis für das Entdecken neuer Informationen. Dies nur als einen Effekt der Digitalisierung und der damit einhergehenden Entmaterialisierung von Informationen zu betrachten würde zu kurz greifen. Denn der vermeintlichen Immaterialität digitaler Informationen steht ihre fortwährende Materialität entgegen, die bedingt, was in Computern auf welche Weise als Information verwaltet und verarbeitet werden kann.

Wenn im Folgenden nach der Techno-Logik digitaler Datenbanken gefragt wird, sollen die materiellen Voraussetzungen der computertechnischen Verwaltung digitaler Informationen beleuchtet werden. Den Ausgangspunkt bildet die Betrachtung der Festplatte als hardwaretechnischer Horizont, vor dem sich die Entwicklung digitaler Datenbanken abzeichnet. Die Einführung der Festplatte in den 1950er Jahren eröffnete nicht nur Möglichkeiten für vielfältige neuartige Computeranwendungen, sondern brachte auch Herausforderungen mit sich, die die Entwicklung von Datenbankmanagementsystemen (DBMS) zur Folge hatten. Durch die besondere Betrachtung der Festplatte als einer spezifischen Speichertechnologie sollen andere medienhistorisch ebenso bedeutsame Entwicklungen im Bereich der Computerhardware jedoch nicht marginalisiert werden. Das Ziel ist vielmehr, die Bedeutung einer Technologie herauszustellen, die in der Mediengeschichte des Computers bisher noch nicht hinreichend gewürdigt wurde.

Davon ausgehend wird im Abschnitt »Datenbankmodelle« die Entwicklung der ANSI/X3/SPARC-Datenbankarchitektur nachgezeichnet, welche die konzeptuelle, aber auch idealisierte Grundlage der Verwaltung von Informationssammlungen in

digitalen Datenbanken bildet. Vor dem Hintergrund dieser Datenbankarchitektur können Datenbanksysteme als diejenigen materiellen Infrastrukturen verstanden werden, die digitalen Informationen ein gewisses Maß an Autonomie verleihen. Der Gebrauch von Informationen wird dabei zunehmend von ihrer technischen Verwaltung in der Tiefe des Computers entkoppelt. Dies eröffnet mannigfaltige Präsentations-, Abfrage- und Auswertungsmöglichkeiten der versammelten Informationen. Sofern der Eindruck der Immaterialität digitaler Information auf diesen flexiblen Gebrauchsmöglichkeiten beruht, sind Datenbankmanagementsysteme die materielle Basis immaterieller Informationen.

Schließlich werden im dritten Teil zu »Data + Access« verschiedene Modi der computertechnischen Verwaltung und Verarbeitung von Bedeutung thematisiert. Das Hauptaugenmerk wird zunächst auf relationale Datenbanken gelegt, die spätestens seit den 1980er Jahren zum Paradigma für Datenbanken im engen Sinn der Informatik geworden sind. Vor diesem Hintergrund werden abschließend Websuchmaschinen und das Semantic Web als alternative Technologien der Zuschreibung und Verarbeitung von Bedeutung im doppelten Sinn von Gehalt und Relevanz thematisiert.[1]

Direct Access: Zur Festplatte als Herausforderung digitaler Datenbanken

Für die Formulierung der Datenbankidee Anfang der 1960er Jahre und die konzeptuelle und technische Entwicklung von DBMS seit den 1970er Jahren erwies sich die Erfindung der Festplatte als persistenter Sekundärspeicher mit wahlfreiem Zugriff als grundlegend.[2] Ihre Markteinführung 1956 steht im Kontext zahlreicher Bemühungen, brauchbare Alternativen zur Lochkarte als Datenträger zu finden. Motiviert wurde diese Suche von dem Wunsch, die Verwaltung digitaler Informationen zu automatisieren. Wie im Folgenden zu sehen sein wird, hat

1 | Websuchmaschinen und Semantic Web-Anwendungen können als Datenbanken im weiten Sinn betrachtet werden, die auf unterschiedliche Weise das Suchen, Finden und Verarbeiten in bzw. von Informationssammlungen ermöglichen.

2 | Als Sekundär- oder Massenspeicher werden Speichermedien bezeichnet, auf die Computerprozessoren keinen direkten Zugriff haben, die jedoch fester Bestandteil des Computers sind. In der Speicherhierarchie des Computers werden Sekundärspeicher, wie Festplatten oder Solid-State-Drives, gemeinhin vom Primärspeicher unterschieden. Letzterer wird auch als Arbeits- oder Hauptspeicher bezeichnet und dient nicht der dauerhaften Speicherung von Informationen, sondern deren Verarbeitung. Im Folgenden werden ausschließlich Technologien behandelt, die dem dauerhaften Vorhalten von binär-digital codierten Informationen dienen. Daher wird zum Zweck der besseren Lesbarkeit vereinfacht von Speichertechnologien gesprochen, wobei jedoch stets Sekundärspeicher gemeint sind.

die Einführung der Festplatte nicht nur vielfältige Möglichkeiten der computertechnischen Informationsverarbeitung eröffnetet, sondern bestimmte Herausforderungen mit sich gebracht, die letztlich auch die Entwicklung von Datenbanktechnologien vorangetrieben haben. Eine zentrale Bedeutung in dieser Entwicklung hat das Problem der Datenunabhängigkeit, das seit den 1970er Jahren als eine Kernherausforderung für digitale Datenbanktechnologien erachtet wird.

Entkopplung von Ort und Ordnungen digitaler Sammlungen

In der Frühzeit der computertechnischen Informationsverarbeitung war der Großteil der Informationen, mit denen Computer operierten, noch nicht *in* diesen gespeichert. Die zu verarbeitenden Informationen blieben dem Computer äußerlich und mussten stets aufs Neue via Lochkarten eingelesen werden. Elektronische Datenverarbeitung meinte zunächst nur die programmgesteuerte Auswertung von Informationen. Die Verwaltung der auf Lochkarten gespeicherten Informationssammlungen war Aufgabe der Computernutzer, wie das folgende Anwendungsszenario veranschaulicht:

»In a typical operation, clerks receiving an order would search the tubs, pick out cards containing the needed customer and item order information, and send the cards to the machine room, where the needed documents – shipping room instructions, packing slips, invoices, shipping labels and bills of lading – were produced.« (American Society of Mechanical Engineers 1984: 8)

Die in diesem Beispiel beschriebene Verkopplung von menschlicher Informationsverwaltung und technischer Informationsauswertung war überaus fehleranfällig und ineffizient. Solange Computer vorrangig für wissenschaftliche und militärische Kalkulationen eingesetzt wurden, war dies noch tolerierbar. Doch dies änderte sich im Lauf der 1950er Jahre, als Computer zunehmend auch in betriebswirtschaftlichen Kontexten Einsatz fanden und man begann, die Einsatzmöglichkeiten von Computertechnologien zum Information Retrieval zu erkunden (vgl. Haigh 2009: 7).[3] Hierbei wurden Computer nicht mehr nur als Rechenmaschinen gebraucht, sondern sollten der Administration von Unternehmensprozessen und der Recherche in Informationssammlungen dienen. Infolgedessen wurde dem Problem der automatisierten Verwaltung von Informationen ein immer größerer Stellenwert beigemessen. Von maßgeblicher Bedeutung war in diesem Zusammenhang die Erfindung und Nutzbarmachung neuer Massenspeichertechnologien. Denn, so

3 | Larson identifiziert in seiner Einführung zum Panel »The Social Problems of Automation« auf der *Western Joint Computer Conference* 1958 drei Haupteinsatzgebiete von Computern: 1. Wissenschaft und Ingenieurwesen, 2. militärische Kontrolle, 3. betriebswirtschaftliche Datenverarbeitung sowie die industrielle Regelungs- und Steuerungstechnik (vgl. Larson 1958: 7).

stellte Jacob Rabinow – der Erfinder des *Notched-Disk Memory*, einem Vorläufer heutiger Festplatten – fest: »In all systems of automatic handling of information, one of the key elements is the device for storing of information« (Rabinow 1952: 745).[4]

Die zahlreichen Entwicklungsbemühungen führten unter anderem zur Einführung von Magnetbandspeichern im Jahr 1951 als Teil des UNIVAC I-Computersystems, das von J. Presper Eckert und John W. Mauchly entwickelt wurde (vgl. Williams 1997: 358ff.).[5] Auf Magnetbändern können relativ große Mengen digitaler Informationen dauerhaft gespeichert werden. Da Bandspeicher jedoch einer linearen Zugriffslogik folgen, haben sie einen entscheidenden Nachteil. Mit dieser Speichertechnologie ist die automatische Verwaltung großer Informationssammlungen ausgesprochen ineffektiv. Hierauf weist Louis N. Ridenour mit einem medienhistorischen Vergleich zwischen digitalen Speichertechnologien und unterschiedlichen Organisationsformen von Schrift hin:

»Two millennia ago human beings had just the same difficulties with scrolls – the ancients' counterpart of books. The scroll form was dictated by the need for protecting the edges of their brittle papyrus; the scroll left only two edges exposed. Shortly after the tougher parchment was introduced, the book form was invented, either in Greece or in Asia Minor. Called the codex, it was originated primarily for law codes, so that pages could be removed or added as statutes changed.
Reels of magnetic tape are the scroll stage in the history of computing machines. It remains for someone to invent the machine's analogue of the codex.« (Ridenour 1955: 100)[6]

4 | Zur Bedeutung von Rabinows *Notched-Disk Memory* für die Entwicklung der Festplatte bei IBM siehe Bashe (1986: 273ff.).

5 | Magnetspeicher wurden bereits Ende des 19. Jahrhunderts von Valdemar Poulsen erfunden. Poulsen verwendete Drähte zur Aufzeichnung von Tönen. Fritz Pfleumer hat in Anlehnung an diese Technologie in den 1920er Jahren das Tonband entwickelt (Pfleumer 1928). Der Einsatz von Magnetbändern als Speicher für Computer wurde jedoch erst seit Mitte der 1940er Jahre intensiv erkundet (vgl. Bashe et al. 1986: 187f.).

6 | Bereits drei Jahre bevor Ridenour die beschränkten Zugriffsmöglichkeiten sequentieller Speicher kritisierte, hat Rabinow eine ähnliche Kritik formuliert: »It has long been recognized that 3-dimensional storage of information, as in a book, utilizes space most efficiently. Previous three dimensional storage systems, however, have had the disadvantage of relatively long access time. The usual method of storing information in three dimensions has been to record it on a film strip, or on magnetic wire or tape. The difficulty is that the whole reel may have to be played in order to reach a particular bit of data; thus while the storage is 3-dimensional, the playback is either 1- or 2-dimensional, an is sequential« (Rabinow 1952: 745).

Nach Ansicht von Ridenour sind Magnetbandspeicher also mit Schriftrollen vergleichbar, da auf diese nur fortlaufend zugegriffen werden kann.[7] Will man an eine bestimmte Stelle in einer Schriftrolle oder auf einem Bandspeicher gelangen, so ist es nicht möglich, einfach dahin zu springen, vielmehr muss der gesamte Datenträger abgerollt werden, bis die gewünschte Stelle erreicht ist. Es handelt sich um sequentielle Speicher, die die Möglichkeit des Zugriffs auf eine Dimension beschränken, weshalb sie nur Schritt für Schritt, von vorn nach hinten (oder umgekehrt) durchsucht werden können. Infolgedessen ist der Zugriff auf den Datenträger sehr zeitaufwendig, was insbesondere dann zum Nachteil wird, wenn neben der Speicherung auch der schnelle Zugriff auf gespeicherte Informationen von Bedeutung ist.[8]

Im Unterschied zur Schriftrolle handelt es sich beim Kodex um eine Form der Verkörperung medialer Konstellationen, die den direkten Zugriff auf die Speicherinhalte ermöglicht. Das digitale Analogon zum Kodex, dessen Mangel Ridenour in seinem Artikel von 1955 noch beklagt, gelangte bereits ein Jahr später zur Marktreife und wurde von IBM als Teil des *305 RAMAC*-Systems vertrieben. Bezeichnet als *350 Disk Storage Unit*, wog die erste Festplatte mehr als eine Tonne und hatte eine Kapazität von gerade einmal fünf Megabyte (vgl. Kirschenbaum 2008: 76). Ebenso wie Kodizes ermöglichen Festplatten den Direktzugriff auf den Speicher, der als *Random Access Memory* bezeichnet wurde, was nicht mit der heute üblichen

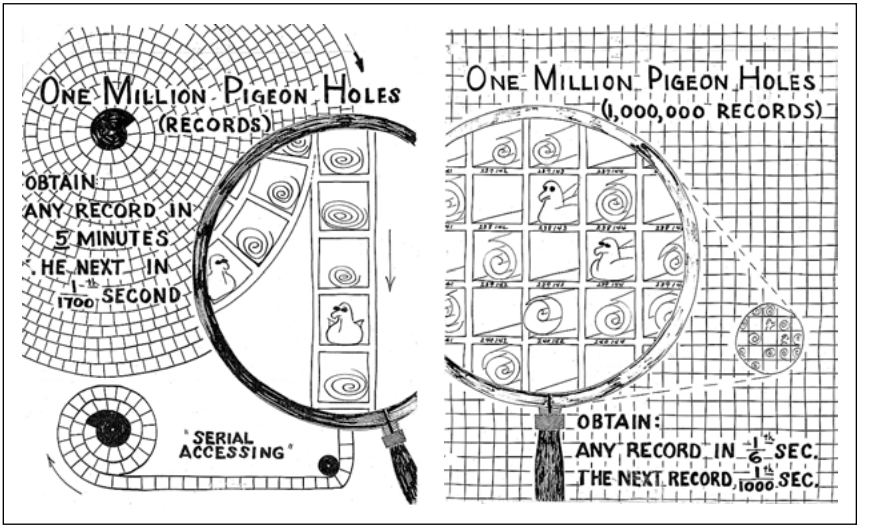
7 | Diese Kritik trifft in gewisser Hinsicht auch auf Lochkarten zu, da diese von Computern zumeist sequentiell verarbeitet wurden. Dennoch können Lochkarten nicht uneingeschränkt als sequentielle Speichertechnologie begriffen werden. Sind Lochkarten in Karteikästen angeordnet, dann können Nutzer wahlfrei auf diese zugreifen. Hierauf weist Kirschenbaum in seiner Beschreibung von Lochkarten als *Random Access Memories* hin (vgl. Kirschenbaum 2008: 82). Zudem sind Informationen auf einer einzelnen Lochkarte in zwei Dimensionen organisiert. Dahingehend ist die Lochkarte ebenfalls eher mit dem Kodex als mit der Schriftrolle vergleichbar. Dennoch sind Lochkarten im Kontext ihrer computertechnischen Verarbeitung eher als sequentieller Speicher zu betrachten. Die Verwaltung von Lochkarten in Karteikästen ist dem Computer äußerlich. In Computern werden Lochkarten hingegen ausschließlich in Stapeln verarbeitet, d.h. Computer können auf Informationen nur gemäß der Ordnung der Karten im Stapel und damit sequentiell zugreifen. Eine einzelne Lochkarte mag vor diesem Hintergrund zwar als Digitalspeicher mit Direktzugriff erscheinen, aber bei Lochkarten im Plural handelt es sich um eine sequentielle Speichertechnologie.

8 | Da die Speicherung auf sequentiellen Datenträgern durchaus effizient und preiswert möglich ist, werden Magnetbandspeicher beispielsweise heute noch immer in Anwendungsbereichen eingesetzt, in denen die persistente Langzeitspeicherung von Informationen im Vordergrund steht.

Bezeichnung für den Arbeitsspeicher digitaler Computer zu verwechseln ist.[9] Wahlfrei oder direkt ist der Zugriff auf Festplatten deshalb, weil alle Stellen im Speicher in konstanter Zeit erreicht werden können. Daher macht es keinen Unterschied, ob bestimmte Informationen am Anfang, in der Mitte oder am Ende stehen.

Random Access Memories wie der Kodex und die Festplatte erweitern den eindimensionalen Zugriff sequentieller Speicher auf zwei oder mehr Dimensionen. Im Kodex wird der ersten Dimension der Schrift(rolle) die zweite Dimension der Seite hinzugefügt; bei digitalen Datenträgern geht die lineare Anordnung binär-digital codierter Informationen in eine Flächenordnung über (vgl. Buchholz 1963: 89). Hieraus resultiert ein wichtiger Vorteil beim Zugriff auf Speicherinhalte, was die Gegenüberstellung beider Zugriffsverfahren von Charles Bachman veranschaulicht (vgl. Abb.5).

Abb. 5: Sequentieller und wahlfreier Datenzugriff



Quelle: Bachman 1962a

Um die Vorteile von nichtsequentiellen Speichertechnologien zu illustrieren, zieht Bachman das Beispiel von einer Million Ablagefächer heran, die entweder etwas beinhalten – hier sind es Tauben – oder nicht. Die ingenieurtechnisch höchstbri-

9 | Wenn im Folgenden vom *Random Access Memory* Festplatte gesprochen wird, dann wird damit ein Speicher mit Direktzugriff beschrieben. Auch der heute als *Random Access Memory* (RAM) bezeichnete Arbeitsspeicher von Computern ist ein Speicher mit Direktzugriff, der im Vergleich zu Festplatten weitaus kürzere Zugriffszeiten zu Informationen gestattet. Im RAM (Arbeitsspeicher) werden Informationen anders als bei Festplatten nicht persistent gespeichert, weshalb dieser in Computern als flüchtiger Primärspeicher verwendet wird.

sante Frage lautet: Wie lange dauert es, auf einen beliebigen, aber festen Eintrag zuzugreifen? Sequentielle Speicher benötigen dafür bis zu fünf Minuten, wohingegen Speicher mit Direktzugriff in nur einer Sechstel Sekunde auf jeden beliebigen Datensatz zugreifen können. Die dieser Gegenüberstellung zugrunde gelegten Zugriffszeiten sind ein Relikt der Technologien der 1960er Jahre und entsprechen heute keineswegs mehr der technischen Realität. An dem prinzipiellen Unterschied zwischen diesen Speicherformen hat sich jedoch nichts geändert, der sich in der Differenz zwischen Bandspeichern und Festplattenspeichern funktional niederschlägt. Nicht der Übergang vom Papierspeicher *Lochkarte* zum Magnetspeicher oder zu anderen elektronischen, aber weiterhin sequentiellen Speichertechnologien stellt die eigentliche Revolution in der Entwicklung neuer persistenter Speichermedien dar, sondern die Umstellung vom sequentiellen auf den direkten Speicherzugriff, der sich mit der Einführung und sukzessiven Durchsetzung von Festplatten vollzogen hat.

Medienhistorisch folgenreich ist die Erfindung der Festplatte unter anderem deshalb, weil die materiale Logik der Speicherung auf die Gebrauchs- und Umgangsformen mit Computern zurückwirkt. Durch Festplattenspeicher rückte der interaktive Umgang mit Computern in den Horizont des technisch Möglichen und wurde zu einer Entwicklungsaufgabe. Das in der Frühzeit der computertechnischen Informationsverarbeitung vorherrschende *Batch Processing* folgt der linearen Logik sequentieller Speicher. Kann auf Informationen nur in der Reihenfolge zugegriffen werden, in der sie auf dem Datenträger abgelegt sind, dann ist ihre Verarbeitung nur dann effizient möglich, wenn diese ebenso sequentiell erfolgt: »Data was stored on tape as a sequence of codes, and efficient processing was possible only when the tape was read from start to end with a minimum of rewinding or searching« (Haigh 2009: 7).[10] Eben dies leistet die Stapelverarbeitung, die dem Problem des linearen Zugriffs durch die gesammelte Durchführung einer Vielzahl gleichförmiger Verarbeitungsaufgaben begegnet. Es handelt sich um eine Strategie, die Beschränkungen des linearen Speicherzugriffs zu kompensieren. Dies wurde mit der Verfügbarkeit der Festplatte als Sekundärspeicher mit Direktzugriff hinfällig:

10 | Die Stapelverarbeitung gleichförmiger Informationsverarbeitungsaufgaben erschien aufgrund der beschränkten Zugriffsmöglichkeiten sequentieller Speichertechnologien als praktisch notwendig, war aber dennoch ineffizient. Verstärkt wurde der Nachteil sequentieller Speicher unter anderem dadurch, dass damalige Computer nur über einen relativ kleinen internen Hauptspeicher verfügten. Aufgrund dessen war es, wie Haigh herausgestellt hat, nur möglich, relativ einfache Verarbeitungsaufgaben in einem Speicherdurchgang abzuarbeiten: »Memory limitations, coupled with the inflexible, serial nature of tape storage, meant that a single major job might require dozens of programs to be run one after another, each reading and writing information from several tapes. Most of these programs processed intermediate results writing during earlier runs« (Haigh 2009: 7).

»However, if the restriction that file information is available only in a fixed, predetermined serial order is removed, the batching requirement is eliminated. If there is random access, with equal facility, to any place in the memory, it is possible to pick out directly the necessary items from each of the reference files in the sequence established by the order in which the input data is received and the secondary order established by the transaction.« (Lesser/Haanstra 1957: 140)

Dem Paradigma der Stapelverarbeitung setzt das IBM-Forscherteam in San Jose, das seit 1952 an der Entwicklung von Festplatten arbeitete, die Idee des *In-Line Processing* entgegen. Dies wird in der Betriebsanleitung des 305 RAMAC am Beispiel der manuellen Buchführung durch Buchhalter erläutert:

»Because the clerk has direct access to all of these accounts, he can complete the posting of each transaction before beginning the posting of the next. This accounting method is called in-line processing. In-line processing has previously not been practical in automatic accounting systems because of the difficulty of reaching and changing single records in large files.« (IBM 1957: 5)

Entscheidend an der *In-Line*-Verarbeitung von Information, für die später die Bezeichnung *Interactive Processing* geläufiger wurde, ist, dass komplexe Verarbeitungsaufgaben sofort und unabhängig von anderen Aufgaben abgearbeitet werden können (vgl. Schwartz 1968: 89f.).

Die computergestützte Lösung eines Problems, wie z.B. die Lohnabrechnung, erfordert die Durchführung einer Vielzahl einzelner Verarbeitungsschritte, welche u.a. die Abfrage der geleisteten Arbeitsstunden, die Abfrage der Personalinformationen, die Berechnung des Lohns sowie die Ausgabe individueller Lohnabrechnungen umfassen.[11] Im Modus der Stapelverarbeitung wird zuerst ein Teilverarbeitungsschritt für alle im *Batch* versammelten Aufgaben durchgeführt, bevor zum nächsten Verarbeitungsschritt übergegangen wird. Geht man davon aus, dass vier elementare Verarbeitungsschritte für eine Lohnabrechnung notwendig sind, dann werden diese im *Batch Processing* wie folgt ausgeführt: 1-1-1-2-2-2-3-3-3-4-4-4. Bei der interaktiven Informationsverarbeitung ändert sich dies. Sämtliche Schritte bei der Erstellung der Lohnabrechnung werden erst für eine Person abgearbeitet,

11 | Bashe et al. beschreiben die Lohnabrechnung in Unternehmen im Modus der Stapelverarbeitung wie folgt: »A payroll master file might consist of one punched card per employee, each record containing long-lived data such as name, employee number, location, pay rate, and so on. Typically, this file was kept in employee-number order. After weekly time cards, suitably punched, were sorted into masterfile order, processing runs brought together the data needed during preparation of pay checks and stubs, payroll registers, and accounting summaries. Payroll processing ran smoothly and efficiently because most of the work could be done weekly« (Bashe et al. 1986: 277f.).

bevor die Bearbeitung der nächsten Person erfolgt. Dementsprechend haben die Verarbeitungsschritte beim *Interactive Processing* die Reihenfolge: 1-2-3-4-1-2-3-4-1-2-3-4. Die Funktionseinheit bildet nicht mehr der einzelne Verarbeitungsschritt, sondern eine Prozedur, die unterschiedliche Verarbeitungsschritte zu einer komplexen Einheit zusammenfasst, welche als solche wiederholt werden kann.[12] Ob die genannte Aufgabe der Lohnabrechnung für eine, hundert oder tausend Personen durchgeführt werden soll, macht unter den Bedingungen sequentieller Speicher insofern keinen Unterschied, als es stets ungefähr dieselbe, relativ lange Zeit dauert, bis man zu einem Ergebnis gelangt ist. Speicher mit Direktzugriff ermöglichen demgegenüber den raschen Zugriff auf verteilt gespeicherte Informationen, sodass die einzelnen Verarbeitungsschritte schnell hintereinander ausgeführt werden können. Infolgedessen kann die Lohnabrechnung für einzelne Personen schnell und effizient erstellt werden.

Heute ist der interaktive Umgang mit Computern selbstverständlich. Ende der 1950er Jahre war Interaktivität kaum mehr als eine vielversprechende Vision und so war der Alltag der Computernutzung weiterhin vom Umgang mit Lochkarten und Bandspeichern geprägt und folgte der Logik der Stapelverarbeitung (vgl. Haigh 2009: 15). Denn die interaktive Verarbeitung von Informationen auf Grundlage von Festplatten erwies sich als schwierig. Voraussetzung für die Verwirklichung des *Interactive Processing* war die Automatisierung der Informationsverwaltung im Computer. Unmittelbar verfügten Festplatten nicht über dieses Leistungsvermögen. Was die Festplatte als *Random Access Memory* leistet, ist nicht der direkte Zugriff auf Information, sondern der direkte Zugriff auf den Datenträger. Daher sah man sich fortan mit der Frage konfrontiert, wie die Speicherung und Abfrage von Informationen realisiert werden kann, wenn der wahlfreie Zugriff auf jeden Ort im Speicher möglich ist. Die Beschäftigung mit diesem Problem führte zu der Entwicklung von zentralen Konzepten zur Funktionsweise digitaler Datenbanken, die in den Entwurf und die Implementierung von DBMS einflossen.

---

12 | Diese Form der Gruppierung von vielen Einzelverarbeitungsschritten zu einer komplexen Einheit kann im Anschluss an Paul Klee als *dividuell* bezeichnet werden. Dividuelle Reihungen unterscheidet Klee von individuellen Anordnungen: »Die Frage, ob dividuell oder individuell, wird entschieden durch unübersichtliche Ausdehnung oder übersichtliche Knappheit. Denn bei unübersichtlicher Ausdehnung können Teilungen willkürlich vorgenommen werden, ohne die vorliegende Gliederungsart zu stören. Bei übersichtlicher Knappheit aber kann nichts durch Teilung wegfallen und auch nichts hinzutreten, ohne das Individuum zu verändern oder es in ein anderes Individuum zu verwandeln« (Klee 1971: 237). Die Verarbeitungsfolge bei der Stapelverarbeitung ist individuell, da der Prozessfolge 1-1-1-2-2-2-3-3-3-4-4-4 kein Element entnommen werden kann, ohne dass dies Auswirkungen auf die gesamte Verarbeitungsprozedur hat. Die Reihenfolge der Prozessschritte im *Interactive Processing* (1-2-3-4-1-2-3-4-1-2-3-4) ist dividuell, da sich die Prozedur der Verarbeitungsschritte 1-2-3-4 wiederholt.

Folgenreich war die Erfindung der Festplatte folglich nicht nur wegen der durch diese Speichertechnologie eröffneten Möglichkeiten, sondern auch aufgrund der Probleme, die mit ihrem Gebrauch einhergingen. Festplatten bilden, wie Thomas Haigh in einer Studie zur Vorgeschichte der DBMS herausgestellt hat, den materiellen Horizont, vor dem sich die konzeptuelle und technische Entwicklung konkreter Datenbanktechnologien vollzog:

»File management systems were designed around tape storage (although they were later widely used with disk drives). They worked efficiently when processing an entire file in sequence. In contrast, DBMSs were designed around disk storage. Instead of treating files in isolation, they worked on a database of multiple files, representing linkages between individual records within those files. Unlike file management systems, they could be used by application programs, remaining resident in memory to process data operations.« (Haigh 2009: 13)

Nimmt man Haighs technikhistorische These medientheoretisch ernst, muss man nach den Herausforderungen fragen, die sich mit der Einführung von Festplatten verbanden. Kurz gesagt bestehen diese in der Adressierbarkeit von Information und der automatischen Verwaltung der Speicherinhalte.

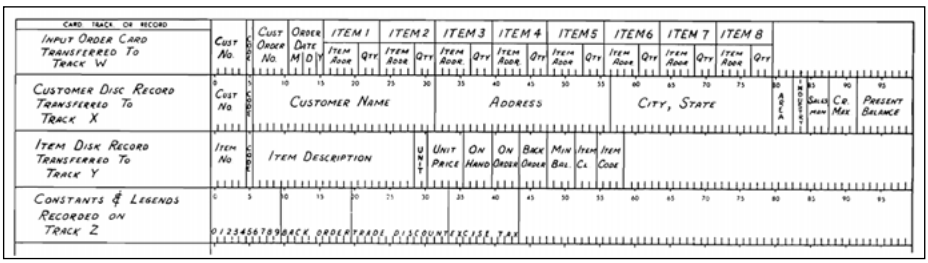
Der lineare Zugriff sequentieller Datenträger macht eine geordnete Speicherung von Informationen notwendig. Entscheidend ist dann nicht die absolute physikalische Adresse von Informationseinheiten, sondern ihre relative Position im Gesamtgefüge der in einem *File* gespeicherten Informationen. Die physische Anordnung der Informationen im Speicher ist dabei eng an deren logische respektive semantische Ordnung geknüpft.[13] Infolgedessen erwies sich auch das Speichern von Informationen auf Magnetbändern als aufwendig, da diese entsprechend der logischen Ordnung der Informationssammlung in die materielle Ordnung des Speichers eingefügt werden mussten: »[S]equential file systems like telephone

13 | Wie Buchholz herausstellt, verfügen Bandspeicher über keine vorgegebene Adressstruktur: »Magnetic tape has one very important characteristic; it is essentially continuous along its recording dimension. Except at the end of a reel, there are as a rule no physically predetermined starting and stopping points. Consequently there are no predetermined locations for writing data and there are no restrictions on the length of a block of data being written. The end of a block may simply be marked by a gap. The next time the tape is written the gaps may be somewhere else. On reading tape one can only ask for the next block. Except for the limited technique of counting gaps as the pass by, there is no way to select a specific record without inspecting each record in sequence« (Buchholz 1963: 89f.). Infolgedessen wurden Informationen nicht anhand ihres physischen Orts im Speicher, sondern anhand eines Schlüssels adressiert, der auf dem Datenträger ausfindig gemacht werden muss, um zu einer bestimmten Information zu gelangen.

directories and conventional tape systems must be completely rewritten to incorporate a single addition directly into the file« (Heising 1958: 195).[14]

Mit dem wahlfreien Zugriff auf Festplatten wird die lineare Speicherlogik sequentieller Speichertechnologien obsolet. Der Ort der Informationen im Speicher und die logische respektive semantische Ordnung von Informationssammlungen werden voneinander entkoppelt. Diese Möglichkeit beruht auf der Voraussetzung, dass jede Stelle im Speicher mit einer eindeutigen Adresse versehen ist. Die *350 Disk Storage Unit* ist beispielsweise in 50.000 Datensätze unterteilt, die jeweils hundert Zeichen beinhalten können, wobei für jedes Zeichen sieben Bit reserviert sind. Eindeutig angesprochen werden kann jeder Datensatz der Festplatte durch Angabe der Plattennummer (00-49), der Spurnummer (00-99) sowie der Nummer des Sektors (0-9). Das Wissen um die physikalische Adresse ist die Bedingung dafür, auf eine bestimmte Information direkt zugreifen zu können. Dann und nur dann kann der Schreib-Lesekopf direkt an die entsprechende Stelle springen und die gewünschte Information auslesen.

Abb. 6: Beispiel für die Planung der Speicherbelegung eines Speichertracks



Quelle: IBM 1957: 17

Der Direktzugriff löst das Problem des Herausfindens, welche konkrete Information an einem bestimmten Ort im Speicher hinterlegt ist. Dies führt jedoch zu dem Anschlussproblem, wie gewusst werden kann, wo genau eine bestimmte Information auf dem Datenträger gespeichert ist. Bevor dieses Problem technisch gelöst wurde, mussten die Nutzer der Festplatte die Belegung des Speichers planen, wie eine

14 | Auch Buchholz weist darauf hin, dass Änderungen an Magnetbandspeichern schwierig sind: »To change anything on a tape, every block must be read, whether changed or not, and rewritten on another tape« (Buchholz 1963: 90). Demgegenüber ermöglichen traditionelle Ablagesysteme, wie z.B. Karteikästen, das relative einfache Einfügen und Entfernen von Informationen: »Most manual filing systems have a very important characteristics that is so obvious that it is often overlooked; additions, changes, and deletions to the file are simple operations. Records can be inserted, replaced, or thrown away without affecting the rest of the file. This characteristic is extremely desirable in any system of file organization. Manual [...] systems readily permit additions and deletions« (Heising 1958: 195).

Anleitung zur Speicherplanung aus dem Benutzerhandbuch des *305 RAMAC*-Computersystems zeigt (vgl. Abb. 6). Die Verwaltung des Festplattenspeichers war zunächst den Nutzern überlassen. Sie mussten die Anordnung der Informationen im Speicher planen und koordinieren. In dieser Hinsicht war der Umgang mit Festplatten der gewohnten Praxis mit Lochkarten nicht unähnlich. Die Festplatte wurde analog zu den Dateien der Lochkartenära vom Nutzer als »disk file« (IBM 1958: 7) oder »Random Access File« (IBM 1959: 7) administriert. Die Notwendigkeit der nutzerseitigen Verwaltung des Festplattenspeichers stand den antizipierten Möglichkeiten von Festplatten diametral entgegen. Daher galt es, die logische Ordnung von Informationen automatisch mit deren materieller Anordnung im Adressraum des Speichers zu koordinieren. Diese Herausforderung wurde 1957 von W. Wesley Peterson als *addressing problem* bezeichnet:

»Whenever a file of records is stored in a data-processing system, some procedure must be devised for deciding where to store each record and for locating a stored record, given its identification number. Such a procedure will be called *addressing system*. The addressing system should make the average access time, i.e., the average time required for obtaining a record, as small as possible.« (Peterson 1957: 131)

Eine Lösung des Adressierungsproblems stellen Dateisysteme dar, die es ermöglichen, dass Nutzer nicht mehr physische Orte im Speicher, sondern logische Dateien adressieren. Wo und wie eine Datei auf dem Datenträger materiell abgelegt ist, wird damit für den Nutzer irrelevant.[15] Kennen muss dieser nur den Ort im logischen Raum des Dateisystems, der hierarchisch strukturiert ist. Aus der Festplatte als »disk file« (IBM 1958: 7) wird durch Dateisysteme ein Speicher für Dateien. Ermöglicht wird dadurch die automatische Verwaltung *von* und der automatische Zugriff *auf* Dokumente.

Dateisysteme entlasten die Nutzer von der Organisation der Informationen im physischen Adressraum des Speichers, indem sie es erlauben, die gespeicherten Information in logische Einheiten, d.h. Dateien, zu segmentieren und ihnen einen Platz in einer hierarchischen Verzeichnisstruktur zuzuweisen. Welche konkreten Informationen in welchen Dokumenten enthalten sind, hierüber gibt das Dateisystem keine Auskunft. An dieser Stelle bzw. diesem Problem setzt die Entwicklung digitaler Datenbanktechnologien und ihrer Vorläufer, wie z.B. Reportgeneratoren, an (vgl. Haigh 2009: 9ff.). Ihr Ziel ist es, die Adressierung von Informationen *als* Informationen zu ermöglichen und damit das Vergessen des physischen und

15 | Die materielle Anordnung von Informationen im Speicher ist infolgedessen für den Nutzer einzig unter Performancegesichtspunkten relevant. Sind die Dateien auf einem Datenträger stark segmentiert, dauert der Zugriff länger als bei kontinuierlich gespeicherten Dateien.

logischen Orts der Informationen im Speicher.[16] In einem 1962 verfassten Artikel über den *Integrated Data Store*, einem Vorläufer heutiger Datenbankmanagementsysteme, beschreibt Bachman dies als die Übersetzung einer elementaren Funktionalität von Festplatten in das Leistungsvermögen von Datenbanksystemen: »[T]he mass memory's ability to retrieve any specified record is translated into the ability to retrieve exactly the information needed to solve a problem« (Bachman 1962b: 3).

Wenn Computer die von Bachman beschriebene Übersetzungsleistung vollbringen, werden sowohl die Endnutzer von Datenbanken als auch Anwendungsprogramme von der Frage der Organisation der Informationen im Speicher entlastet; sie können unmittelbar mit digitalen Informationen umgehen, sie abfragen, modifizieren und speichern. Hierdurch gewinnen die im Computer verwalteten Informationen zunehmend an Autonomie gegenüber ihrem Gebrauch in unterschiedlichen Anwendungskontexten. Die mit diesem Ziel verbundenen technischen Herausforderungen begannen sich vor dem Hintergrund der Einführung der Festplatte als einem persistenten Sekundärspeicher mit Direktzugriff jedoch erst abzuzeichnen. Im computerwissenschaftlichen Datenbankdiskurs werden sie unter dem Stichwort *Datenunabhängigkeit* verhandelt, welche heute ein zentrales Leistungsmerkmal von Datenverwaltungssystemen ist.

Datenunabhängigkeit als Entwicklungsaufgabe

Als der Begriff der Datenunabhängigkeit Ende der 1960er Jahre gebräuchlich wurde, erfreute er sich rasch wachsender Popularität in der Entwicklergemeinschaft. Er bezeichnet eines der »nagging problems« (Bachman 1974: 17), mit denen man sich bei der Entwicklung von Informationsverwaltungssystemen konfrontiert sah. Die Debatte über und die Arbeit an Datenunabhängigkeit trug dabei der Tatsache Rechnung, dass Informationsbedürfnisse nicht statisch sind. Dies hat Auswirkungen auf die Datenbanksysteme, die der Speicherung und Abfrage von Informationen dienen. Sich wandelnde Anforderungen machen Änderungen an der Datenbank notwendig. Ist der Gebrauch von Informationen abhängig von deren Verwaltung in der Datenbank, führt jede Modifikation des Systems dazu, dass sämtliche Anwendungsprogramme, die auf die Datenbank zugreifen, um deren Informationen zu verarbeiten, an die neuen Strukturen angepasst werden müssen. Dies zu verhindern ist eines der unter dem Begriff Datenunabhängigkeit verhandelten Ziele.

16 | Die Suche in traditionellen Sammlungen, wie z.B. in dem Bestand einer Bibliothek, ist stets eine Suche im doppelten Sinn. Erstens gilt es herauszufinden, welche Bücher verfügbar sind, und zweitens müssen diese im Bibliotheksbestand aufgefunden werden: »[T]he first search yields only an address, and a second is required to reach the information« (Benson-Lehner Corporation 1959). Die Festplatte erlaubt das schnelle Auffinden von Informationen im Adressraum des Speichers. Das Herausfinden der Adresse der Information stellt jedoch eine Herausforderung dar, die durch Datenbanktechnologien gelöst werden soll.

Kopfzerbrechen bereitete die Gewährleistung von Datenunabhängigkeit jedoch nicht nur in technischer Hinsicht. Unklar war zunächst, was genau mit Datenunabhängigkeit gemeint sei. Da die Datenbankforschung zu dieser Zeit »more an empirics than a theory« (Liu 2008: 250) war, erwuchs die Forderung nach Datenunabhängigkeit nicht aus einem präzise formulierten theoretischen Problem, sondern aus einer Vielzahl unterschiedlicher praktischer Aufgabenstellungen.[17] Datenunabhängigkeit war daher zunächst kaum mehr als eine Modewort, das sehr uneinheitlich gebraucht wurde, weshalb Bachman 1974 kritisch konstatierte, der Begriff sei einer der »least precise terms now extant in the Database Management field« (Bachman 1974: 19).[18] Ungeachtet dieser Kritik verdichteten sich in dem mehrdeutigen Gebrauch des Begriffs die vielfältigen Probleme der computertechnischen Handhabung großer Informationssammlungen zu einer programmatischen Entwicklungsaufgabe.

Im Hintergrund der Forderung nach Datenunabhängigkeit stand der ökonomisch motivierte Wunsch, die Gebrauchslogiken von Informationen und die technische Logik ihrer Verwaltung voneinander zu entkoppeln und beide gegeneinander abzuschirmen.[19] Edgar F. Codd hat dieses Ziel in dem 1970 publizierten Entwurf des relationalen Datenmodells in der Forderung auf den Punkt gebracht: »Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation)« (Codd 1970: 377). Der Einsatz von Computern in ökonomischen, militärischen und wissenschaftlichen Kontexten war in den 1960er und 70er Jahren noch immer sehr zeitaufwendig und kostspielig. Software im heutigen Sinne gab es noch nicht (vgl. Haigh 2002). Nahezu alle Computerprogramme mussten von den Nutzern selbst implementiert werden und waren zumeist nicht auf anderen Computern oder Computersystemen

17 | Auch Haigh weist darauf hin, dass die Entwicklung von Datenbanktechnologien zunächst weniger ein akademisches Unterfangen war: »[D]atabase tools evolved from the daily work of the data processing staff, grappling with tight schedules and working intimately with simple hardware to tackle ambitious assignments« (Haigh 2009: 6).

18 | Bachmans Diagnose ist umso bemerkenswerter, wenn man bedenkt, dass während dieser Zeit eine Vielzahl unterschiedlicher und vor allem unklarer Terminologien im Datenbankdiskurs virulent waren. Symptomatisch hierfür ist die Bemerkung, mit der Albert C. Patterson 1971 einen Vortrag zu DBMS einleitete:»Each data base presentation offers a distinct new jargon to our already over-developed Tower of Babel and it is not at all clear that this presentation will violate that rule« (Patterson 1971: 197).

19 | Die so verstandene Datenunabhängigkeit ist ein zentrales Leistungsmerkmal heutiger DBMS, wie Haigh unterstreicht: »Eine der wichtigsten Eigenschaften des Datenbankmanagementsystems besteht darin, die Personen und Programme, die diese Daten benutzen, von den Details ihrer physischen Speicherung abzuschirmen« (Haigh 2007: 57).

lauffähig. Zu dieser relativen Statik von Computeranwendungen steht sowohl die Dynamik der Gebrauchskontexte, in denen die Computer zum Einsatz kommen, in einem Spannungsverhältnis als auch die rasche Entwicklung immer leistungsfähigerer Hardware. In einer Welt sich wandelnder Anforderungen an Technik und sich wandelnder Technologien gilt es, wie Jardine auf einer SHARE-Konferenz im Juli 1973 festgestellt hat, die geleisteten Investitionen in Daten und Softwareanwendungen zu schützen.[20] Das Streben nach Datenunabhängigkeit sei motiviert durch den Wunsch nach »[p]rotection of investment in data & programs in a changing business & computing environment« (Jardine 1973: 2). Ermöglicht werden soll somit die Fortentwicklung von Datenbanksystemen, ohne dass die Integrität der Daten und die Funktion der Anwendungsprogramme beeinträchtigt wird:

»Neither usage nor maintenance of the stored data can be independent of the enterprise's requirements. Data independence permits each to be independent of each other, while responding to the business requirements. Programs should not be subject to impact of influences external to themselves. Data independence insulates a user from the adverse effects of the evolution of the data environment.« (Study Group on Data Base Management Systems 1975: II-29)

Der Gebrauch und die Verwaltung von Informationen sollen voneinander entkoppelt und gegeneinander abgeschirmt werden, um zu verhindern, dass sich Veränderungen an einer Datenbank auf die Funktion bereits existierender Anwendungsprogramme auswirken. Diese Änderungen können sich mindestens auf zwei Ebenen manifestieren: der logischen Ebene der Datenstrukturen und der physischen Ebene der Speicherung. Letztere betrifft die Anordnung von Informationen auf digitalen Datenträgern. Änderungen an dieser Ordnung sollen keine Auswirkungen auf die Abfrage von Informationen haben, d.h. Endnutzer und Anwendungsprogramme sollen in der Lage sein, auf bestimmte Informationen zuzugreifen, ganz gleich, wo diese im Speicher abgelegt sind (vgl. Date/Hopewell 1971b). Um dies zu gewährleisten, sei Codd zufolge eine »clear distinction between order of presentation on the one hand and stored ordering on the other« (Codd 1970: 378) einzuführen. Von der physischen Datenunabhängigkeit, welche die materielle Verkörperung von Informationen im Speicher betrifft, unterscheiden Date und Hopewell die logische Datenunabhängigkeit (vgl. Date/Hopewell 1971a). Wird es notwendig, die Struktur der Datenbank zu verändern, um nicht nur neue Informationen zu einer Datenbank hinzuzufügen, sondern genuin neue Formen von Informationen in diese

20 | SHARE ist eine 1955 gegründete IBM-Nutzergruppe. Als Zusammenschluss von IBM-Computernutzern diente SHARE dem gemeinsamen Wissens- und Kompetenzaustausch sowie der kollaborativen Entwicklung von Betriebssystemen und Anwendungsprogrammen für IBM-Computer. In dieser Hinsicht weist SHARE Parallelen zur heutigen Open Source-Bewegung auf (vgl. Haigh 2002: 8; 2009: 11).

aufnehmen zu können, soll dies ebenfalls keine Auswirkungen auf existierende Anwendungen haben.[21]

Mit der Entwicklung immer komplexerer Informationssysteme – deren Datenbestände verteilt auf mehreren Computern gespeichert und verwaltet werden oder deren semantische Integrität durch logische Konsistenzregeln abgesichert wird – entstehen weitere Abhängigkeiten und somit neue Herausforderungen für Datenunabhängigkeit. Daher unterscheidet Codd neben der physischen und der logischen Datenunabhängigkeit zwei weitere Formen von Unabhängigkeiten, welche technisch zu gewährleisteten sind: die »Distribution Independence« und die »Integrity Independence« (vgl. Codd 1990: 345ff.).[22] An dieser Erweiterung wird deutlich, dass Datenunabhängigkeit kein absoluter Zustand ist, sondern von den konkreten technischen Praktiken der Versammlung, Verwaltung und Verarbeitung digitaler Information(ssammlung)en abhängt.[23]

---

21 | Die Unterscheidung zwischen logischer und physischer Datenunabhängigkeit ist keineswegs trennscharf. Beispielsweise unterscheidet Codd in *A Relational Model of Data for Large Shared Data Bank*s drei Formen von Datenabhängigkeiten: die »Ordering Dependence«, die »Access Path Dependence« sowie die »Indexing Dependence« (Codd 1970: 378f.). Während die ersten beiden Abhängigkeiten in der Unterscheidung zwischen logischer und physischer Datenunabhängigkeit aufgehen, ist die Forderung nach der Unabhängigkeit von Datenbankindexen weder eindeutig der Seite der Physik noch der Logik zuzuordnen. Bei der Verwaltung großer Datenmengen wurde früh von Indexen Gebrauch gemacht, um die Performanz des Systems bei der Abfrage bestimmter Informationen zu erhöhen, indem alternative Ordnungen der in der Datenbank enthaltenen Informationen erzeugt werden. Erweisen sich Indexe bei der Informationsabfrage einerseits als vorteilhaft, zieht ihr Gebrauch andererseits Nachteile beim Einfügen und bei der Modifikation von Informationen nach sich. Daher bleibt ihr praktischer Einsatz stets ein Kompromiss, was zur häufig wechselnden Indexen führt. Hieraus ergibt sich die Forderung, dass Anwendungen unabhängig von den aktuell vorhandenen Indexen die gewünschten Informationen abfragen können und zugleich durch das DBMS in die Lage versetzt werden sollen, diese so schnell wie möglich zu finden (vgl. Codd 1970: 378).

22 | Als »Integrity Independence« bezeichnet Codd die Gewährleistung der logischen Integrität einer Datenbank, die nicht durch Nutzer, sondern durch das DBMS sicherzustellen sei. Die »Distribution Independence« betrifft Herausforderungen, welche bei der verteilten Speicherung von Datenbanken auftreten. In diesem Zusammenhang sei zu gewährleisten, dass Informationsbestände in verteilten Datenbanksystemen reorganisiert werden können, ohne dass dies Auswirkungen auf die Nutzer bzw. Anwendungsprogramme hat (vgl. Codd 1990: 345ff.). Unter einer verteilten Datenbank versteht man eine Datenbank, deren Informationsbestand auf mehreren unabhängig voneinander operierenden Computern verteilt und verwaltet wird.

23 | Die Relativität der Datenunabhängigkeit zeigt sich darüber hinaus auf einer weiteren Ebene, die Stonebraker in seiner Kritik an der von Date und Hopewell

An der Diskussion und den Entwicklungsbemühungen um Datenunabhängigkeit wird deutlich, dass digitale Informationen keine autonomen Entitäten sind, die rein immateriell oder virtuell sind und die beliebig von Computern verarbeitet werden können. Sofern die in Datenbanken gespeicherten Informationen ein gewisses Maß an Autonomie aufweisen, ist diese Autonomie ein Effekt der Informationssysteme, in denen die Informationen versammelt sind und verwaltet werden. Kurzum: Autonomie ist keine Wesenseigenschaft digitaler Information, sondern Leistungsmerkmal von digitalen Informationssystemen. Wie im Folgenden zu sehen sein wird, boten die seit Ende der 1960er Jahre entwickelten abstrakten Datenbankarchitekturen eine konzeptuelle Lösung des Problems der physischen und logischen Datenunabhängigkeit.

Datenbankmodelle: Architekturen für Datenunabhängigkeit

Die medien- und kulturwissenschaftliche Auseinandersetzung mit digitalen Datenbanktechnologien fokussierte bisher einseitig die Datenmodelle, mit denen DBMS operieren, d.h. die konkreten Verfahren, mit welchen Daten modelliert und verarbeitet werden. Aus einer technikhistorischen Perspektive rekonstruiert Gugerli beispielsweise den in den 1970er Jahre schwelenden Disput zwischen Vertretern des Netzwerkmodells und des relationalen Datenmodells (Gugerli 2007a, 2009). Und Krajweski unterzieht das relationale Datenmodell einem medientheoretischen Vergleich mit dem objektorientierten Modell der Datenverwaltung (Krajewski 2007). Auch wenn Datenmodelle im Kontext digitaler Datenbankentechnologien zweifellos wichtig sind, soll im Folgenden zunächst die zentrale Bedeutung von Datenbankarchitekturen herausgearbeitet werden, welche mit dem Ziel entwickelt wurden, das Problem der Datenunabhängigkeit zu lösen. Maßgeblich waren die Vorschläge der *Conference on Data Systems Languages* und der *ANSI/X3/SPARC Study Group on DBMS*.

---

eingeführten Unterscheidung zwischen logischer und physischer Datenunabhängigkeit herausarbeitet (Stonebraker 1974). Seines Erachtens sind die möglichen Änderungen, die an der physischen Ordnung sowie der logischen Struktur einer Datenbank vorgenommen werden können, nicht gleich zu behandeln. Vielmehr seien triviale von komplexen Fällen zu unterscheiden. Während Datenunabhängigkeit bei trivialen Änderungen leicht gewährleistet werden könne, sei dies bei komplexen Modifikationen nicht der Fall. Nach Ansicht von Stonebraker ist es möglich, in der Theorie DBMS zu konzeptualisieren, die Nutzer von komplexen Änderungen an einer Datenbank abschirmen. Hierdurch werde das System seines Erachtens jedoch »hopelessly complex« (Stonebraker 1974: 64), sodass die praktische Realisierung von derartigen DBMS impraktikabel wird. Deshalb macht sich Stonebraker notwendigerweise für eine graduelle Sicht auf Datenunabhängigkeit stark.

Von der Datenbank zur Datenbankarchitektur

Die Conference on Data Systems Languages (CODASYL) wurde 1959 mit dem Ziel gegründet, eine Programmiersprache zu entwickeln, die sich zur Implementierung datenintensiver Anwendungen eignet, wie sie insbesondere in betriebswirtschaftlichen Kontexten vorkommen (vgl. Sammet 1985: 289). Ein Ergebnis dieses Zusammenschlusses von Vertretern aus der Wissenschaft, der Computerwirtschaft sowie von Computernutzern war die Spezifikation der Programmiersprache COBOL, die ihren primären Anwendungszweck bereits im Namen trägt, der die Abkürzung für Common Business Oriented Language ist. Darüber hinaus wird CODASYL heute vor allem mit einer Reihe wichtiger konzeptioneller Entwicklungen im Bereich digitaler Datenbanktechnologien in Verbindung gebracht.

Die dezidierte Hinwendung zu Datenbanken wurde 1965 angestoßen, als das COBOL-Programmiersprachenkomitee auf Initiative von Warren Simmons, einem Vertreter der Firma US Steel, eine Arbeitsgruppe einsetzte, die sich mit der Erweiterung von COBOL um Funktionalitäten zur Handhabung großer Datensammlungen, d.h. Datenbanken, befassen sollte (vgl. Olle 1978: 3). Zunächst als List Processing Task Force gegründet, benannte sich die Gruppe 1967 in Data Base Task Group (DBTG) um. Zwischen 1968 und 1971 legte die Arbeitsgruppe eine Reihe von Berichten vor, in denen sie die Ergebnisse ihrer Tätigkeit präsentierte. Als wirkmächtig erwies sich vor allem die funktionale Spezifikation von Datenbankmanagementsystemen, die nicht zuletzt auch die Anerkennung digitaler Datenbanken als einem eigenständigen Forschungsfeld in der Informatik befördert hat (vgl. CODASYL Data Base Task Group 1968, 1969, 1971).[24] DBMS werden als eigenständige Software-Hardware-Konfigurationen konzipiert, die Informationssammlungen unabhängig im Computer verwalten und diese als Informationsbestand anderen Anwendungen zur Verfügung stellen.[25] Hierdurch wird die programmgesteuerte Auswertung und der nutzerseitige Umgang mit digitalen Informationen von ihrer Verwaltung im Computer abgetrennt, sodass Datenbanktechnologien weitgehend losgelöst von Fragen der Softwareentwicklung respektive Programmiersprachen entwickelt werden können.

Die Entkopplung von Datenauswertung und Datenverwaltung spiegelt sich auch in der zweiten wichtigen Neuerung wider, welche die CODASYL-DBTG eingeführt hat. Vorgeschlagen wurde die strikte Unterscheidung von zwei Arten von (Program-

24 | Ein Indiz für die wachsende Anerkennung von Datenbankproblemen ist Haigh zufolge die Verleihung des *Turing Award* an Charles Bachman im Jahr 1973 (vgl. Haigh 2007: 81)

25 | Auch wenn unter DBMS heute zumeist Softwareanwendungen verstanden werden, sollte der Aspekt der Hardware nicht vernachlässigt werden. DBMS wurden und werden häufig in Kombination mit speziell für Datenbankverwaltungsaufgaben optimierten Computersystemen, sogenannten Datenbankmaschinen, vertrieben und eingesetzt.

mier-)Sprachen, die DBMS bereitstellen müssen: eine Data Description Language (DDL) einerseits und eine Data Manipulation Language (DML) andererseits. Während die DDL der strukturellen Beschreibung von Informationsbeständen und damit der internen Verwaltung von Datensammlungen dient, stellt die DML elementare Operationen zur Verfügung, um Daten in eine Datenbank einzufügen, sie abzufragen, zu ändern und zu löschen.[26] Die DML zielt nach außen und bildet die Schnittstelle, mittels derer andere Anwendungsprogramme und Nutzer auf Datenbanken zugreifen können.[27]

Schließlich hat die *Data Base Task Group* das Netzwerkdatenmodell formuliert, auf dessen Grundlage die strukturelle Beschreibung von Datenbanken möglich ist. Bei dem als Alternative zu hierarchischen Modellierungsverfahren entwickelten Datenmodell handelt es sich sicherlich um den bekanntesten Beitrag von CODASYL zur Theorie digitaler Datenbanksysteme. Bekannt ist das Netzwerkdatenmodell aber vor allem deshalb, weil es sich in der Folgezeit nicht gegen das relationale Datenmodell durchsetzen konnte. Diese zweifelhafte Prominenz erweist sich in medienhistorischer Hinsicht durchaus als problematisch, da mit dem Scheitern des Netzwerkdatenmodells tendenziell die Wirkung aus dem Blick geraten ist, die die Arbeiten der CODASYL-DBTG auf die spätere Datenbankentwicklung hatten.[28]

Fortbestand hatte nicht das Netzwerkdatenmodell, sondern die grundlegende Idee von DBMS, die auf dem Ziel beruht, die technische Informationsverwaltung von der prozeduralen Informationsverarbeitung zu entkoppeln. Konzeptionell schlägt sich dies zum einen in der bereits erwähnten Unterscheidung von DDL und DML nieder, deren Verhältnis zueinander wie folgt bestimmt wurde: »The relationship

26 | Der Unterschied zwischen der Datendefinitions- und der Datenmanipulationssprache wurde im Zwischenbericht der CODASYL-DBTG wie folgt erklärt: »The DDL is the language used to declare a SCHEMA. A SCHEMA is a description of a DATABASE, in terms of the names and characteristics of the DATA-ITEMS, RECORDS, AREAS, and SETS included in the database, and the relationships that exist and must be maintained between occurrences of those elements in the database. […] The DML is the language which the programmer uses to cause data to be transferred between his program and the database« (CODASYL Data Base Task Group 1969: 2-1).

27 | Ein Vorteil der Unterscheidung zwischen DDL und DML ist nach Ansicht der Mitglieder der CODASYL-Arbeitsgruppe, dass die Definition der Datenstruktur in der Datenbank unabhängig ist von den Programmiersprachen, mit denen Anwendungen für die Datenbank implementiert werden: »The specification of separate Data Description and Data Manipulation Languages is significant in that it allows databases described by the Data Description Language to be independent of the host languages used for processing the data. Of course, for this to be possible, the host language processors must be able to interface with such independent descriptions of data« (CODASYL Data Base Task Group 1969: i).

28 | Auf das Netzwerkdatenmodell wird im Unterkapitel »Data + Access« (S. 245ff.) noch näher eingegangen.

between the DDL and DML is the relationship between declarations and procedure« (CODASYL Data Base Task Group 1969: II-2). Auf der Seite der strukturellen Beschreibung des Datenbestands manifestiert sich dieses Ziel zum anderen in der Unterscheidung zweier Beschreibungsebenen und damit zweier Sichten auf digitale Informationen, die als Schema und Subschema bezeichnet werden. Das Schema beschreibt die Struktur aller in der Datenbank enthaltenen Informationen, wohingegen Subschemata Teilsichten auf den Datenbestand modellieren: »The concept of separate schema and sub-schema allows the separation of the description of the entire database from the description of portions of the database known to individual programs« (CODASYL Data Base Task Group 1969: II-5).

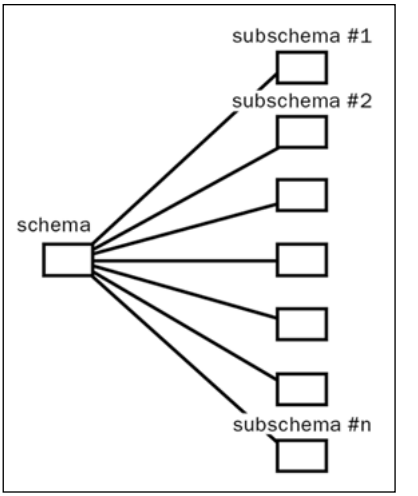
Die beiden Sichten auf Informationen, die in die Unterscheidung von Schema und Subschema eingeschrieben sind, spiegeln die unterschiedlichen Bedürfnisse zweier Interessengruppen wider, die in der DBTG aufeinander trafen, wie ein ehemaliges Mitglied der Gruppe später berichtet:

»The arguments which were raging during the years 1967 and 1968 reflected the two principle types of background from which contributors to the data base field came. People like Bachman, Dodd and Simmons epitomize the manufacturing environment [...]. Others, such as those who had spoken at the early 1963 SDC symposium, and indeed myself had seen the need for easy to use retrieval languages which would enable easy access to data by non-programmers. [...] One sentence by Bachman sums up what in retrospect could be the only conclusion with respect to the two approaches: ›Each has its place; both are necessary‹.« (Olle 1978: 3)

Der Disput zwischen Herstellern und Nutzern führt zu der 1969 eingeführten Zwei-Ebenen-Architektur von Schema und Subschema, die als Metamodell der Informationsmodellierung begriffen werden kann. Während Schema und Subschema der Beschreibung einer Informationssammlung respektive einer Teilsicht darauf dienen, modelliert die Unterscheidung zwischen beiden den Informationsfluss zwischen der unsichtbaren Tiefe des Computers und der Benutzeroberfläche, den das DBMS steuert. Das Schema dient der Verwaltung von Informationssammlungen und organisiert die Ablage der Informationen im Speicher: »The schema describes the database in terms of the characteristics of the data as it appears in secondary storage and the implicit and explicit relationship between data elements« (CODASYL Data Base Task Group 1969: 2-2). Dies bleibt für die Nutzer eines Informationsbestands unsichtbar. Sie können auf die Datenbank nur mittels eines Subschemas zugreifen, welches eine spezifische Sicht auf die Datenbank eröffnet. Während das Schema die Verwaltung von Informationen in der Tiefe des Computers modelliert, zielen Subschemata auf die Verarbeitung der Datenbankinformationen in verschiedenen Anwendungsprogrammen und deren nutzerseitigen Gebrauch an der Oberfläche. Die Unterscheidung beider Ebenen gewährleistet die Anschlussfähigkeit eines Informationsbestands an unterschiedliche Gebrauchskontexte, wie Abbildung 7 verdeutlicht. Der Gebrauch von Informationen an der Benutzerober-

fläche und ihre Verwaltung in der Tiefe des Computers unterliegen dabei unterschiedlichen Gebrauchslogiken und organisieren dieselben Informationen auf verschiedene Weise. Die Übersetzung zwischen beiden Ebenen soll durch das DBMS gewährleistet werden, was nach Ansicht der Mitglieder der DBTG ein gewisses Maß an Datenunabhängigkeit sicherstellt.

Abb. 7: Zwei-Ebenen-Architektur[29]



Änderungen im Datenbankschema, d.h. in der Art und Weise, wie Informationen im Speicher abgelegt sind, wirken sich nicht notwendig auf Anwendungsprogramme aus, da deren Sicht auf die gespeicherten Informationen durch das Subschema definiert wird, welches sich »in certain important aspects« (CODASYL Data Base Task Group 1969: II-5) vom Schema unterscheiden kann.[30] Insofern stellen Subschemata

29 | In den Berichten der CODASYL-DBTG findet sich noch keine Visualisierung der Zwei-Ebenen-Architektur. Diese Darstellung lehnt sich an Bachmans Vergleich zwischen der CODASYL- und der ANSI/X3/SPARC-Architektur an (vgl. Bachman 1975: 570).

30 | Die Autoren des Berichts führen neben der Datenunabhängigkeit zwei weitere Vorteile an, die die Zwei-Ebenen-Architektur ihrer Ansicht nach hat. So dient die Einführung von Nutzersichten auf Datenbanken auch der Entlastung von Programmierern und dem Schutz der Datensicherheit und der -integrität. Solange Informationen noch nicht in Datenbanken integriert gesammelt und verwaltet wurden, mussten Programmierer nur mit denjenigen Informationen umgehen, die für ihren konkreten Anwendungsfall relevant waren. Dies ändert sich durch Datenbanken, sodass sich Programmierer unter Umständen mit Informationen konfrontiert sehen, die für sie irrelevant sind. Deshalb werden durch die Definition von Subschemata Teilsichten auf den Datenbestand deklariert, die nur diejenigen

eigene Informationsmodelle dar, welche jedoch auf dem darüber liegenden Schema beruhen und mit diesem kompatibel sein müssen.[31]

Tertium Datur: Die ANSI/X3/SPARC-Datenbankarchitektur

Die Idee, verschiedene Sichten auf Informationen und damit einhergehend unterschiedliche Ebenen des Umgangs mit digitalen Informationen systematisch voneinander zu unterscheiden, wurde in der Entwicklergemeinschaft positiv aufgenommen. Jedoch stellte sich bald heraus, dass die Differenzierung von nur zwei Ebenen nicht hinreichend ist, um Datenbanken und die sie verwaltenden Systeme zu beschreiben. Die Differenzierung von Schema und Subschema wiederholt die doppelte Logik von unsichtbarer Repräsentation im digitalen Code und phänomenaler Präsentation an der Benutzeroberfläche, die allen digitalen Medienobjekten eingeschrieben ist (vgl. National Institute of Standards and Technology 1993: A2). Infolgedessen fällt in der CODASYL-Architektur die semantische Ordnung der zu speichernden Informationen mit ihrer Speicherordnung zusammen.

Im Schema sind zwei Sichten auf Information miteinander verschaltet und überlagern sich notwendig in dessen Definition. Um diesem Problem zu entgehen, wurde eine dritte Beschreibungsebene digitaler Informationen eingeführt: das konzeptuelle Schema. Der erste Entwurf einer Drei-Ebenen-Datenbankarchitektur wurde 1970 von der *Joint GUIDE-SHARE Data Base Requirements Group* in einem Bericht vorgelegt, der es zum Ziel hatte, die Anforderungen an ein Datenbankmanagementsystem zu spezifizieren.[32] Die Mitglieder der gemeinsamen GUIDE-SHARE-Arbeitsgruppe standen den Vorschlägen der CODASYL-DBTG zwar kritisch gegenüber (vgl. Haigh 2009: 18). Dennoch findet sich in ihrem Bericht auch die Unterscheidung verschiedener Sichten auf Informationen in Datenbanken, wobei der externen Sicht der Nutzer bzw. der Anwendungsprogramme (*Logical*

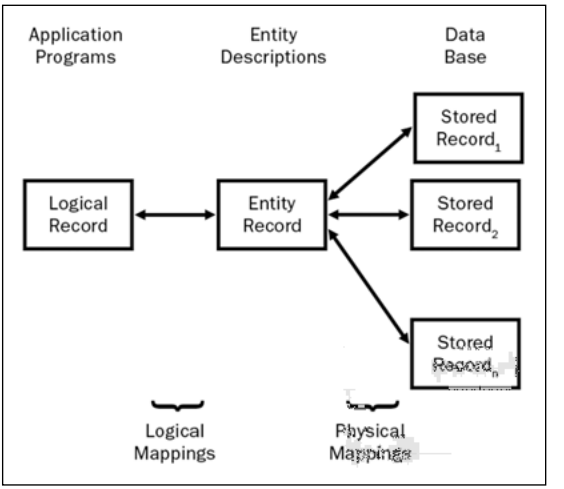
Informationen zugänglich machen, die gebraucht werden. Zudem birgt der freie Zugriff auf eine Datenbank die Gefahr, dass geschützte Informationen unerlaubt eingesehen oder gar verändert werden, weshalb die Definition von Nutzersichten auf Datenbanken auch zur Datensicherheit beiträgt (vgl. CODASYL Data Base Task Group 1969: II-5).

31 | Das Maß an Datenunabhängigkeit, welches durch die Differenzierung von Schema und Subschema gewährleistet wird, ist relativ gering. So hat Canning kritisiert, dass sich Änderungen im Schema durchaus auf ein Subschema auswirken können, was zur Konsequenz hat, dass eine Dependenz zwischen DDL und DML gibt. Kurzum: Anwendungsprogramme müssen bei Schemaänderungen angepasst werden (vgl. Canning 1972: 11)

32 | Neben SHARE war GUIDE (Guidance of Users of Integrated Data-Processing Equipment) eine zweite IBM-Nutzergruppe, die ebenfalls in den 1950er Jahren gegründet wurde. Während sicht GUIDE 1999 aufgelöst hat, besteht SHARE noch immer fort.

Record) und der materiellen Verkörperung von Informationen im Speicher (*Stored Record*) eine dritte Ebene hinzugefügt wird, welche die Autoren als *Entity Record* bezeichnen (vgl. Joint GUIDE-SHARE Database Requirements Group 1970: 3.2ff.). Dieser situiert sich zwischen der internen und der externen Sicht auf Information und dient der Vermittlung zwischen beiden (vgl. Abb. 8).

Abb. 8: GUIDE-SHARE-Datenbankarchitektur



Quelle: Joint GUIDE-SHARE Database Requirements Group 1970: 3.5

Die im GUIDE-SHARE-Bericht dargelegte Datenbankarchitektur hatte wenig Einfluss auf die weitere konzeptuelle Entwicklung von Datenbanksystemen. Eine Ursache hierfür ist, dass die Unterscheidung der drei Ebenen im Rahmen des Berichts nur eine marginale Rolle einnimmt, wobei insbesondere die Darlegung des *Entity Record*-Konzepts unterbestimmt bleibt, wie Canning in seinem 1972 erschienenen *Report on Database Management* kritisierte: »The third concept is entity data. Entity data is a not-well-defined term« (Canning 1972: 7).[33]

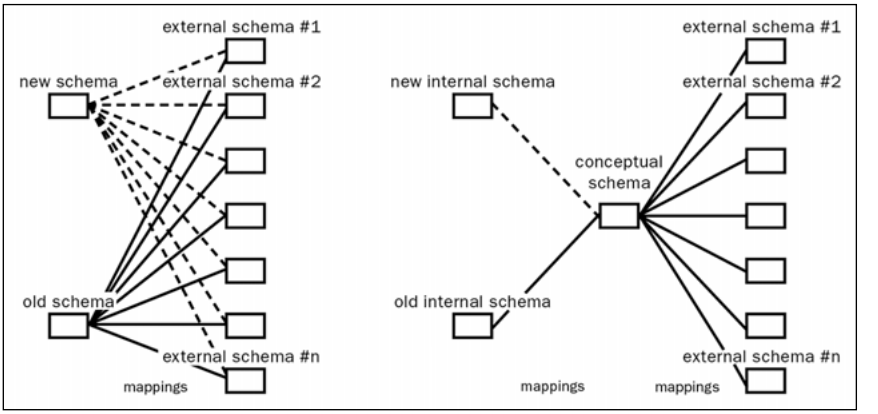
Im Anschluss an die Vorarbeiten der CODASYL-Arbeitsgruppe und der Joint GUIDE-SHARE Data Base Requirements Group hat die ANSI/X3/SPARC Study Group on DBMS eine Datenbankarchitektur entwickelt, die bis heute den konzeptuellen, aber auch idealisierten Rahmen bildet, in dem sich der Entwurf digitaler Datenbanken vollzieht. Gegründet wurde die Datenbank-Forschergruppe 1972 von der Abteilung *X3* (Computers & Information Processing) des *Standards Planning and Requirements Committee* (SPARC) im *American National Standards Institute* (ANSI). Die Gruppe hatte den Auftrag, Möglichkeiten für Standardisierungen im

33 | Eine weitere Kritik wurde von Everest und Sibley vorgebracht, die einwenden, dass das Modell nur eine externe Sicht auf digitale Datensammlungen vorsieht (vgl. Everest/Sibley 1971: 105).

Bereich von DBMS zu evaluieren (vgl. Tsichritzis/Klug 1978: 176). Ähnlich wie es bei CODASYL der Fall war, fanden sich in der Arbeitsgruppe Vertreter aus der Wissenschaft, der Computerwirtschaft und Anwender zusammen. Auch personell gab es eine Überschneidung zwischen der CODASYL-DBTG und der ANSI/X3/SPARC Study Group. So wirkte insbesondere Charles Bachman in beiden Arbeitsgruppen mit und prägte deren Ergebnisse in erheblichem Maße.

Als zentrale Herausforderung für den Entwurf von DBMS identifizierten die Mitglieder der ANSI/X3/SPARC-Gruppe das Problem der Datenunabhängigkeit (vgl. Tsichritzis/Klug 1978: 183). Obwohl die genaue Bedeutung des Begriffs Mitte der 1970er Jahre noch immer zur Disposition stand, war man in der Arbeitsgruppe davon überzeugt, dass die vorgeschlagene Datenbankarchitektur einen wichtigen Beitrag zur Gewährleistung von mehr Datenunabhängigkeit leisten wird. Der 1975 veröffentlichte Zwischenbericht charakterisiert die Drei-Ebenen-Datenbankarchitektur deshalb als Eckpfeiler des vorgeschlagenen Datenbanksystemmodells: »The three-level approach to modelling [sic] a database described in the Interim Report is the most significant aspect of the proposed system Model [sic]« (Study Group on Data Base Management Systems 1975: 2).

Abb. 9: Vergleich der CODASYL- und ANSI/X3/SPARC-Datenbankarchitektur



Quelle: Bachman 1975: 570

Der Unterscheidung von Schema und Subschema wird in der ANSI/X3/SPARC-Architektur die dreistufige Unterscheidung von internen, konzeptuellen und externen Schema gegenübergestellt. Während das externe Schema, so Bachman, mit dem Subschema der DBTG gleichzusetzen ist, spezifizieren das interne und das konzeptuelle Schema die zweite Ebene der CODASYL-Architektur: »[T]he External Data Description Language […] can easily be related to the Sub Schema of the DBTG Report. The other two can be related to the schema of the DBTG Report« (Bachman 1974: 22). Demzufolge richtet sich der Vorschlag der ANSI/X3/SPARC-Arbeitsgruppe auf eine Ausdifferenzierung der Ebene des Schemas, bei der die

konzeptuelle Beschreibung der zu versammelnden Informationen von der Logik der Speicherung entkoppelt wird. Hierdurch wird die semantische oder logische Struktur des Informationsbestands von dessen Strukturierung im Speicher abgelöst. Beide fallen nicht mehr in eins, sondern werden als unterschiedliche Ebenen betrachtet, zwischen denen es zu übersetzen gilt.

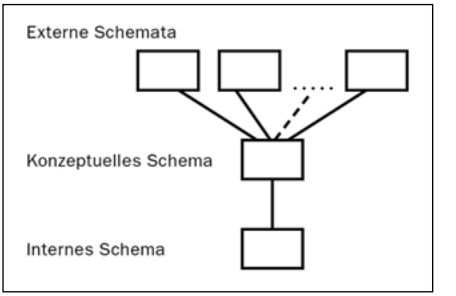
Im Rahmen der Drei-Ebenen-Datenbankarchitektur nimmt das konzeptuelle Schema eine Mittel- und Mittlerposition ein. Es situiert sich zwischen der internen Verarbeitungslogik des Computers sowie den externen Gebrauchslogiken der Nutzer und dient der Vermittlung zwischen diesen. Dabei stellt die Einführung des konzeptuellen Schemas im Vergleich zur Zwei-Ebenen-Architektur der CODASYL-DBTG streng genommen einen Umweg dar. An die Stelle der direkten Übersetzung zwischen der internen Repräsentation von Informationen und der externen Nutzersicht auf diese tritt ein zweistufiger Vermittlungsprozess. Auf diesen Umstand wird bemerkenswerter Weise sowohl im GUIDE-SHARE-Bericht als auch in den Publikationen der ANSI/X3/SPARC-Arbeitsgruppe hingewiesen. Beiden Gruppen erscheint dieser Umweg jedoch als notwendig, da er die Unabhängigkeit des Gebrauchs von Informationen von ihrer Ordnung im Speicher gewährleistet: »The placement of the conceptual schema between an external schema and the internal schema is necessary to provide the level of indirection essential to data independence« (Tsichritzis/Klug 1978: 184).[34] Deutlich wird dies in der folgenden Gegenüberstellung der beiden Architekturen (vgl. Abb. 9). Während die CODASYL-Datenbankarchitektur eine direkte Übersetzung zwischen der internen Speicherlogik und den externen Gebrauchslogiken von Information vorsieht, führt die ANSI/X3/SPARC-Datenbankarchitektur den »Umweg« eines konzeptuellen Schemas ein, das zwischen der Tiefe des Speichers und den Benutzeroberflächen von Datenbankinterfaces vermittelt.

In der CODASYL-Architektur wirkt sich jede Änderung am Schema einer Datenbank auf sämtliche Subschemata aus, da zumindest die »mappings« zwischen den beiden Ebenen neu definiert werden müssen und abhängig von den Änderungen am Schema gegebenenfalls auch die Deklarationen der Subschemata. Im Unterschied dazu können ausgehend vom konzeptionellen Schema unterschiedliche interne Schemata definiert werden, die nach außen hin für die Nutzer identisch sind. Auswirkungen haben Änderungen an der internen Speicherordnung infolgedessen allenfalls auf die Performance des Informationssystems, aber nicht auf dessen Funktionen. Darauf hat nur das konzeptuelle Schema einen Einfluss, von dem man jedoch annahm, dass es relativ beständig sei: »[I]t is anticipated that the Conceptual Schema will be very stable in nature« (Bachman 1974: 23).

34 | Im Bericht der Joint GUIDE-SHARE Data Base Requirements Group findet sich eine ähnliche Formulierung: »The level of indirection created by this use of the entity record type concept provides the environment of data independence« (Joint GUIDE-SHARE Database Requirements Group 1970: 3.6).

Durch die ANSI/X3/SPARC-Architektur wird der Informationsfluss zwischen Oberfläche und Tiefe modelliert. Es ist eine doppelte Bewegung von oben nach unten und von außen nach innen. An der Oberfläche operieren die Nutzer mit Informationen, die in der unsichtbaren Tiefe der Maschine gespeichert sind. Informationen fließen vertikal zwischen dem Speicher und den Benutzeranwendungen. Diese Vorstellung spiegelt sich in der gebräuchlichen Visualisierung der ANSI/X3/SPARC-Architektur wider, welche die horizontale Übersetzung zwischen dem internen, konzeptuellen und externen Schema(ta) in Bachmans Diagramm um 90 Grad dreht (vgl. Abb. 10). Oben sind die Nutzer und unten die Informationen, wie sie im Speicher hinterlegt sind.

Abb. 10: ANSI/X3/SPARC-Datenbankarchitektur



Der vertikale Informationsfluss zwischen dem Speicher und den Nutzern wird zudem als Übersetzung zwischen innen und außen gedacht. Das DBMS erscheint als eine geschachtelte Maschine, die im Kern eine Apparatur zur Verarbeitung von Signalen ist. Aus Sicht der Nutzer aber handelt es sich um eine Technologie, die es Ihnen erlaubt, mit Informationen und nicht mit Signalen umzugehen:

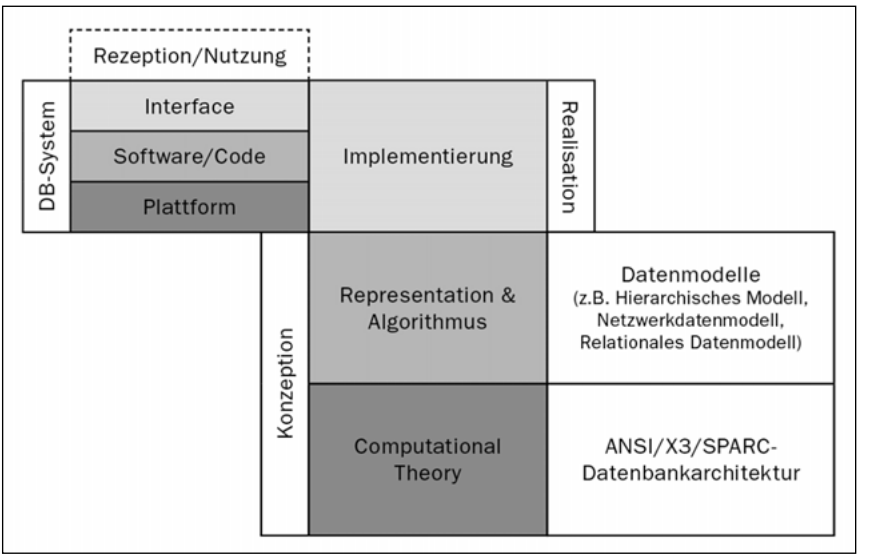
»The gross architecture is based in part upon the concept of nested machines. These machines began at the outside with the most complete support of database functionality and as each layer of machine is stripped away we find less logical capability and more physical capability until we reach the actual secondary storage device and its capability to read and write.« (Bachman 1974: 18)

Die elementaren Funktionen von Computern werden, so Bachman, stufenweise in komplexe logische Funktionen übersetzt, die zum nutzerseitigen Umgang mit Datenbanken notwendig sind. Erneut ist der Übergang nicht direkt, sondern basiert auf einem Schichten- oder Zwiebelmodell, bei dem die einzelnen funktionellen Ebenen autonom gegenüber den anderen sind.[35] Für die Implementierung komplexer

35 | Die Idee der Verschachtelung funktionaler Schichten ist Ende der 1960er Jahre im Bereich des Software Engineering aufgekommen. Auf einer 1968 von der NATO gesponserten Konferenz hat Edsger W. Dijkstra sein Modell von »Complexity con-

Anwendungen, wie z.B. die Suche nach sämtlichen Publikationen eines Autors, ist es weitgehend irrelevant, wie die hierfür benötigten elementaren Funktionen auf einer der darunterliegenden Ebenen technisch realisiert wurden.[36] Bedeutsam ist nur der Übergang zwischen den Schichten, der durch Schnittstellen gewährleistet wird, weshalb diese zu standardisieren seien: »[I]t emerged that what any standardization should treat is interfaces. There is potential disaster and little merit in developing standards that specify how components are to work« (Study Group on Data Base Management Systems 1975).

Abb. 11: Analyseebenen digitaler Datenbanktechnologien angelehnt an Marrs Modell von Informationssystemen



Die Drei-Ebenen-Architektur der ANSI/X3/SPARC Study Group kann im Anschluss an David Marr als Beschreibung der »computational theory« (Marr 1982: 27) digitaler Datenbanken im engen Sinn der Informatik betrachtet werden (vgl. Abb. 11). Sie beschreibt die elementaren Funktionen respektive Aufgaben von DBMS

trolled by hierarchical ordering of function and variability« (181) vorgestellt. Das Ziel der von Dijkstra beschriebenen Herangehensweise ist die Umwandlung von »a (for its user or for its manager) less attractive machine (or class of machines) into a more attractive one« (Dijkstra 1969: 181; siehe hierzu auch Rayley 2006).

36 | Beispiele hierfür sind die sogenannten höheren Programmiersprachen, wie z.B. C und Java, sowie *Frameworks*. Als *Framework* werden besondere Programmierumgebungen bezeichnet, die ein modulares Programmgerüst zur Verfügung stellen, welches die Grundlage oder den Rahmen für Programme darstellt (vgl. Gumm/Sommer 2009: 762).

sowie eine abstrakte Strategie zur Lösung des Informationsverwaltungsproblems, das sich im Begriff der Datenunabhängigkeit äußert. Nach Ansicht von Marr stellt die *Computational Theory* nur eine, wenngleich eine bedeutsame Dimension eines Informationssystems dar, die sich auf den Zweck und die Logik eines computertechnisch zu lösenden Problems bezieht. Davon unterscheidet er zwei weitere Ebenen: die der Repräsentation und der Algorithmen einerseits sowie die der physischen Implementierung des Systems andererseits.

Er weist darauf hin, dass zur Lösung jedes Informationsverarbeitungsproblems eine Reihe von Algorithmen und Datenrepräsentationen herangezogen werden können, die wiederum auf unterschiedliche Weise im tatsächlichen Informationssystem realisierbar sind. Demzufolge ist der Übergang von der abstrakten Ebene der funktionalen Spezifikation eines Systems hin zu einem konkreten Informationssystem doppelt kontingent. Darum ist es Marr zufolge unzureichend, nur eine der drei Ebenen zu betrachten. Wolle man ein Informationssystem verstehen, müsse man es auf den drei benannten Ebenen untersuchen, wobei die Betrachtung einer jeden Ebene Fragen aufwirft, die ziemlich unabhängig (*rather independent*) voneinander sind (vgl. Marr 1982: 25). Auch wenn sich Marr für eine integrierte Betrachtung stark macht, sind die drei Analyseebenen nicht gleich wichtig: »Although algorithms and mechanisms are empirically more accessible, it is the top level, the level of computational theory, which is critically important« (Marr 1982: 27). Daher sei es einfacher, die Funktionsweise eines Algorithmus zu verstehen, wenn man das zugrundeliegende Problem kennt, welches der Algorithmus löst, als wenn man eine konkrete Implementierung desselben betrachtet (vgl. Marr 1982: 27).[37]

Mit Marrs Modell von Informationssystemen lässt sich die Bedeutung und Funktion der ANSI/X3/SPARC-Architektur für die Entwicklung von Datenbanktechnologien gut nachvollziehen.[38] Denn die in der Architektur eingeschriebene Unterscheidung von drei Informationsebenen und die hierauf beruhende Modellierung des Informationsflusses in Datenbanken kann als Basis für die Implementierung von partikularen DBMS betrachtet werden. Sie modelliert die Funktion eines DBMS als zweifache Übersetzungsleitung zwischen dem internen Schema des Computers und den externen Schemata der Nutzer respektive Anwendungsprogramme auf Grundlage eines konzeptuellen Informationsmodells.

---

37 | Ein Indiz hierfür ist, dass Algorithmen in Publikationen zumeist nicht in Programmcode, sondern in Pseudocode dargestellt werden.

38 | Sofern das Modell von Marr den unidirektionalen Übergang von einem Problem zu einem Lösungsansatz und schließlich zu einer konkreten Lösung nahe legt, greift es zu kurz, die Entwicklung von digitalen Medientechnologien zu beschreiben. Wie eingangs des Kapitels dargelegt wurde, eröffnete die Festplatte als konkrete Speichertechnologie den Problemhorizont, vor dem zentrale Datenbankkonzepte entwickelt wurden. Hierbei hat eine konkrete Technologie auf die Probleme zurückgewirkt, die technisch gelöst werden mussten.

Beschrieben wird jedoch nicht, wie dies tatsächlich mit Computern zu realisieren ist. An dieser Stelle setzen die diversen Datenmodelle an, welche in konkreten DBMS Verwendung finden. Durch die Drei-Ebenen-Datenbankarchitektur werden vielmehr die materiellen Bedingungen beschrieben, unter denen digitale Informationen ein gewisses Maß an Unabhängigkeit gegenüber den Programmen gewinnen, in denen sie verarbeitet werden. Hieraus resultiert der Eindruck der Immaterialität digitaler Informationen, der weder bloßer Schein, noch ein Wesensmerkmal dieser Informationen ist, sondern ein Leistungsmerkmal von Informationssystemen, die auf der *Computational Theory* der ANSI/X3/SPARC-Architektur beruhen.[39]

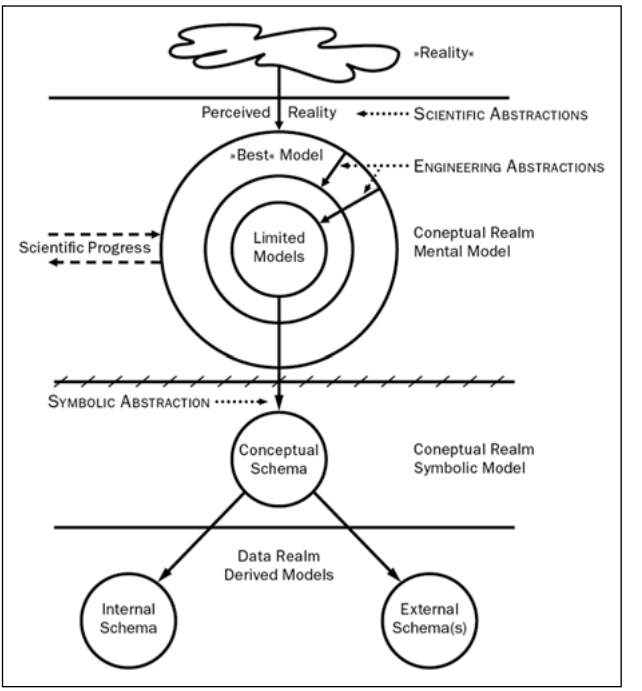
Beschreibt die ANSI/X3/SPARC-Architektur das *Computational Model* von Datenbanken im engen Sinn der Informatik, so ist dieses als ein kontingentes Modell der Versammlung und Verwaltung digitaler Informationen zu verstehen, welches gegenüber anderen Formen des Umgangs mit Sammlungen in digitalen Medien abgegrenzt werden muss. Dies ist möglich, wenn man danach fragt, wie die in der Architektur unterschiedenen drei Ebenen digitaler Informationen in anderen Gebrauchskontexten praktisch miteinander verschaltet werden. Hierdurch werden die mit DBMS verwalteten Informationssammlungen vergleichbar mit anderen Informationssammlungen, die aus Sicht der Informatik allenfalls Datenbanken im metaphorischen Sinn sind. Die von der ANSI/X3/SPARC Study Group on DBMS vorgeschlagene Architektur lässt sich daher als eine bestimmte Form der Verknüpfung der drei Ebenen digitaler Information begreifen, welche den Informationsfluss zwischen Oberfläche und Tiefe auf eine spezifische Weise konfiguriert. Kennzeichnend ist für diese unter anderem, dass die Ebene exakt bestimmt worden ist, auf der die strukturelle Beschreibung derjenigen Informationen ansetzt, die in Datenbanken gespeichert werden sollen.

Während das interne Schema die Ordnung von Informationen im Speicher modelliert und die externen Schemata unterschiedliche Sichten auf die Datenbank beschreiben, definiert das konzeptuelle Schema, welche Informationen potenziell in einer konkreten Datenbank verwaltet werden können. Das konzeptuelle Schema modelliert *eine* Realität, auf die sich die Datenbank bezieht und über die sie informieren wird. Es verweist zugleich auf ein Außen, auf *die* Welt, in der die Datenbank entworfen, entwickelt und betrieben wird und über die sie Auskunft gibt. Das Außen des konzeptuellen Schemas ist demzufolge ein anderes als das der externen Schemata, die die äußere Sicht der Nutzer auf den Informationsbestand im Inneren der Datenbank definieren. Das konzeptuelle Schema verweist auf eine Wirklichkeit,

39 | Die ANSI/X3/SPARC-Architektur stellt nur *eine* mögliche *Computational Theory* für digitale Datenbanken dar, in deren Zentrum die Forderung nach Datenunabhängigkeit steht. Dies zeigt sich in der Debatte um NoSQL-Datenbanken, die nicht nur explizit mit der SQL-Abfragesprache und dem ihr zugrunde liegenden relationalen Datenmodell brechen, sondern implizit auch mit der *Computational Theory* der ANSI/X3/SPARC-Architektur, S. 232f.

deren Existenz zumindest als regulative Idee angenommen wird, wie die Mitglieder der ANSI/X3/SPARC Study Group am Ende ihres Zwischenberichts konstatieren (vgl. Study Group on Data Base Management Systems 1975: VIII-30) und in einem Diagramm veranschaulichen (vgl. Abb. 12).

Abb. 12: Not just Reality – Data



Quelle: Study Group on Data Base Management Systems 1975: VIII-32

Die Deklaration eines Modells der Wirklichkeit im konzeptuellen Schema wird als das Ergebnis einer dreifachen Abstraktion begriffen. Im ersten Abstraktionsschritt wird in einem wissenschaftlichen Erkenntnisprozess aus der wahrgenommenen Wirklichkeit ein Modell entwickelt. Von dem wissenschaftlich »besten« Modell wird im zweiten Abstraktionsschritt derjenige Ausschnitt bestimmt, der für das zu entwickelnde Datenbanksystem relevant ist. Schließlich wird dieses mentale Modell in einem symbolischen Abstraktionsprozess formalisiert und so als konzeptuelles Schema expliziert.[40] An diesem Metamodell kommen zwei zentrale Aspekte oder Motive des Entwurfs konzeptueller Schemata zum Vorschein, die für

40 | Dieses Modell der Modellierung konzeptueller Schemata basiert ohne Zweifel auf problematischen philosophischen und medientheoretischen Grundannahmen, die von den Autoren weder genau expliziert noch diskutiert werden. Daher ist anzunehmen, dass die Mitglieder der Arbeitsgruppe diesen Aspekten, Fragen und Pro-

eine weiterführende medientheoretische Auseinandersetzung mit Datenbanken grundlegend sind: die ökonomisch motivierte Reduktion des Schemas auf einen relevanten Weltausschnitt sowie der Aspekt der Formalisierung.

In Datenbanken ist ein beschränktes Modell einer Wirklichkeit eingeschrieben, welches im konzeptuellen Schema formalisiert wird. Derjenige Ausschnitt der Wirklichkeit, für den sich die Autoren der ANSI/X3/SPARC-Empfehlungen hauptsächlich interessieren, sind wirtschaftliche Unternehmen, weshalb ihre Erörterungen stets um diesen Kontext kreisen. Dementsprechend wird das konzeptuelle Schema beschrieben als »the enterprise's view of the structure it is attempting to model in the data base« (Study Group on Data Base Management Systems 1975: I-5). Bei der Explikation dieses Schemas geht es, wie Bachman dargelegt hat, um die Beschreibung der »basic nature of the enterprise« (Bachman 1974: 23). Gemessen an der Komplexität der »Realität« ist dieses Modell stets unvollständig, d.h. das konzeptuelle Schema reduziert die Komplexität der Wirklichkeit auf ein bestimmtes Abstraktionsniveau, welches die äußeren Grenzen der Datenbank markiert:

»While it may be the case that the universe is ›best‹ described by the interactions of 3.1081 quarks, the typical engineer is more apt to build his bridge by combining girders, cross braces and rivets. The molecular biologist may view the human being as a complex structure of water, protein molecules, DNA and other assorted chemicals, but to the insurance agent a human being is not much more than an age, sex, and checkbook. For any application one abstracts those aspects of ›reality‹ considered relevant and ignores the rest.« (Study Group on Data Base Management Systems 1975: VIII-30f.)

Was im Entwurf des konzeptuellen Schemas keinen Platz findet, hat in der Welt der Datenbank keine Existenz.[41] Die Grenzen des konzeptuellen Schemas sind die Grenzen der Datenbank. Dieser notwendigen Beschränktheit von Datenbanksystemen stellen die Mitglieder der Study Group das fragwürdige Versprechen entgegen, dass im konzeptuellen Schema zwar nicht alle, aber doch zumindest alle *relevanten* Aspekte der Wirklichkeit modelliert werden. Hierbei ersetzen die Autoren letztlich eine Vollständigkeitsutopie (die Datenbank enthält alle Informationen) durch eine andere (die Datenbank enthält alle wichtigen Informationen). Darüber, welche Aspekte von Bedeutung sind und welche nicht, lässt sich jedoch trefflich streiten. Daher sind die konzeptuellen Modelle von Datenbanken nie bloßes Abbild der Wirklichkeit, sondern eine Konstruktion dieser.

---

blemen keine zentrale Bedeutung beigemessen haben, weshalb von dem Versuch abgesehen wird, das Modell einer detaillierten Kritik zu unterziehen.

41 | Die Unvollständigkeit von konzeptuellen Schemata wird auch in Bezug auf den ökonomischen Gebrauchskontext erwähnt: »As the conceptual schema is a formal model of the enterprise, when the situation is at all complex, the model may be logically incomplete« (Study Group on Data Base Management Systems 1975: I-6).

An dieser Stelle lohnt sich ein Rückblick auf die These von Lev Manovich, die Datenbank fungiere als symbolische Form der digitalen Medienkultur. Manovich versteht die Datenbank in erster Linie als Gegenmodell *von* und Voraussetzung *für* Erzählungen im Zeitalter digitaler Medientechnologien. Seine Analyse beschränkt sich, wie im Kapitel »Datenbank« bereits kritisch angemerkt wurde, jedoch weitgehend auf die Oberfläche verschiedener Benutzerinterfaces. Demgegenüber erlaubt es die Betrachtung der ANSI/X3/SPARC-Datenbankarchitektur, den Prozess der symbolischen Formung in den Blick zu nehmen, der sich *in*, *mit* und *durch* digitale Datenbanken im engeren Sinn von DBMS vollzieht.

Die Drei-Ebenen-Datenbankarchitektur als Metamodell der Informationsmodellierung in Datenbanken im Allgemeinen und das konzeptuelle Schema als Medium (im doppelten Sinn von Mitte und Mittler) zwischen der internen Verarbeitungslogik des Computers und den externen Gebrauchslogiken der Nutzer im Besonderen strukturieren die Versammlung, Verwaltung, Abfrage und Auswertung von Informationen in und mit Computerdatenbanken. Es handelt sich dabei gleichermaßen um eine Weise der Herstellung »computer-lesbare[r] Signifikanz« (Becker/Stalder 2009b: 8) wie um eine der Wirklichkeitserschließung, da das konzeptuelle Schema festlegt, was im Rahmen einer Datenbank als Information zählt und somit spezifiziert, welcher Ausschnitt der Wirklichkeit wie beschrieben werden kann. In dieser Hinsicht kann das konzeptuelle Schema als eine jener »Formen des Weltbegreifens und Weltverstehens« (Cassirer 1956 [1938]: 209) betrachtet werden, die Ernst Cassirer in seiner Kulturphilosophie mit dem Begriff der symbolischen Form konzeptualisiert und untersucht hat.

Cassirers Hinwendung zu symbolischen Formen kann als kulturwissenschaftliche Umdeutung von Kants kritischer Philosophie betrachtet werden, wie er in der Einleitung zum ersten Band der *Philosophie der symbolischen Formen* herausstellt (vgl. Cassirer 2001 [1923]: 8ff.). Anders als Kant fragt Cassirer jedoch nicht nach den transzendentalen Bedingungen der Möglichkeit von Erkenntnis überhaupt, sondern versucht, unterschiedliche historisch wandelbare und kulturell bedingte Erkenntnisformen freizulegen, wie z.B. Sprache, Mythos, Wissenschaft und Kunst. Cassirers kulturphilosophisches Projekt zielt auf eine »Phänomenologie der Erkenntnis« (Cassirer 1956 [1938]: 208), wobei er den Erkenntnisbegriff explizit nicht auf den Bereich des naturwissenschaftlich-mathematischen Wissens begrenzt wissen will. Erkenntnis konstituiert für Cassirer vielmehr »jede geistige Tätigkeit, in der wir uns eine ›Welt‹ in ihrer charakteristischen Gestaltung, in ihrer Ordnung und in ihrem ›So-Sein‹, aufbauen« (Cassirer 1956 [1938]: 208). Entscheidend für das Fragen nach und die Untersuchung von »verschiedene[n] ›Dimensionen‹ des Erkennens, des Verstehens, des Denkens« (Cassirer 1956 [1938]: 208) ist die von Kant übernommene Überzeugung, dass Erkenntnis kein bloßes Empfangen von Gegebenem ist, sondern ein aktives Geschehen, dem Cassirer zufolge »eine ursprünglich-bildende, nicht bloß eine nachbildende Kraft innewohnt« (Cassirer 2001

[1923]: 8).[42] Diese *ursprünglich-bildende* Kraft des Erkennens artikuliert sich auf unterschiedliche Weise in verschiedenen symbolischen Formen, welche Cassirer wie folgt definiert:

»Unter einer ›symbolischen Form‹ soll jede Energie des Geistes verstanden werden, durch welche ein geistiger Bedeutungsgehalt an ein konkretes sinnliches Zeichen geknüpft und diesem Zeichen innerlich zugeeignet wird. In diesem Sinne tritt uns die Sprache, tritt uns die mythisch-religiöse Welt und die Kunst als je eine besondere symbolische Form entgegen. Denn in ihnen allen prägt sich das Grundphänomen aus, daß unser Bewußtsein sich nicht damit begnügt, den Eindruck des Äußeren zu empfangen, sondern daß es jeden Eindruck mit einer freien Tätigkeit des Ausdrucks verknüpft und durchdringt. Eine Welt selbstgeschaffener Zeichen und Bilder tritt dem, was wir die objektive Wirklichkeit der Dinge nennen, gegenüber und behauptet sich gegen sie in selbständiger Fülle und ursprünglicher Kraft.« (Cassirer 1923: 15)

Sprachliche, mythische, künstlerische ebenso wie wissenschaftliche Erkenntnis stellen für Cassirer somit verschiedene Weisen der symbolischen Formung des Gegebenen dar, welche der Autor in *Der Gegenstand der Kulturwissenschaft* auch als »vorlogische Strukturierung« (Cassirer 2008 [1942]: 174) bezeichnet. Diese vor- oder wenn man so will: kulturlogische Strukturierung artikuliert sich in unterschiedlichen »Weisen der Zuordnung« (Cassirer 2008 [1942]: 174) des Gegebenen. Der Geist, dem Cassirer zufolge die bildende Kraft symbolischer Formen innewohnt, erfindet demzufolge keine bloßen Phantasiewelten. Die »Erschaffung bestimmter geistiger Bildwelten« (Cassirer 2001 [1923]: 24) ist vielmehr abhängig von der Welt, die sich durch verschiedene symbolische Formen jedoch auf je unterschiedliche Weise als Welt entfaltet (vgl. Cassirer 1956 [1938]: 208f.).[43] Dabei müssen die Weisen der Zuordnung Cassirer zufolge »überall an anschauliche Gliederungen anknüpfen« (Cassirer 2008 [1942]: 173). Ähnlich wie Cassirer in Bezug auf Technik feststellt, ver-

42 | In der Kritik der reinen Vernunft postuliert Kant, dass es sich von der Vorstellung zu verabschieden gelte, »alle unsere Erkenntnis müsse sich nach den Gegenständen richten« (Kant 2000: B-XVI). Vielmehr sei anzunehmen, dass »sich aber der Gegenstand (als Objekt der Sinne) nach der Beschaffenheit unseres Anschauungsvermögens« (Kant 2000: B-XVIII) bildet. Daher ist Kant zufolge danach zu fragen, wie sich das menschliche Anschauungs- und Erkenntnisvermögen in das Erkennen a priori einschreibt und unser Wissen von der Welt bedingt. Bei Cassirer erfährt Kants Frage nach den transzendentalen Bedingungen der Möglichkeit von Erkenntnis eine kulturphilosophische Wendung; siehe hierzu auch Wirth (2008b: 14ff.).

43 | Nach Ansicht von Cassirer ist es unmöglich, die Welt losgelöst von symbolischen Formen zu erkennen. Daher besteht die Aufgabe der Kulturphilosophie darin, diese Bildwelten »in ihrem gestaltenden Grundprinzip zu verstehen und bewußt zu machen« (Cassirer 2001 [1923]: 50f.).

dunkeln symbolische Formen das »Sein nicht, sondern mach[en] es von einer neuen Seite her sichtbar« (Cassirer 2009 [1930]: 44).

Sofern es sich bei symbolischen Formen im Sinne Cassirers um Weisen der Sichtbarmachung bzw. Erschließung der Welt handelt, kann das konzeptuelle Schema der ANSI/X3/SPARC-Datenbankarchitektur als Weise der symbolischen Formung *in*, *mit* und *durch* Computer verstanden werden. Denn das konzeptuelle Datenbankschema stellt keine nachträgliche Interpretation von Informationen dar, sondern gibt eine Struktur vor, die bestimmt, wie Informationen in Datenbanken versammelt werden können und wie auf sie zugegriffen werden kann. Kurzum, das Schema bedingt, was DBMS verwalten und worüber sie informieren können. Es dient als Vermittler zwischen der Speicherlogik des Computers und unterschiedlicher Gebrauchslogiken von Information, indem es ermöglicht, die digitalen Daten im Speicher (Binärketten) als bestimmte Information (beispielsweise als Name, Freund, Autor, ISBN-Nummer, Nachricht, Blogeintrag, Artikel, Kommentar, Bewertung, Schlagwort etc.) zu adressieren. Infolgedessen dürfen Datenbanken nicht als passive Container verstanden werden, die als bloßer Speicher bereits existierender Informationen dienen. Vielmehr bringen sie die Informationen, die in ihnen verwaltet werden, mit hervor. Einerseits geben Datenbankinformationen zwar Auskunft über etwas außerhalb der Datenbank Liegendes (Repräsentation), anderseits legt das konzeptuelle Informationsmodell der Datenbank fest, was überhaupt als Information zählt, d.h. durch welche Informationen die Wirklichkeit beschrieben wird (Konstruktion). Insofern macht das konzeptuelle Schema die Welt »von einer neuen Seite her sichtbar« (Cassirer 2009 [1930]: 44). Dass es sich hierbei stets um einen beschränkten Wirklichkeitsausschnitt handelt, dessen informationelle Beschreibung zumeist ökonomisch motiviert ist, wurde bereits erwähnt. Entscheidend ist an dieser Stelle, dass die im konzeptuellen Schema formalisierte *Weise der Zuordnung* nicht sekundär ist, sondern vorschreibt, was in einer Datenbank als Information zählt und in dieser als Information über Realität versammelt werden kann. Besonders offenkundig wird dies bei genuin digitalen Medienobjekten bzw. Inhalten, wie z.B. den Nutzerprofilen in sozialen Netzwerken, deren Struktur bestimmt, wie sich Personen präsentieren, wie sie mit anderen Nutzern interagieren und welche Beziehungen zwischen ihnen bestehen können. Weniger offensichtlich ist dies bei der Versammlung von Informationen über realweltliche Objekte, die scheinbar eindeutig bestimmt sind, wie dies z.B. bei der Katalogisierung von Büchern der Fall ist. Doch auch Bücher lassen sich nicht nur auf eine Weise beschreiben, woran zugleich deutlich wird, dass es sich hierbei um kein Spezifikum digitaler Datenbanken handelt, sondern auf Informationsinfrastrukturen im Allgemeinen zutrifft, die z.B. auch in Kategorisierungen oder Standards realisiert werden.[44]

44 | In *Sorting Things Out* haben Geoffrey Bowker und Susan Leigh Star dies am Beispiel von Standards und Klassifikationssystemen wie dem *International Statistical*

Ein Buch, welches in einer bibliographischen Datenbank verzeichnet wird, existiert auch unabhängig von seiner Katalogisierung. Was durch den Datenbankkatalog hervorgebracht wird ist deshalb nicht das Buch als Text, sondern Informationen über ein Buch als Entität, in Gestalt eines bibliographischen Eintrags. Historisch haben sich die Aspekte (Autor, Titel, Verlag, Erscheinungsort, Erscheinungsjahr, ...), durch die Bücher beschrieben werden, weithin stabilisiert. Jedoch stehen die Modi der bibliographischen Beschreibung nicht fest. So kann »dasselbe Buch« beispielsweise im Kontext der bibliothekarischen und archivarischen Erschließung unterschiedliche Gestalt annehmen. Wird es vom Bibliothekar als eigenständige Entität katalogisiert, betrachtet es der Archivar als Teil einer Sammlung oder eines Bestands und erschließt es als solches.[45] Hieran werden unterschiedliche Politiken und Praktiken des »als Eins zählen[s]« (Badiou 2005: 37) deutlich, welche in unterschiedlichen konzeptuellen Schemata formalisiert und technisch operativ werden können. Durch die Erfassung eines Dokuments in einem Katalog wird dieses als Entität eines bestimmten Typs festgelegt. Dem Buch wird etwas hinzugefügt, was diesem nicht immanent ist. Daher ist es im Kontext einer Bibliothek etwas anderes als im Kontext eines Archivs, was sich in den Informationen zeigt, die über »dasselbe Buch« in unterschiedlichen Katalogdatenbanken gespeichert werden.[46] Die Formulierung »dasselbe Buch« ist zu apostrophieren, da die unterschiedlichen Formen der informationellen Beschreibung von Entitäten festschreiben, was als dasselbe gilt und was nicht. Unterschiedliche Exemplare der gleichen Ausgabe von Goethes Faust *können* in einer Bibliothek als dasselbe Buch gelten. Archive hingegen referenzieren die Provenienz der Dokumente, weshalb »dasselbe Buch« in unterschiedlichen Sammlungsbeständen als unterschiedlich betrachtet wird.[47] Editionsphilologen können wiederum andere

Classification of Diseases and Related Health Problems (ICD) aufgezeigt (vgl. Bowker/Star 2000: 33ff.).

45 | Ein Beispiel hierfür ist die Sammlung *Gordon Everest Monographs on Database Development* im Archiv des Charles Babbage Institute for the History of Information Technology der University of Minnesota, Minneapolis. Dieser Bestand beinhaltet die von Gordon Everest zwischen 1957 und 2003 gesammelten Monographien über Datenbanktechnologien. Im Archiv sind diese Bücher als Teil einer Sammlung erschlossen; vgl. Charles Babbage Institute/Horowitz (2007).

46 | Auch der Prozess der bibliothekarischen Erschließung von Dokumenten ist kontingent. Dies zeigt sich beispielsweise an der Katalogisierung von Sammelbänden, die häufig nicht mehr nur als Bücher verzeichnet werden. In digitalen Bibliothekskatalogen werden immer häufiger auch die einzelnen Aufsätze in Sammelbänden verzeichnet und können vom Nutzer gesucht werden. Damit ändert sich die dokumentarische Bezugseinheit bibliothekarischer Katalogisierung vom materiellen Dokument zum immateriellen Text.

47 | Sofern unterschiedliche Medienprodukte in verschiedenen Kontexten als dasselbe gelten, muss die im Kapitel »Medium« (S. 47f.) diskutierte These Wiesings,

Kriterien für die Selbigkeit eines Buchs anlegen, da für sie gerade zur Debatte steht, welche Bücher, Ausgaben oder Textvarianten als dieselben zählen sollen (vgl. Martens 1991: 22f.)

An dem genannten Beispiel wird deutlich, dass die Informationsmodelle, auf denen Datenbanken beruhen, kein Abbild der objektiven Realität sind. Sie fügen dem, worüber sie informieren, etwas hinzu, indem sie festschreiben, was im Rahmen der Datenbank als Entität (und damit als seiend) gilt und welche Attribute diese Entität auszeichnen. Einer an den platonischen Idealismus anschließenden Lesart von Datenbanken als defizitärem Abbild der Wirklichkeit setzt Floridi daher eine ontologische Interpretation entgegen (vgl. Floridi 1999: 110). Ihr zufolge sind Datenbanken kein sekundäres Abbild, sondern die Basis für Wirklichkeit: »[T]he infosphere is the authentic reality that underlies the physical world« (Floridi 1999: 110). Jedoch ist auch diese Sichtweise problematisch, da der Verweischarakter von Datenbankinformationen, d.h. der Weltbezug von Datenbanken, in der medialen Praxis nicht aufgegeben wird. Wären Datenbanken bloße Konstruktionen, dann müssten die versammelten Informationen nicht empirisch erfasst werden, sondern könnten nach bestimmten Regeln berechnet werden. Doch Datenbanken sind nie bloßes Kalkül, d.h. das Weltmodell, welches dem Datenbanksystem als konzeptuelles Schema zugrunde gelegt wird, mag zwar konstruiert sein, aber die Daten, die diesem Modell folgend in die Datenbank eingespeist werden, lassen sich nicht berechnen; sie sind Informationen über Realität, deren Richtigkeit sich nicht mathematisch belegen, sondern nur empirisch überprüfen lässt.[48] Datenbanken konstruieren ein Modell der Wirklichkeit und liefern zugleich ein Bild der Wirklichkeit. Diesen Doppelcharakter gilt es ernst zu nehmen, wie in Rekurs auf Cassirers Philosophie der symbolischen Formen gezeigt werden konnte. Die Betrachtung des konzeptuellen Schemas als einer Form der *vorlogischen Strukturierung* von Wirklichkeit geht jedoch auch über Cassirers »Kritik der Kultur« (Cassirer 2001 [1923]: 11) hinaus. Denn die von ihm vorgeschlagene kritische Philosophie der Kultur ist stark am Geist orientiert, als jener Energie oder Kraft »durch die das schlichte Dasein der Erscheinung eine bestimmte ›Bedeutung‹, einen eigentümlichen ideellen Gehalt empfängt« (Cassirer 2001 [1923]: 9). Im Kontext einer Analyse, die es sich zur Aufgabe macht, den Eigensinn freizulegen, den je spezifische mediale Konfigurationen entfalten, gilt es die Kritik der Kultur hin zu einer Kritik der Kul-

---

dass mediale Konstellationen über artifizielle Selbigkeit verfügen, modifiziert werden. Artifizielle Selbigkeit ist keine stabile, überzeitliche Eigenschaft medialer Konstellationen, sondern wird in mediale Praxen mitkonstituiert.

48 | Gegen Floridis ontologische Interpretation von Datenbanken lässt sich die Argumentation von Winkler ins Feld führen, der sich dafür stark macht, dass Formalsprachen anders als häufig behauptet nicht von »*Weltbezug und Referenz*« (Winkler im Druck) entkoppelt sind. Die zunehmende Formalisierung abstrahiert nicht von der Wirklichkeit, sondern gibt ihr eine rigide Form, die Voraussetzung für die automatisierte Verarbeitung von Informationen über Realität ist.

turtechniken zu erweitern. Eine solche Kritik fragt nicht ausschließlich nach den geistigen Funktionen, die es ermöglichen »die passive Welt der bloßen Eindrücke [...] zu einer Welt des reinen geistigen Ausdrucks umzubilden« (Cassirer 2001 [1923]: 12). Sofern das »Problem der *Wirklichkeitserschließung*« (Kreis 2010: 173) im Zentrum der Philosophie der symbolischen Formen steht, wie Guido Kreis in seiner Studie zu Cassirers Kulturphilosophie nahelegt, sind spätestens unter der Bedingung digitaler Medientechnologien auch nicht-geistige, technische Formen der Wirklichkeitserschließung zu berücksichtigen. Als Beispiel hierfür diente die Betrachtung der Herstellung computer-lesbarer Signifikanz im Rahmen der ANSI/X3/SPARC-Datenbankarchitektur, auf der das zweifache Potenzial von Datenbanken beruht: Ressource bekannter Informationen über Realität zu sein und Basis für neue Informationen.[49] Die Möglichkeit hierzu wird durch die Formalisierung des konzeptuellen Schemas geschaffen, aber auch beschränkt.

Während das Modell die Datenbank nach außen hin begrenzt, bestimmt der Formalismus, wie die Daten im Computer operativ werden können, d.h. durch die Übersetzung des konzeptuellen Weltmodells in ein formales Modell, das konzeptuelle Schema, werden die Daten für den Computer als Informationen verarbeitbar. Den binär-digital codierten Signalfolgen (Information als Realität) wird eine Form gegeben, die sie auf *eine* Bedeutung festlegt und sie als Informationen (über Realität) adressierbar macht. In DBMS wird dieses formale Modell im sogenannten *Data Dictionary* gespeichert, das als »meta data data base« (Study Group on Data Base Management Systems 1975: II-32) dient. Ähnlich wie Wörterbücher Beschreibungen sprachlicher Einheiten beinhalten, aber keineswegs sämtliche Aussagen, die mit diesen getätigt werden können, enthält das Datenwörterbuch keine Informationen darüber, welche Informationen tatsächlich in einer Datenbank gespeichert sind – dies würde ein Katalog leisten –, sondern nur darüber, welcher Art Informationen potenziell in dieser gespeichert und aus dieser abgefragt werden können.[50]

Im Folgenden soll betrachtet werden, wie Bedeutung durch konkrete Datenmodelle in DBMS expliziert wird und wie sich dies auf die computertechnische Verwaltung und Verarbeitung digitaler Informationen auswirkt. Davon ausgehend

49 | Um den epistemologischen Unterschied zwischen der ästhetischen und der ontologischen Interpretation zu erläutern, zieht Floridi einen Vergleich zwischen Biologie und Physik einerseits und Mathematik und Informatik andererseits heran. Während man in Biologie und Physik von einer präexistenten Welt ausgeht, die durch theoretische Modelle beschrieben werden soll, um Vorhersagen über die Welt treffen zu können, dienen Modelle in der Mathematik und Informatik als Ausgangspunkt oder Basis, an der die Welt schließlich bemessen wird (vgl. Floridi 1999: 111).

50 | Neben dem konzeptuellen Schema enthält das *Data Dictionary* auch Informationen über das interne Schema sowie die externen Sichten auf die Datenbank. Es ist das »repository of information about the database« (Tsichritzis/Klug 1978: 185).

werden alternative Modi der computertechnischen Zuschreibung und Verarbeitung von Bedeutung betrachtet, die über Datenbanken im engen Sinn der Informatik hinausweisen.

Data + Access: Datenmodelle und Algorithmen

Datenbankinformationen entsprechend eines konzeptuellen Schemas zu strukturieren schafft computer-lesbare Signifikanz, indem Informationen *über* Realität in Informationen *als* Realität übersetzt werden. Eine (nicht *die*) Bedeutung medialer Konstellationen wird formal expliziert. Im undifferenzierten Strom digitaler Daten werden so syntaktische Unterschiede eingeführt, die einen semantischen Unterschied machen. Zeichenketten, wie z.B. »Red, Rose; Color Inc., Blueroad 31, 32442 Greentown; 1954-01-15«, sind für Computer zunächst nur Aneinanderreihungen von Zeichen. In Kombination mit den im konzeptuellen Schema enthaltenen Metainformationen wird diese Zeichenfolge, die sich gleichermaßen auf eine Person (Name, Kontaktadresse, Geburtsdatum) wie auf einen Farbton (Farbbezeichnung, Bezugsadresse, Bestellnummer) beziehen kann, für Computer als eine bestimmte Information interpretierbar. Computer, die Information ausschließlich auf dem Niveau von Information *als* Realität prozessieren, können digitale Daten durch eine solche Strukturierung als Information *über* Realität verarbeiten und automatisch in Datenbanken verwalten. Die Explikation von Bedeutung beruht auf der Segmentierung von Daten in bedeutungstragende Einheiten, die als Informationspartikel bezeichnet werden können, und der doppelten Zuschreibung von Gehalt und Referenz zu dieser Information.[51] Demgemäß kann »1954-01-15« als Attribut des Typs *Geburtsdatum* identifiziert werden, welches der *Person* »Rose Red« gehört. Eine Kette von Zeichen wird hierbei als Informationspartikel betrachtet, dem eine bestimmte Bedeutung zukommt (Gehalt) und der sich auf eine bestimmte Entität bezieht (Referenz).[52] Als Entitäten können nicht nur Dinge oder Personen beschrieben werden, sondern auch Ereignisse, Handlungen und Transaktionen, wie z.B. das Abheben von Geld an einem Bankautomaten, das Tätigen eines Anrufs mit

51 | Dietrich gebraucht die Metapher des Datenpartikels, um die Verkörperung von Information im binär-digitalen Code zu bezeichnen. Ein Datenpartikel sei »the smallest undividable physical unit capable of carrying digital information« (Dietrich 1986: 137). Im Unterschied hierzu lassen sich Informationspartikel als spezifische Konstellationen von Datenpartikeln verstehen, die eine bestimmte Information zum Ausdruck bringen.

52 | Dieses Verständnis von Informationsartikeln hat Bachman in einem Interview mit Haigh herausgestellt: »We believed that the most elementary particle of information had three properties. Essentially they were 1) a data value, 2) a name that characterized what that data value meant, and 3) an identification of who/what it belonged to« (Bachman 2006: 24).

dem Telefon oder der Aufruf einer bestimmten Webseite. Im Kontext eines Datenbanksystems sind die Entitäten vollständig durch ihre Attribute bestimmt, denen bei der Dateneingabe Werte zugewiesen werden. Insofern definiert das konzeptuelle Schema eine abstrakte Struktur, die im Umgang mit der Datenbank erst mit konkreten Informationen gefüllt werden muss. Sie bleibt den Daten gewissermaßen äußerlich, weist ihnen aber zugleich *eine* Bedeutung zu. Diese Zuschreibung legt die gespeicherten Werte auf etwas – ein Attribut einer Entität – fest und macht sie hierdurch als Werte eines bestimmten Typs adressierbar, die ausgewertet und mit anderen Werten kombiniert werden können, sodass man im Umgang mit Datenbanken nicht nur bekannte, sondern genuin neue Informationen erhalten kann.

Im Rahmen der ANSI/X3/SPARC-Datenbankarchitektur bildet das konzeptuelle Schema den Dreh- und Angelpunkt von DBMS. Die Explikation des Informationsmodells ist nicht Selbstzweck, sondern dient der automatisierten Verwaltung von Informationen durch Computer. Nach Bachman kommt es auf vier grundlegende Funktionen im Umgang mit Informationen an: »If records are put into a file so they can be used, four basic functions are involved. We must be able to store, retrieve, modify and delete« (Bachman 1966: 225). Datenbankverwaltungssysteme sollen demzufolge das Speichern, Abfragen, Verändern und Löschen von Informationen in Computerdatenbanken ermöglichen. Wie genau diese Operationen mit Computern automatisiert werden können und auf welche Weise die zu verwaltenden Informationen im konzeptuellen Schema zu modellieren sind, darüber geben die Berichte der ANSI/X3/SPARC keine Auskunft. Vielmehr hofften die Mitglieder der Arbeitsgruppe, dass künftige DBMS es ihren Nutzern freistellen werden, welches Modellierungsverfahren in konkreten Anwendungen zum Einsatz kommen soll:

»[T]here is a continuing argument on the appropriate data model: e.g., relational, hierarchical, network. If, indeed this debate is as it seems, then it follows that the correct answer to this question of which data model to use is necessarily ›all of the above‹. A major consequence of the model described in this Report is a mechanism that permits this answer in a meaningful sense.« (Study Group on Data Base Management Systems 1975: I-1)

Die Autoren hegten die Hoffnung, dass sich eine standarisierte Sprache zur formalen Deklaration des konzeptuellen Schemas etablieren werde, welche die Differenzen zwischen den Anfang der 1970er Jahre zur Disposition stehenden Datenmodellen überwinden würde. Dieser Wunsch ist nie technische Realität geworden, da sich die Weise der Modellierung von Information stets auf deren computertechnische Verarbeitung auswirkt.[53] Das hierarchische Datenmodell, das Netzwerk-Daten-

53 | Im Interview mit Haigh bestätigt Bachman, dass man bei dem Entwurf der Drei-Ebenen-Architektur die Hoffnung hegte, DBMS würden künftig nicht nur mit spezifischen Datenmodellen operieren, sondern die abstrakte Beschreibung eines kon-

modell und das relationale Datenmodell übersetzen die in der ANSI/X3/SPARC-Architektur formulierte *Computational Theory* digitaler Datenbanken in logische Repräsentationsformen und Algorithmen. Im Rahmen dieser Datenmodelle nimmt das abstrakte Ziel der formalen Explikation eines Informationsmodells gegenüber dem Datenbanksystem konkrete Gestalt an. Sie ermöglichen auf eine ihnen je eigne Weise die Modellierung von Information im konzeptuellen Schema. Die Wahl der Repräsentationsform bedingt zugleich die Operationalisierung der elementaren Datenbankfunktionen der *Speicherung von* und des *Zugriffs auf* Informationen mit Algorithmen, wie auch Marr hervorgehoben hat: »[T]he choice of algorithm often depends rather critically on the particular representation that is employed« (Marr 1982: 23).[54] Eine rein semantische Modellierung von Information im konzeptuellen Schema ist demzufolge ein Ideal, welches in der medialen Praxis digitaler Datenbanken stets unterlaufen wird.[55] Dies stellt keinen Mangel der Drei-Ebenen-Datenbankarchitektur dar, sondern verweist vielmehr auf ein Charakteristikum der computertechnischen Verarbeitung von Informationen, wenn nicht sogar des Umgangs mit Informationen im Allgemeinen. Informationen bedürfen einer Verkörperung als mediale Konstellation und die jeweilige Form der Verkörperung bedingt die möglichen Weisen des Umgangs mit ihnen.[56] Dies wird im Folgenden am

---

zeptuellen Schemas erlauben, welches im Datenbanksystem auf unterschiedliche Weise umgesetzt werden kann. Ebenso gesteht Bachman ein, dass diese Idee nie in DBMS umgesetzt wurde. Die Idee des konzeptuellen Schemas sei vielmehr in Produkte eingegangen, »which sit on top of, and are independent of particular database systems. They have the ability to take their data model and translate it to meet the requirements of various different database systems« (Bachman 2006: 133).

54 | Darüber hinaus weist Marr darauf hin, dass vor dem Hintergrund bestimmter Repräsentationen jede Funktion durch verschiedene Algorithmen realisiert werden kann: »[E]ven for a given fixed representation, there are often several possible algorithms for carrying out the same process« (Marr 1982: 23).

55 | Hierauf weisen die Autoren des *Integration Definition For Information Modeling* Standards hin: »The logical data structure of a DBMS, whether hierarchical, network, or relational, cannot totally satisfy the requirements for a conceptual definition of data because it is limited in scope and biased toward the implementation strategy employed by the DBMS. Therefore, the need to define data from a conceptual view has led to the development of semantic data modeling techniques« (National Institute of Standards and Technology 1993). Zur abstrakten Beschreibung des konzeptuellen Schemas einer Datenbank hat sich das Entity/Relationship-Modell durchgesetzt, welches von Peter Chen (1976) entwickelt wurde.

56 | Nicht zuletzt weil die ANSI/X3/SPARC-Datenbankarchitektur gegenüber den verschiedenen konkurrierenden Datenmodellen indifferent war, wurde der Vorschlag der Study Group on DBMS von den Befürwortern der unterschiedlichen Ansätze zur Datenmodellierung positiv aufgegriffen, sodass es sich als Modell der Handhabung digitaler Informationen durchsetzen konnte.

Beispiel des relationalen Datenmodells dargelegt, das spätestens seit Anfang der 1980er Jahre *de facto* der Standard der Informationsmodellierung in DBMS ist. Dabei hat sich das relationale Datenmodell gegenüber dem hierarchischen Datenmodell und dem Netzwerkdatenmodell durchgesetzt. Das relationale Datenmodell entfaltet eine mediale Eigenlogik, die besonders gut im Kontrast zum Netzwerkdatenmodell sichtbar wird. Vor dem Hintergrund der Analyse des relationalen Datenbankparadigmas werden weitere Modi der Herstellung computer-lesbarer Signifikanz betrachtet: Websuchmaschinen einerseits und das Semantic Web andererseits.

Das relationale Paradigma

Algorithmen als geregelte Prozeduren zur automatischen Lösung von Problemen und Repräsentationen als Formen der Verkörperung digitaler Informationen sind nicht unabhängig voneinander, sondern stehen in einem wechselseitigen Bedingungsverhältnis. Dies hat einen Einfluss auf die Konstruktion und den Gebrauch von digitalen Datenbanken. Wenn in den vorangegangenen Abschnitten die Bedeutung der Drei-Ebenen-Datenbankarchitektur und des konzeptuellen Schemas herausgearbeitet wurde, sollen im Folgenden Verfahren der Modellierung von Information im konzeptuellen Schema beschrieben werden. Denn im Kontext von verschiedenen als Datenmodelle bezeichneten Modellierungsverfahren nimmt die *Automatisierung* der elementaren Datenbankoperationen, d.h. des Speicherns, Abfragens, Veränderns und Löschens von Informationen, unterschiedliche Gestalt an. Der in den 1970er Jahren ausgetragene Konflikt zwischen den Befürwortern der verschiedenen Datenmodelle richtete sich demzufolge nicht nur auf die formalsemantische Ausdrucksstärke der unterschiedlichen Modellierungsverfahren. Diskutiert wurde auch, wie Nutzer mit Datenbanken interagieren können sollen und welche Rolle in diesem Zusammenhang den Administratoren von Datenbanken respektive den Programmierern von Datenbankanwendungen zukommen soll. Bevor das relationale Datenmodell im Detail analysiert wird, soll dessen Siegeszug in den vergangenen 30 Jahren in einem kurzen Abriss dargestellt werden – auch um deutlich zu machen, warum dieses Modell gegenwärtig zunehmend verdrängt wird.

Im Lauf der 1980er Jahre setzten sich das von Edgar F. Codd entwickelte relationale Datenmodell und die darauf aufbauende Datenbanksprache SQL als Standards der Modellierung und Verwaltung von Informationssammlungen in digitalen Datenbanken durch. Nicht unmaßgeblich hierfür war die Markteinführung des *Oracle V2*-Datenbankmanagementsystems (1979) und von IBMs DB2 (1983).[57] Diese Systeme bewiesen durch ihre hohe Anpassungsfähigkeit die praktische Leistungsfähigkeit des relationalen Ansatzes, wodurch dem bis dahin eher theoretisch

57 | *Oracle V2* und *IBM DB2* waren die ersten kommerziellen relationalen DBMS (rDMBS). Zuvor wurde bei IBM das experimentelle DBMS *System R* entwickelt und an der Universität Berkeley das rDBMS *Ingres*.

geführten Disput, den vor allem die Anhänger des relationalen Lagers mit den Befürwortern des von der CODASYL Data Base Task Group entwickelten Netzwerkdatenmodells ausfochten, praktisch ein Ende gesetzt wurde.[58]

Hierarchische Datenbanken und Netzwerkdatenbanken wurden in Folge des Erfolgs relationaler Datenbanktechnologien zunehmend in die Nische hochspezialisierter Anwendungen verdrängt.[59] Auch die im Laufe der 1980er und 1990er Jahre entwickelten objektorientierten Datenbanksysteme vermochten die relationalen Datenbanken nicht zu verdrängen, weshalb der Begriff der Datenbank im Kontext der Informatik heute noch immer weithin gleichbedeutend ist mit relationale Datenbank. Symptomatisch für die langjährige Dominanz des relationalen Paradigmas ist das seit einigen Jahren immer populärer werdende Stichwort NoSQL, welches als *umbrella term* unterschiedlichste Entwicklungen im Bereich digitaler Datenbanktechnologien zusammenfasst, deren kleinster gemeinsamer Nenner das Ziel ist, Alternativen zur relationalen Datenverarbeitung und zu SQL zu entwickeln (vgl. Edlich et al. 2010: 1ff.).

Nach dreißigjähriger Vorherrschaft kommt das relationale Datenmodell angesichts der Datenmenge, die Web 2.0-Angebote wie z.B. Facebook, Amazon, Twitter, Google etc. produzieren, vermehrt an seine Grenzen. Automatisch zu verwalten sind nicht mehr nur große Daten- und Informationssammlungen, sondern *Big Data*. Damit sind Informationsmengen gemeint, die durch etablierte Strategien und Technologien zur Datenbankverwaltung nicht mehr zu bewältigen sind.[60] Insbesondere bei erfolgreichen Web 2.0-Services kommen deshalb Technologien zum Einsatz, die nicht mehr dem relationalen Datenmodell verpflichtet sind und dabei zugleich mit der *artifiziellen Physik* digitaler Informationen des traditionellen Datenbankmanagements brechen, wie sie in der ANSI/X3/SPARC-Architektur entworfen wurde. Abschied genommen wird implizit oder explizit ebenso vom Ziel der Datenunabhängigkeit wie von dem Streben nach strenger Konsistenz, d.h. inhaltlicher Zuverlässigkeit der gespeicherten Informationen. Die Entkopplung der internen Speicherlogik digitaler Informationen von ihren externen Gebrauchslogiken, welche Anfang der 1970er Jahre als eine der Hauptaufgaben digitaler Datenbanktechnologien identifiziert wurde, wird zunehmend aufgehoben.[61] Dies befördert

58 | Bachmann selbst blieb bis ans Ende seine Lebens dem relationalen Datenmodell gegenüber skeptisch (vgl. Bachman 2006: 104f.).

59 | Mit dem Aufkommen postrelationaler Datenbanken gewinnen hierarchische Datenstrukturen und Netzwerkdatenstrukturen erneut an Relevanz. Beispiele hierfür sind XML-Datenbanken und Graphdatenbanken (vgl. Edlich et al. 2010: 207ff.)

60 | Zum Schlagwort Big Data und zu den damit verbundenen Chancen und Herausforderungen siehe exemplarisch Bollier (2010).

61 | Ein Charakteristikum von NoSQL-Datenbaken ist nach Ansicht von Edlich et al., dass diese weitgehend ohne konzeptuelles Schema funktionieren: »Das System ist schemafrei oder hat nur schwächere Schemarestriktionen« (Edlich et al. 2010: 2). Infolgedessen können die in der Datenbank gespeicherten Informationen vom

die Herausbildung von technisch abgeschlossenen Informationssilos im Internet, deren Offenheit und Anschlussfähigkeit nach außen hin nicht in die Gestaltung der technischen Informationsinfrastruktur eingeschrieben ist, sondern nur ein kontingentes *Feature* des Service darstellt, über das die Nutzer keine Kontrolle haben und auf das sie sich in letzter Konsequenz nicht verlassen können.[62] Die Abkehr von der Konsistenzforderung hat darüber hinaus zur Konsequenz, dass die auf eine Suchanfrage zurückgegebenen Informationen nicht notwendig dem neuesten Stand entsprechen und dementsprechend nicht mehr aktuell oder sogar falsch sind. Unter den Bedingungen millionenfacher Änderungen und Abfragen im Echtzeitbetrieb begnügt man sich mit *Eventual Consistency* (vgl. Vogels 2008). *Real Time* bedeutet dabei stets unsichere Information, was durch das Versprechen der Aktualität des Echtzeitbetriebs systematisch verdeckt wird.

Welche genauen Auswirkungen diese jüngsten Veränderungen in der medientechnischen Infrastruktur von Informationssystemen auf die digitale Medienkultur haben werden, die sich in ihrem Rahmen etabliert, lässt sich noch nicht absehen. Um die sich abzeichnenden Transformationen beschreiben zu können, gilt es zunächst das relationale Paradigma genauer zu betrachten. Einen Ansatzpunkt bietet der in den 1970er Jahren ausgefochtene Disput um das beste Datenmodell (vgl. hierzu auch Gugerli 2007a, 2009). Als Gegenspieler Codds trat insbesondere Charles Bachman auf, der nicht nur als einer der Erfinder des Netzwerkdatenmodells gilt, sondern auch sein vehementester Fürsprecher war.[63]

Den Kern des Streits bildete die Frage, welche *Mathematik* der Verwaltung von Informationssammlungen in Datenbanken zugrunde gelegt werden soll. Diese Grundsatzentscheidung hatte weitreichende Konsequenzen für die Gestaltung und Verwendung von Datenbanksystemen. Bachman rückt den Programmierer als zentrale Vermittlungsfigur ins Zentrum, der als »Navigator, Architect, Communicator, Modeler, Collaborator, and Supervisor« (Bachman 1987: 281) fungieren soll. Demgegenüber fordert Codd die Ermächtigung der Endnutzer, die in die Lage versetzt

---

DBMS nicht mehr als Informationen adressiert werden. Die Attribuierung und Verwaltung von Bedeutung wird auf die Ebene des Informationssystems verlagert.

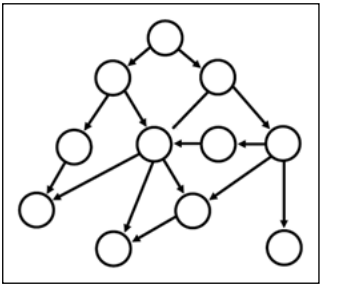
62 | Dies wird immer dann offenkundig, wenn Serviceanbieter Dienste einstellen oder deren Gebrauch einschränken, wie z.B. durch Änderungen am Application Programming Interface (API).

63 | Es gilt anzumerken, dass Bachman der Bezeichnung des CODASYL-Datenmodells als Netzwerkdatenmodell skeptisch gegenüber stand: »Let me try to answer why I have not used the word ›network‹, although I think many people would characterize IDS and DBTG-like systems as network systems. The reason I avoided it is that the basic data-structure-set (or owner-coupled set as Ted Codd described it) is simply a facility for describing an owner-coupled set. With such a mechanism you can describe very complex hierarchies, you can describe trees (somewhat in contradiction to what Ted said), and you can describe networks of like objects or networks of unlike objects« (Anonymus 1974: 123f.).

werden sollen, auch eigenständig und unabhängig von Datenbankadministratoren und Anwendungsprogrammieren mit Datenbanken zu interagieren. Codd, dem 1981 – acht Jahre nach Bachman – der renommierte Turing-Award verliehen wurde, stellt dies in seinem Vortrag anlässlich der Preisverleihung deutlich heraus. Ein Ziel relationaler Datenbanksysteme sei es, die Nutzer mit Datenbankinformationen in direkten Kontakt zu bringen: »put end users into direct touch with the information stored in computers« (Codd 1982: 109).

Die Suchmöglichkeiten der Nutzer sollten nicht von vorprogrammierten Suchroutinen (in Anwendungsprogrammen) abhängen. Im Gegenteil, die Leistungsfähigkeit eines Datenbanksystems ist nach Ansicht Codds daran zu messen, ob es seine Nutzer in die Lage versetzt, selbständig Suchanfragen zu formulieren. Sofern diese Forderung durch relationale DBMS erfüllt wird, zeichnet sich mit deren Aufkommen der »Übergang von der gezielten Suche nach Einträgen hin zur Recherche als einer ergebnisoffenen Abfrage, also ein Übergang zur rechnergestützten Befragung des Orakels statt« (Gugerli 2009: 72). Die Übersetzung eines nutzerseitigen Informationsbedürfnisses (Was) in eine Suchprozedur (Wie) leistet im Rahmen des relationalen Modells nicht der Programmierer, sondern der Computer, wodurch der Eindruck der direkten Interaktion mit Information entsteht. Der Speicherraum der Datenbank wird für den Suchenden opak; er ist eine Black Box, der gegenüber Suchanfragen formuliert werden, die der Computer automatisch in Ergebnisse übersetzt.

Abb. 13: Visualisierung des Netzwerkdatenmodells als gerichteter Graph



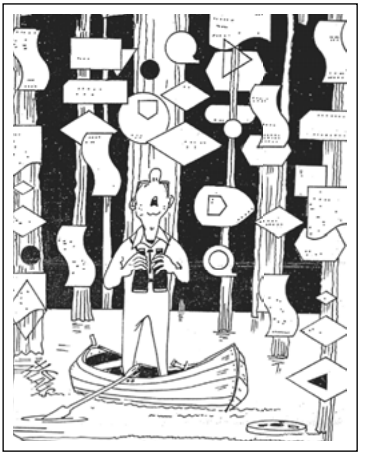
Quelle: CODASYL Data Base Task Group 1971: 61

Das Netzwerkdatenmodell privilegiert eine andere Form der Suche. Es basiert wie auch das hierarchische Modell auf einer Strukturierung von Informationen in Form von Graphen.[64] Entitäten werden im Netzwerkdatenmodell als Record-

64 | Hierarchische Strukturen und Netzwerkstrukturen sind nicht grundlegend verschieden, da Hierarchien als ein besonderer Typ von Netzwerken begriffen werden können, bei dem die möglichen Beziehungen zwischen Knoten eingeschränkt sind. Abgesehen vom Wurzelknoten hat jeder Knoten (Record-Typ) in einer Hierarchie genau einen Vorgänger. Infolgedessen können in Hierarchien nur Beziehungen der

Typen modelliert, welche die Knoten in einem Informationsnetzwerk bilden, deren Beziehungen zueinander als gerichtete Kanten dargestellt werden und als Verweise materialisiert werden.

Abb. 14: Navigieren im Datenraum: »Navigare necesse est«



Quelle: Bachman 1973a: 867[65]

Datenbankeinträge werden in dieser Netzwerkstruktur abgebildet und entsprechend gespeichert (vgl. Abb. 13). Damit verpflichtet sich dieses Modell implizit der mathematischen Graphentheorie, wodurch die Suche nach Information in der Datenbank zu einem Navigationsproblem wird. Anders als bei dem berühmten Königsberger Brückenproblem geht es nicht um das Entdecken eines beliebigen Wegs durch einen Graphen, sondern um die Festlegung eines bestimmten Suchpfads, der zu spezifischen Informationen führt.[66] Das Finden von Information fällt im Rahmen des Netzwerkdatenmodells mit dem Folgen eines bestimmten Suchwegs in eins. Im Meer von Datenbankinformationen kann sich nur orientieren, wer sich im konzeptuellen Informationsnetzwerk der Datenbank zu bewegen weiß (vgl. Abb. 14). Was heißt das?

Kardinalität 1:n abgebildet werden, d.h. es wird eine Eltern-Kind-Beziehung etabliert, bei der jeder Elternknoten eine Vielzahl von Kindknoten haben kann, aber jeder Kindknoten nur genau einen Elternknoten besitzt.

65 | Die mit »Navigare necesse est« betitelte Visualisierung ist der deutschen Übersetzung von Bachmans Turing Award-Rede entnommen, die im Dezember 1973 in *Online: Zeitschrift für Datenverarbeitung* veröffentlicht wurde.

66 | Das Königsberger Brückenproblem, welches im 18. Jahrhundert von Euler mathematisch beschrieben wurde, behandelt die Frage, ob es einen Weg durch Königsberg gibt, bei dem alle sieben Brücken der Stadt genau einmal überquert werden. Euler bewies, dass dies nicht möglich sei.

Der Suchpfad kann vom Computer nicht automatisch gefunden, sondern muss gegenüber dem Datenbanksystem definiert werden, was nach Ansicht von Bachman nicht von den Nutzern der Datenbank zu leisten sei, sondern eine Programmieraufgabe darstelle. Bei der Suche nach Informationen in der Datenbank bedarf es demzufolge der Vermittlungsinstanz des Programmierers, der Suchpfade deklariert und hierdurch Formen von Suchanfragen festlegt, die überhaupt gegenüber einer Datenbank formuliert werden können. Ein Beispiel für eine solche Anfrageform ist: »Finde alle Alben, welche die Band X bei dem Musiklabel Y herausgebracht hat!« Der Nutzer kann die Variablen X und Y der Suchroutine anpassen, die Form der Suche vermag er ohne weitreichende Kenntnisse des Informationsmodells jedoch nicht zu ändern. An der Oberfläche erweist sich die Suche in Netzwerkdatenbanken daher als bloße Variation von Parametern für Suchroutinen, die unsichtbar in der Tiefe ablaufen und deren Semantik vordefiniert ist.

Das Hauptproblem des Netzwerkdatenmodells und seiner Suchlogik ist, dass sich in dem Datenbanksystem nicht formulieren lässt, *was* gesucht wird. Ohne die Übersetzung eines Informationsbedürfnisses in eine Findroutine kann in derartigen Datenbanken nicht gesucht und nichts gefunden werden. Dass diese Übersetzungsleistung im Rahmen des Netzwerkdatenmodells nur von Programmierern vollbracht werden kann, hat Bachman wahrscheinlich zu Recht vermutet. Seine Überzeugung, dass die effiziente Suche in digitalen Datenbanken prinzipiell eines menschlichen Navigators bedarf, hat sich jedoch als falsch erwiesen. Der Gegenbeweis wurde in den 1980er Jahren erbracht, als sich relationale Datenbanksysteme in der Praxis zunehmend bewährten. Die theoretischen Grundlagen hat Codd mit der Formulierung des relationalen Datenmodells gelegt, welches den Umgang mit Informationen in Datenbanken als mengentheoretisches Problem begreift.

Grundbegriff des von Codd entwickelten Datenmodells ist die Relation, der einen besonderen Typus von Mengen bezeichnet. Eine Relation ist eine Menge von (Unter-)Mengen, die Tupel genannt werden. Jedes Tupel einer Relation ist eine Zusammenstellung von Elementen, wobei alle Tupel dieselbe Struktur aufweisen. Dies wird auch in der von Codd zugrunde gelegten Definition deutlich, in der Tupel als Gruppierung von Elementen aus Wertebereichmengen beschrieben werden:

»Given sets S_1, S_2,..., S_n (not necessarily distinct), *R* is a relation on these *n* sets if it is a set of *n*-tuples each of which has its first element from S_1, its second element from S_2, and so on. We shall refer to S_j as the *j*th *domain* of *R*.« (Codd 1970: 379)

Die in Tupeln zusammengefassten Elemente (Informationspartikel) beziehen sich im relationalen Datenmodell auf eine Entität, über die sie informieren. Da die konkrete Anordnung der Werte in einem Tupel für den Nutzer irrelevant sein soll, muss den Elementen des Tupels ein Typ zugewiesen werden, der es erlaubt, diese als Attribut einer Entität und damit als Information mit einer bestimmten Bedeutung zu adressieren. In dieser Hinsicht unterscheidet sich Codds Begriff der Relation von

dem in der Mathematik geläufigen Verständnis.[67] Dessen ungeachtet bildet die mathematische Mengenlehre den Eckpfeiler des relationalen Datenmodells, mit der ein Perspektivwechsel auf die Modellierung von Informationen im konzeptuellen Schema einhergeht.

Zwar steht im relationalen Datenmodell auch die formale Beschreibung von Entitäten und deren Beziehungen zueinander im Vordergrund. Diese werden aber als Relationen konzeptualisiert, welche Mengen von n-Tupeln sind und damit streng genommen Sammlungen von Informationssammlungen über Entitäten. Oder anders formuliert: Auf der Grundlage der Mengenlehre werden Entitäten als Informationssammlungen modelliert, die Teil der Informationssammlung »Datenbank« sind.[68] Hierdurch wird der Unterschied zwischen dem Einen (Informationen über eine Entität) und dem Vielen (Sammlungen von Informationen über Entitäten) unterlaufen. Ob eine Relation nur ein Tupel beinhaltet oder eine große Zahl an Tupeln ist in mengentheoretischer Hinsicht irrelevant, es handelt sich gleichermaßen um eine Relation. Insofern lässt sich der Übergang vom Vielen zum Einen als Übersetzung einer Relation in eine andere begreifen und formalisieren.

Zudem wird die Unterscheidung von Entitäten und Beziehungen aufgelöst, welche im hierarchischen Datenmodell und im Netzwerkdatenmodell von zentraler Bedeutung war. Im relationalen Datenmodell ist der Unterschied zwischen diesen relativ. Entitäten werden als Relation, d.h. Beziehung, von Werten in einem Tupel begriffen und Beziehungen zwischen Entitäten werden in Relationen abgebildet. Die in Netzwerkstrukturen als Linien (Kanten) modellierten Beziehungen werden in Informationen übersetzt, die ebenso in der Datenbank zu speichern sind wie die Informationen über Entitäten. Relationale Datenbanken deklarieren Beziehungen durch sogenannte Fremdschlüsselattribute in einer Relation, welche die Identifikation von Entitäten anderer Relationen erlauben, wie das Beispiel in Abb. 15 zeigt.[69] In diesem vereinfachten Modell gibt es die drei Relationen *Buch, Autor* und *Buchautor*. Bücher werden in der Relation *Buch* durch eine Identifikationsnummer, einen Titel, den Verlag und das Erscheinungsjahr beschrieben. Die Informationen über Autoren werden in einer anderen Relation gespeichert, die neben einer Identifikationsnummer den Namen, den Vornamen sowie weitere Informationen über einen Autor

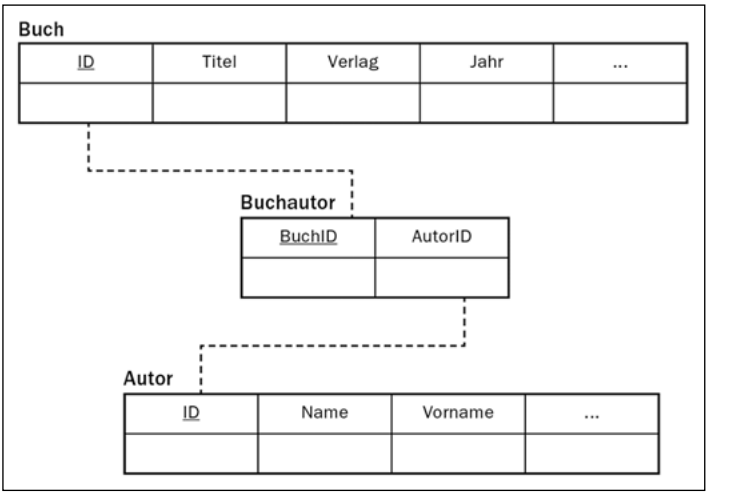
67 | Auf diesen Umstand weist Codd selbst hin. Bei Relationen im Kontext des relationalen Datenmodells handelt es sich mengentheoretisch betrachtet streng genommen um *Relationship*s, welche im Unterschied zum mathematischen Begriff der Relation keine feste Anordnung der Elemente eines Tupels erfordern (vgl. Codd 1970: 374).

68 | Das relationale Datenmodell erlaubt es zudem, die Datenbank als Ganzes zu beschreiben. Eine Datenbank ist eine variable Informationssammlung auf Grundlage eines konzeptuellen Schemas: »The totality of data in a data bank may be viewed as a collection of time-varying relations« (Codd 1970: 379).

69 | Innerhalb einer Relation können Entitäten durch sogenannte Primärschlüssel eindeutig identifiziert werden, die ein oder mehr Attribute umfassen.

beinhaltet. Welche Autoren an welchen Büchern mitgewirkt haben ist schließlich in der Relation *Buchautor* gespeichert. Diese enthält die Attribute *BuchID* und *AutorID*, die als Fremdschlüsselattribute fungieren und auf Entitäten in den Relationen *Buch* und *Autor* verweisen. Hierdurch wird es möglich, jedem Buch mehrere Autoren zuzuweisen. Zugleich existiert zu jedem Autor nur ein Eintrag, weshalb Änderungen an den Autorinformationen nur an einer Stelle in der Relation *Autor* gespeichert werden müssen und nicht in den Einträgen zu jeder Publikation, die ein Autor verfasst hat.

Abb. 15: Beispiel eines relationalen Modells einer Datenbank



Das mengentheoretische Konstrukt der Relation erlaubt nicht nur die Modellierung von Informationssammlungen, sondern auch die mathematisch präzise Spezifikation von Operationen mit diesen. Als Grundoperationen identifiziert Codd die *Selektion*, die *Projektion* und den *Verbund*, welche die Kernelemente der relationalen Algebra bilden und Transformationen von Relationen in Relationen definieren.[70] Die Selektion betrifft die Auswahl von Tupeln einer Relation, die Projektion die Rekonfiguration einer Relation und der Verbund die Verknüpfung von

70 | Neben der Selektion, dem Verbund und der Projektion gehören die Bildung der Vereinigungsmenge und der Differenzmenge sowie die Umbenennung von Attributen respektive Relationen zu den Operationen, die zur Verfügung gestellt werden müssen, um alle Ausdrücke der relationalen Algebra vollständig abbilden zu können. Dennoch erweisen sich die drei erstgenannten Operatoren als grundlegend für relationale Datenbanksysteme, denn »[m]uch of the derivability power of the relational algebra is obtained from the SELECT, PROJECT, and JOIN operators« (Codd 1982: 112)

Relationen. Diese Operationen sind für Codd der »yardstick of power« (Codd 1982: 112), an dem Implementierungen des relationalen Datenmodells zu messen sind.[71]

Die Mathematik bildet nur eine Seite des relationalen Datenmodells. Ebenso wichtig ist nach Ansicht von Codd die Repräsentation von Relationen als Tabellen. Die diagrammatische Form der Tabelle stellt seines Erachtens die geeignete »conceptual representation« (Codd 1982: 111) für Relationen dar, welche als *Denkfigur* das Nachdenken über Operationen in Datenbanken und die Verständigung über deren Sinn respektive Unsinn ermöglicht: »Such a representation facilitates thinking about the effects of whatever operations are under consideration. It is a requirement for programmer and end-user productivity« (Codd 1982: 111).[72] Die Tabelle spannt als Diagramm eine zweidimensionale Matrix auf, welche die »Verteilung von Zeichen auf Zeilen und Spalten« (Krajewski 2007: 37) organisiert. Das Tupel einer Relation entspricht der Zeile in einer Tabelle, in der demzufolge Informationen über eine Entität zusammenfasst sind. Die Attribute, durch die Entitäten beschrieben werden, sind in den Spalten angeordnet, welche Informationen eines Typs zusammenfassen.

Anhand der Denkfigur der Tabelle lassen sich die Grundoperationen der relationalen Algebra verdeutlichen. Bei der Selektion werden bestimmte Zeilen einer Tabelle ausgewählt. Die Projektion erlaubt die Auswahl von Spalten. Durch die Verbundoperation werden zwei Tabellen zusammengeführt. Die Ergebnistabelle umfasst die Spalten der Ursprungstabellen und sämtliche Kombinationen der Zeilen aus beiden Tabellen, sofern keine Kriterien angegeben sind, welche deren Kombination einschränken. Ein sinnvolles Kriterium stellen die bereits erwähnten Fremdschlüsselbeziehungen dar. Die Verbindung jedes Buchs mit jedem Autor ist unsinnig, da hierdurch Entitäten mit Informationen in Beziehung gesetzt werden, die nicht zutreffend sind. Nur auf Grundlage der in der Relation *Buchautor* gespeicherten Fremdschlüsselattribute können Bücher mit ihren jeweiligen Autoren verbunden werden.

---

71 | Im Rahmen des Netzwerkdatenmodells können nach Ansicht von Codd keine äquivalenten Operationen realisiert werden, was in seinen Augen einen zentralen Nachteil dieses Modells darstellt. Dementsprechend stellt Codd in einer Debatte mit Bachman fest: »I do not know how to manipulate network or graph-oriented representations of data in a manner analogous to the set-theoretic. Someone may invent this, but as of now I do not know just what operators I would place on top of a network or graph-oriented representation of the data in order to obtain the same power of retrieval you can obtain with either the algebraic operators on relations or with the predicate calculus« (Anonymus 1974: 134).

72 | Ein Datenmodell, welches eine ernstzunehmende Alternative zum relationalen Modell sein will, muss nach Ansicht von Codd ebenfalls über eine anschlussfähige konzeptuelle Repräsentation verfügen: »Incidentally, if a data model is to be considered as a serious alternative for the relational model, it too should have a clearly defined conceptual representation for database instances« (Codd 1982: 111).

Obwohl die Tabelle nach Ansicht von Codd eine geeignete Repräsentation der relationalen Mengenlogik darstellt, sind diese nicht miteinander gleichzusetzen. Tabelle und Relation befinden sich nicht auf demselben Abstraktionsniveau:

»Tables are at a lower level of abstraction than relations, since they give the impression that positional (array-type) addressing is applicable (which is not true of n-ary relations), and they fail to show that the information content of a table is independent of row order. Nevertheless, even with these minor flaws, tables are the most important conceptual representation of relations, because they are universally understood.« (Codd 1982: 111)

Einerseits eignen sich Tabellen, um die in der relationalen Algebra beschriebenen Operationen zu veranschaulichen. Andererseits aber erweist sich der Verweis auf Tabellen als suggestiv. Die in diesen realisierte »diagrammatische Relation aus Zeilen und Spalten« (Krajewski 2007: 37) legt Umgangsformen mit Datenbankinformationen nahe, die im relationalen Datenmodell unerwünscht sind. So können in Tabellen Informationen gefunden werden, indem man deren Position beschreibt (z.B. gehe hundert Zeilen nach unten und drei Zeilen nach rechts). Die konkrete Anordnung der Zeilen und Spalten soll Codd zufolge für die Nutzer einer relationalen Datenbank jedoch irrelevant sein. Daher wird im relationalen Paradigma die assoziative Adressierung von Informationspartikeln privilegiert:

»In the relational model we replace positional addressing by totally associative addressing. Every datum in a relational database can be uniquely addressed by means of the relation name, primary key value, and attribute name. Associative addressing of this form enables users (yes, and even programmers also!) to leave it to the system to (1) determine the details of placement of a new piece of information that is being inserted into a database and (2) select appropriate access paths when retrieving data.« (Codd 1982: 111)

Das Auffinden des Orts von spezifischen Informationen in der Datenbank wird als Aufgabe an den Computer delegiert. Hierdurch wird die Formulierung festgelegter Suchpfade überflüssig, welche im Rahmen des Netzwerkdatenmodells notwendig war. Nicht der Programmierer übersetzt das Informationsbedürfnis in eine Findroutine, sondern der Computer, dem gegenüber Suchbedingungen deklariert werden.

Suchanfragen nehmen in relationalen Datenbanksystemen nicht die Gestalt *prozedural-imperativer* Befehlsfolgen an, welche gegenüber dem Computer festlegen, *wie* eine Information im Bestand der Datenbank gefunden werden kann.[73] Das Informationsbedürfnis wird *deklarativ* formuliert, wobei festgelegt wird, *was* man

73 | Henning und Vogelsang beschreiben das Charakteristikum imperativer Programmiersprachen wie folgt: »Bei imperativen (oder prozeduralen) Sprachen besteht ein

sucht und nicht, *wie* es gefunden werden kann. Als Standardsprache relationaler Datenbanksysteme hat sich die *Structured Query Language* (SQL) etabliert, die von Donald D. Chamberlin und Raymond F. Boyce bei IBM im Anschluss an Codd entwickelt und spezifiziert wurde (vgl. Chamberlin/Boyce 1974).[74] In SQL werden Suchanfragen als *Select-From-Where*-Block formuliert. Die *Select*-Klausel legt fest, welche Attribute in der Ergebnismenge enthalten sein sollen, d.h. die Spalten einer Tabelle. In der *From*-Klausel wird bestimmt, welche Relationen in die Suche einbegriffen werden, und die *Where*-Klausel gibt die Kriterien an, welche die gesuchten Tupel erfüllen müssen.[75] Dies stellt die elementare Anfrageform in SQL dar, welche die in der relationalen Algebra definierten Operationen der Selektion, Projektion und des Verbunds miteinander verschaltet.

In ihrer ersten Publikation zu SEQUEL führen Chamberlin und Boyce eine Reihe von Beispielen an, wie konkrete Informationsbedürfnisse in diese Anfrageform übertragen werden können. Das erste und einfachste Beispiel, welches die Autoren anführen, ist die Suche nach den Namen (NAME) aller Angestellten (EMP[loyees]), die in der Spielzeugabteilung (DEPT = ›TOY‹) eines Unternehmens arbeiten. In SQL-Pseudocode lässt sich dies wie folgt ausdrücken:

»SELECT NAME
FROM EMP
WHERE DEPT = ›TOY‹« (Chamberlin/Boyce 1974: 253)

An diesem Beispiel wird deutlich, dass die Selektionsoperation der relationalen Algebra und die *SELECT*-Anweisung in SQL nicht miteinander gleichzusetzen sind. Die *Select*-Klausel legt nicht fest, welche Entitäten in der Ergebnismenge enthalten sein sollen, sondern gibt an, welche Attribute von Interesse sind. Sie ermöglicht

---

Programm aus einer Folge von Befehlen an den Rechner, z.B. ›Schreibe in Variable x den Wert 5‹« (Henning/Vogelsang 2004: 33).

74 | Zunächst führten Chamberlin und Boyce SQL als *Structured English Query Language* ein, für die sie das Akronym SEQUEL gebrauchten. Aus Urheberrechtsgründen wurde SEQUEL später in SQL umbenannt (Frana 2001: 17). Neben SEQUEL wurden noch weitere Sprachen für das relationale Modell vorgeschlagen, die im Rahmen verschiedener Entwicklungsprojekte entstanden sind (vgl. Gugerli 2009: 81). In der Praxis hat sich jedoch SQL durchgesetzt, das erstmals 1986 vom *American National Standard Institute* als Standard festgeschrieben wurde. Seither hat der Standard, der von der *International Standards Organization* übernommen wurde, eine Reihe von Revisionen durchlaufen.

75 | Chamberlin und Boyce beschreiben die elementare Form der SQL-Anfrage wie folgt: »The user must specify the columns he wishes to SELECT, the table FROM which the query columns are to be chosen, and the conditions WHERE the rows are to be returned. The SELECT-FROM-WHERE block is the basic component of the language« (Chamberlin/Boyce 1974: 254).

die Projektion von Spalten der Ausgangsrelation in Spalten der Ergebnisrelation. Demgegenüber werden die Selektionsbedingungen für Tupel in der *Where*-Klausel festgelegt.[76] Der Verbund kann implizit realisiert werden, indem man mehrere Relationen in der *From*-Klausel aufführt. Darüber hinaus verfügt die Sprachspezifikation von SQL über Möglichkeiten, verschiedene Formen der Verbundoperation durch explizite *JOIN*-Anweisungen zu deklarieren. Durch die Einführung dieser Operatoren wurden die Ausdrucksmöglichkeiten von SQL zwar teilweise erweitert.[77] In erster Linie dienen sie jedoch dem Ziel, die Formulierung komplexer Suchanfragen zu erleichtern (vgl. Saake et al. 2008: 222). Die unterschiedlichen Formen des Verbunds zwischen Relationen sind für die Suche in relationalen Datenbanken von zentraler Bedeutung. Sie erlauben es, Informationen zueinander in Beziehung zu setzen und hierdurch zu Informationen zu gelangen, die nicht explizit in der Datenbank gespeichert sind.

Die Beziehungen zwischen Entitäten werden in relationalen Datenbanken nicht als feste Verweisstruktur in der Tiefe der Datenbank etabliert, sondern müssen vom Nutzer an der Oberfläche bei der Formulierung von Suchanfragen hergestellt werden.[78] Dies ist eine Konsequenz des von Codd geforderten Verzichts auf »user-visible navigation links between tables« (1982: 113), wie sie im Netzwerkdatenmodell als Kanten zwischen Knoten dargestellt werden und denen bei der Suche nach Informationen in einer solchen Datenbank gefolgt werden muss. Im Hintergrund dieses Verbots steht Codds Überzeugung, dass die Verortung von Datenbankinformationen in einer Netzwerkstruktur von Anwendungsprogrammieren viel und von Endnutzern zu viel Vorwissen über die interne Struktur der Datenbank abverlangt: »It [...] places a heavy and unnecessary navigation burden on application programmers. It also presents an insurmountable obstacle for end users« (Codd 1982: 110, Fn 1).

Diametral hierzu steht Codds Postulat, dass die Nutzer von Datenbanken von der internen Organisation der Informationen im Computerspeicher abgeschirmt

---

76 | Die *WHERE*-Klausel ist optional und kann somit entfallen (vgl. Saake et al. 2008: 219).

77 | Viele explizite JOIN-Anweisungen wie der INNER JOIN oder der CROSS JOIN können auch implizit in SQL realisiert werden. Die im SQL-Standard festgelegten Varianten des äußeren Verbunds (LEFT OUTER JOIN, RIGHT OUTER JOIN und FULL OUTER JOIN) können nicht implizit in SQL deklariert werden. Bei diesen Joins werden Tupel einer oder beider Relationen auch dann in die Ergebnismenge aufgenommen, wenn zwischen den Relationen keine Entsprechung beim Vergleichsattribut existiert. Bei derartigen in die Ergebnisrelation übernommenen Tupeln wird den Attributen der jeweils anderen Relation der Wert *NULL* zugewiesen.

78 | Sofern der suchende Zugriff auf eine relationale Datenbank durch ein spezifisches Interface erfolgt, werden die Verbindungen zwischen den Informationen vom Anwendungsprogrammierer durch die Definition von Anfragemustern hergestellt.

werden müssen (vgl. Codd 1970: 377).[79] Dies ermöglicht das relationale Datenmodell mit der bereits erläuterten Ersetzung physischer Links (Pointer) zwischen Datensätzen durch logische Verknüpfungen (Fremdschlüsselbeziehungen) zwischen Relationen. Dies hat zur Folge, dass die Zusammenhänge zwischen Tupeln (Datensätzen über Entitäten) in der Tiefe des Computers nur noch latent gegeben sind und an der Oberfläche vermittels der Verbundoperation aktualisiert werden müssen.[80]

Ein Nebeneffekt dieser Verlagerung ist die enorme Flexibilisierung von Suchanfragen an der Oberfläche. Das relationale Datenmodell erleichtert die Formulierung von Suchanfragen nicht nur für die »truly *casual* user« (Date/Codd 1975: 95), sondern erweitert auch die prinzipiellen Suchmöglichkeiten. Datenbankinformationen können auf nicht vorhergesehene Weise zueinander in Beziehung gesetzt werden, sodass man zu Informationen gelangen kann, die »quer zu den bereits erfassten Daten« (Gugerli 2007a: 27) liegen. In relationalen Datenbanken kann man durch die Formulierung von Suchanfragen neuartige »Zusammenhänge simulieren, überprüfen und erkennen« (Gugerli 2007a: 30), die während der Formalisierung des konzeptuellen Datenbankschemas noch nicht antizipiert wurden. Daher ist der Einschätzung Gugerlis beizupflichten, dass mit der relationalen Datenbank eine »Suchmaschine von präzedenzloser Radikalität« (Gugerli 2009: 81) geschaffen wurde.

In der Datenbank nach etwas zu suchen bedeutet nicht nur das bloße Auffinden von Informationen in einem Bestand. Die Suche wird zu einem kreativen Akt, bei dem sich unbekannte Zusammenhänge erkunden und erforschen lassen. Aus Bekanntem kann Neues entstehen. Die Datenbank als geschützte Aufbewahrungsstätte vorhandener Informationen wird zur Basis für das Entdecken neuer Informationen. In diesem Zusammenhang lässt sich die Formulierung einer Suchanfrage (mit Verbundanweisung) als Spielzug in dem von Lyotard beschriebenen Spiel vollständiger Information verstehen (Lyotard 2009 [1979]: 126).[81] Ein solcher Spielzug bringt eine Hypothese zum Ausdruck, deren Gültigkeit nicht allein daraus geschlossen werden kann, dass die Suchanfrage ein Ergebnis liefert. Dies wurde bereits in dem oben genannten Beispiel der Verknüpfung von Informationen über ein Buch mit Informationen über dessen Autor deutlich. Erst die Angabe eines *tertium*

79 | Die von Codd geforderte Nutzerorientierung rief, wie Gugerli darlegt, auch deshalb Widerstand hervor, weil sie einen zentralen Grundsatz guter Programmierung infrage stellte. Aufgrund der begrenzten Leistungsfähigkeit von Computern galt es, möglichst effizient die geringen Rechen- und Speicherkapazitäten auszunutzen (Gugerli 2009: 74).

80 | Eine zweite, für die hier verfolgte Fragestellung weniger bedeutsame Konsequenz ist, dass sich beim Einfügen von Informationen in die Datenbank die Integrität nicht aus den physischen Verknüpfungen zwischen Datensätzen oder Tupeln erschließen lässt, sondern einer logischen Prüfung unterworfen werden muss, wobei die Integritätsregeln im konzeptionellen Schema formal expliziert werden müssen.

81 | Zu Lyotards Beschreibung des postmodernen Wissens als Spiel vollständiger Information siehe auch S. 183f.

comparationis (die Relation *Buchautor* mit den Fremdschüsselattributen *BuchID* und *AutorID*) stellt im Kontext des Beispiels einen verlässlichen Zusammenhang zwischen Büchern und Autoren her. Die spielerische Kombination und Rekombination von Informationen aus der Datenbank erfordert vom Suchenden daher eine quasi-wissenschaftliche Einstellung, welche die Suchanfrage als Experiment begreift, deren Ergebnisse vor dem Hintergrund der Experimentalanordnung – relationale Datenbank, konzeptuelles Datenbankschema, Anfrage – interpretiert und auf ihre Validität (für die Welt außerhalb der Datenbank) überprüft werden müssen. Die Ableitung von neuem Wissen aus bekannten Informationen wird durch die Suchtechnologie unterstützt, stellt jedoch keinen Automatismus dar und darf nicht als solcher verstanden werden.[82] Es handelt sich um eine Form der Aussagenproduktion durch den Suchenden bei der Suche in einem prinzipiell begrenzten Informationsbestand.[83] Relationale Datenbanksysteme flexibilisieren die nutzerseitigen Suchmöglichkeiten.

Diese Möglichkeiten beruhen auf der vorgängigen Explikation eines Informationsmodells im konzeptuellen Schema, das festschreibt, was im Rahmen einer konkreten Datenbank als Information zählt. In Anbetracht dessen erweisen sich die Erwartungen, die an die Suche in relationalen Datenbanken geknüpft werden, mitunter als überzogen. Exemplarisch lässt sich dies an Beiträgen des bloggenden Kulturwissenschaftlers Michael Seemann (mspro) darlegen, nach dessen Ansicht die Medien an ihr Ende gekommen sind und durch die Suche als primärem Modus der Sinnproduktion abgelöst werden (Seemann 2011a). An den Anfang dieser Entwicklung stellt Seemann das relationale Datenmodell und die Anfragesprache SQL:

»Die Datenbank ist flach, sie kennt keine Relevanz, keine Hierarchien, keine Zusammenhänge. Ordnung, Struktur und Verknüpfung entsteht erst in dem Moment, wenn ich meine Query formuliere. In Echtzeit. […] Mit SQL war der Startschuß gegeben für das, was wir in der Bibliothek von Babel als neues Paradigma der Sinngenerierung

82 | Ein Beispiel hierfür ist das Entdecken ökonomischer Trends anhand von Suchanfragen (Choi/Varian 2009).

83 | In Anbetracht dessen erweist sich Lyotards Beschreibung des Spiels vollständiger Information als problematisch. Vollständigkeit ist keineswegs eine Voraussetzung, sondern allenfalls eine Setzung, in der sich das Imaginäre digitaler Datenbanken widerspiegelt. Zwar kann das kombinatorische Spiel mit Datenbankinformationen durch Lyotards Vollständigkeitsannahme legitimiert werden. Jedoch birgt dies die Gefahr von Fehlinterpretationen, wie Danah Boyd und Kate Crawford in Reaktion auf die aktuellen Diskussionen um Big Data kritisch herausgestellt haben: »Interpretation is at the center of data analysis. Regardless of the size of a data set, it is subject to limitation and bias. Without those biases and limitations being understood and outlined, misinterpretation is the result. Big Data is at its most effective when researchers take account of the complex methodological processes that underlie the analysis of social data« (boyd/Crawford 2011: 6).

verstehen können. Die Query löst die Medien ab. Wo vorher der Rufer ins Nichts den Sinn produzierte, steht heute der algorithmische Filter im Alles.« (Seemann 2011b)

Obwohl im bisherigen Verlauf des Kapitels gezeigt wurde, dass durch das relationale Datenmodell eine Form der computergestützten Suche in Datenbanken möglich wurde, die dem Suchvorgang eine neue Qualität verleiht, ist der »algorithmische Filter im Alles«, in dem Seemann die Erfüllung von Borges' Gedankenexperiment der Bibliothek von Babel sieht, eine kritisch zu hinterfragende Imagination.[84] Weder das Bild des Filters noch der Verweis auf Algorithmen vermag die Besonderheit der Suche in relationalen Datenbanken zu erklären.[85] Die Suchmöglichkeiten von SQL basieren nicht auf Verfahren der algorithmischen Sinnzuschreibung, sondern beruhen auf der vorgängigen Formalisierung eines Informationsmodells im konzeptuellen Schema und der strukturierten Speicherung von Informationen nach Vorgabe dieses Modells. Algorithmen bedingen dabei allenfalls *wie schnell* etwas in der Datenbank gefunden wird, aber nicht *was* als Ergebnis für eine Suchanfrage zurückgegeben wird. In einer Select-From-Where-Anweisung sind die Kriterien der Auswahl von Datenbankinformationen vollständig und eindeutig bestimmt. Relationale Datenbanken lassen sich nur hinsichtlich der in ihnen enthaltenen Informationen als *Black Box* betrachten.[86] Die Prinzipien der Suche, d.h. die Regeln der Verkopplung von Suchanfragen mit Suchergebnissen, stellen bei

84 | Borges zeichnet nicht das Bild der perfekten Bibliothek, sondern er führt die Praxis der Versammlung von Büchern an ihre hypothetische Grenze, die keine Vollendung, sondern den Zusammenbruch der Bibliothek bedeutet. Die Bibliothek, welche nicht nur alle geschriebenen, sondern alle möglichen Bücher enthält, ist eine Dystopie, insbesondere auch für den Suchenden. In dem Moment, in dem durch die bloße Kombinatorik von Buchstaben alles mögliche gesagt scheint, ist gerade nichts mehr gesagt. Jeder mögliche Autor vertritt jede mögliche Position und jede Position wird von jedem Autor vertreten. Die Bibliothek in der Extremform des Gedankenexperiments ist keine Bibliothek mehr; ihre Bewohner müssen von Neuem beginnen, Aussagen zu produzieren. Ein möglicher Modus der Sinngenerierung besteht in der Selektion bestimmter Bücher durch die Suche. Hierbei gilt es zu bedenken, dass im Kontext von Borges' hypothetischer Bibliothek der Akt des Suchens nach einem Buch dem Schreiben des Buchs durch den Suchenden gleichkommt. Nur wer den kompletten Text eines zu findenden Buchs als Suchbedingung angibt, kann es eindeutig identifizieren. Damit wird das Suchen jedoch überflüssig, da das Ergebnis der Suche prinzipiell keine neuen Informationen hervorbringen kann.

85 | In *Translation as a Filter* zeigt Naoki Sakai die Polysemie der Metapher des Filters auf, weshalb er die Beschreibung von Übersetzung anhand dieser Metapher kritisiert (Sakai 2010). Analog hierzu bleibt Seemans Vision eines »algorithmischen Filter im Alles« weithin unbestimmt.

86 | Auf Grundlage des relationalen Datenmodells wurde es nach Ansicht von Gugerli möglich, die »Datenbank als Back Box [zu, M.B.] behandeln, an die sich auch Fragen

der Formulierung von SQL-Anfragen hingegen kein Geheimnis dar, welches bei der Interaktion mit der Black Box *Datenbank* gelüftet werden müsste.[87] Hierin unterscheidet sich die Suche in relationalen Datenbanken beispielsweise von der Suche mit Websuchmaschinen. Dies soll im Folgenden genauer erklärt werden.

Websuchmaschinen: Algorithmische Relevanzzuschreibung

Internetsuchmaschinen wie z.B. Google, Bing, Baidu, Yandex etc. machen es sich zur Aufgabe, die Gesamtheit der im World Wide Web verfügbaren Informationen auffindbar zu machen. Die Effektivität derartiger Suchmaschinen beruht auf algorithmischen Verfahren der Zuschreibung von Bedeutung. Unbekannt sind nicht nur die potenziell auffindbaren Informationen, sondern auch die Regeln der Attribuierung von Bedeutung sowie der Selektion von Information, weshalb die Suche als eine Black Box zu behandeln ist. Auch Websuchmaschinen dienen der Verarbeitung von Bedeutung. Im Vordergrund steht jedoch nicht der Sinn oder Gehalt von digitalen Informationen, sondern die Relevanz von Webdokumenten für den Suchenden. Dies haben Sergey Brin und Lawrence Page erkannt, die mit der Formulierung des PageRank-Algorithmus die Grundlage zeitgenössischer Websuchmaschinen geschaffen haben. Die praktische Leistungsfähigkeit des PageRank-Verfahrens haben Brin und Page mit ihrer Suchmaschine Google unter Beweis gestellt, welche seit mehr als einem Jahrzehnt eine Vormachtstellung auf dem Suchmaschinenmarkt innehat (Brin et al. 1998; Brin/Page 1998; Page et al. 1998).

Der PageRank-Algorithmus, wie er Ende der 1990er Jahre entwickelt und angewendet wurde, bewertet die Relevanz eines Webdokuments am Grad seiner Verlinkung, d.h. an der relativen Häufigkeit, mit der andere Webseiten auf dieses Dokument verweisen.[88] Ein Vorbild des PageRank ist die Berechnung des Einflussfaktors (*Impact Factor)* wissenschaftlicher Publikationen auf Basis der Zahl der Zitationen eines Artikels.[89] Brin und Page haben diese Idee auf das Web übertragen

---

richten ließen, deren Beantwortbarkeit bislang nicht getestet worden war« (Gugerli 2009: 72).

87 | Eine Unbekannte wird auf der Oberfläche allenfalls bei der Gestaltung von Benutzerschnittstellen eingeführt, welche den Nutzer auf bestimmte Suchmöglichkeiten festlegen, indem sie bestimmte Anfrageformen im Interface vorgeben. Siehe hierzu S. 296ff. im Kapitel »Phänomeno-Logik«.

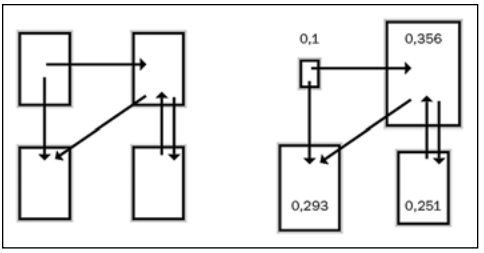
88 | Ein ähnliches Verfahren wie der PageRank-Algorithmus wurde ungefähr zur gleichen Zeit von Jon Kleinberg entwickelt, welches Hyerlink-Induced Topic Search, kurz HITS, genannt wird (Kleinberg 1999, 2000). Ein Vergleich zwischen dem HITS- und dem PageRank-Algorithmus findet sich in Rieder (2012).

89 | Page verweist in den beiden Patenten zum PageRank-Algorithmus explizit auf Eugene Garfield, den Erfinder des *Science Citation Index* (vgl. Page 2001, 2004). Die Wurzeln des PageRank in der Soziometrie und Bibliometrie werden von Mayer (2009) rekonstruiert.

und behandeln Hyperlinks analog zu Zitaten: »One can simply think of every link as being like an academic citation« (Page et al. 1998: 2). Die Qualität oder Relevanz einer Webseite wird nicht anhand inhaltlicher Kriterien bewertet, sondern aus der »hypertextual citation structure« (Page et al. 1998: 2) des gesamten Web abgeleitet, d.h. das gleichwertige Nebeneinander von Webseiten wird im Rückgriff auf die Linkstruktur gewichtet (siehe Abb. 16). Dadurch wird der ungeordneten Vielfalt von Dokumenten im WWW eine Ordnung gegeben, die sich in der Anordnung der Suchergebnisse widerspiegelt.

Webseiten, die in den Ergebnislisten weit vorn erscheinen, wird eine höhere Relevanz beigemessen als anderen weiter hinten aufgeführten Seiten. Diese Relevanzordnung ist für den Umgang mit Websuchmaschinen von zentraler Bedeutung. Für das Suchwort *Datenbank* findet Google beispielsweise circa 135 Millionen Ergebnisse, eine für menschliche Nutzer nicht zu überschauende Treffermenge.[90] Handhabbar werden die Ergebnisse für den Suchenden erst durch deren Anordnung in der Ergebnisliste anhand von Relevanzkriterien.

Abb. 16: Beispiel der Gewichtung von Webseiten mithilfe des PageRank-Verfahrens[91]



Websuchmaschinen operieren mit der Annahme, dass ihre Nutzer nicht an der Gesamtheit der Webseiten interessiert sind, die einem bestimmten Suchkriterium entsprechen, sondern nur an einer oder zumindest an relativ wenigen Seiten. Dementsprechend gibt Google ungeachtet der angezeigten Gesamtzahl von Ergebnissen dem Nutzer je Suchanfrage maximal 1000 Ergebnisse zurück. Die bereits erwähnte Suche nach dem Stichwort *Datenbank* fördert nur 905 tatsächliche Ergeb-

90 | Suche vom 3.2.2012.

91 | Der PageRank *PR* einer Webseite A ist rekursiv definiert und berechnet sich wie folgt: PR(A)=(1-d) + d[PR(T1)/C(T1) + ... + PR(Tn)/C(Tn)]. Parameter d ist ein Dämpfungsfaktor, der typischerweise auf 0,85 festgelegt wird. T1 bis Tn sind die Webseiten, die auf Seite A verweisen. C(Ti) ist die Summe der Links, die von einer Seite Ti auf andere Webseiten verweisen (vgl. Brin/Page 1998: 109f.). Der PageRank einer Webseite ist demzufolge nicht nur abhängig von der Zahl der Links, die auf diese zeigen, sondern auch von dem PageRank der Seiten, die diese Links beinhalten. Ein Link von einer höher bewerteten Webseite hat mehr Gewicht als ein Link von einer Seite mit niedrigem PageRank.

nisse zu Tage. Am Ende der Trefferliste findet sich der Hinweis: »*Um Ihnen nur die treffendsten Ergebnisse anzuzeigen, wurden einige Einträge ausgelassen, die den 905 bereits angezeigten Treffern sehr ähnlich sind*«.[92] Die genaue Zahl der angezeigten Suchergebnisse variiert mit jeder Anfrage leicht, die Botschaft ist aber stets dieselbe: Wer in den ersten tausend Ergebnissen nicht fündig wird, wird auch in den restlichen Ergebnissen nichts finden.

In dieser Hinsicht unterscheiden sich Websuchmaschinen grundlegend von relationalen Datenbanksystemen. Letztere zielen auf Vollständigkeit der Ergebnisse ab, wohingegen bei Suchmaschinen im WWW die Reduktion von Ergebnissen im Vordergrund steht. Die tabellarische Anordnung der Ergebnisse in relationalen Datenbanken orientiert sich, wie Krajewski herausgestellt hat, an »Evidenz und Übersicht« (Krajewski 2007: 47), beinhaltet aber keine Aussage über die Relevanz bestimmter Ergebnisse im Vergleich zu anderen. In der Ordnung der Suchergebnisse besteht demgegenüber ein zentrales Leistungsmerkmal von Websuchmaschinen.[93] Sie soll es Nutzern ermöglichen, aus der Vielzahl von Treffern effektiver solche auszuwählen, die für ihr aktuelles Informationsbedürfnis von Belang sind. Suchmaschinen erfüllen eine Orientierungsleistung und sind, wie der Informationswissenschaftler Michael Zimmer konstatiert hat, deshalb zum »center of gravity for people's everyday information seeking activities« (Zimmer 2008: 82) geworden.[94] Das Beispiel von Websuchmaschinen im Allgemeinen und Google im Besonderen wird an dieser Stelle herangezogen, um im Vergleich mit relationalen Datenbanksystemen eine weitere Dimension der computertechnischen Verarbeitung von Bedeutung darzulegen.[95] Datenstrukturen und Algorithmen stehen bei Suchma-

92 | Für die Beispielsuchanfrage kann der Nutzer nur auf ungefähr sieben Promille der Gesamtergebnismenge zugreifen. Vor diesem Hintergrund erscheint die Formulierung, dass »einige Einträge ausgelassen« wurden, als ein Euphemismus.

93 | Sofern man die von Websuchmaschinen bewirkte Aufmerksamkeitssteuerung der Nutzer prinzipiell kritisiert, verkennt man die Problematik, auf die diese Suchtechnologie eine Antwort ist.

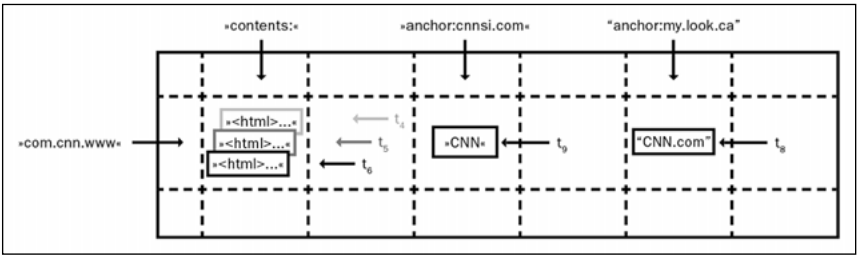
94 | Etwas Ähnliches konstatiert auch Röhle: »Suchmaschinen sind heute die zentralen Instanzen der technisch unterstützten Komplexitätsreduktion im Netz. Ihnen fällt die Aufgabe zu, Ordnung in der neuen Unübersichtlichkeit zu schaffen« (Röhle 2010: 11). Die große Bedeutung von Websuchmaschinen zeigt sich auch in dem wachsenden medien- und kulturwissenschaftlichen Interesse an diesem Gegenstandsbereich. Diskutiert wird vor allem die Frage der Macht von Suchmaschinen. Hierbei werden Websuchmaschinen mitunter als übermächtige Gatekeeper begriffen, die einseitig bestimmen, welche Informationen ihre Nutzer im WWW finden. Ein differenzierteres Bild der Macht von Suchmaschinen entwirft Theo Röhle in seiner Studie *Der Google-Komplex* in Rekurs auf Foucaults Machtkonzeption und die Akteur-Netzwerk-Theorie (vgl. Röhle 2010).

95 | Bei Websuchmaschinen steht anders als bei relationalen Datenbanken nicht der Gehalt von Informationen im Vordergrund, sondern deren Relevanz. Dieser

schinen in einem anderen Verhältnis zueinander als bei relationalen Datenbanken. Eine Konsequenz daraus sind die unterschiedlichen Oberfläche/Tiefe-Verhältnisse, die relationale Datenbanktechnologien einerseits und Suchmaschinen andererseits konstituieren.

Auch Websuchmaschinen werden technisch mittels Datenbankanwendungen realisiert, die auf der Modellierung von Information basieren. Dieses Informationsmodell spiegelt jedoch nicht die Komplexität und Heterogenität der im Web publizierten Informationen wider, sondern bildet die Topologie des Web in einer relativ einfachen Datenstruktur ab. Bei Google findet das *Bigtable*-Datenmodell Anwendung, welches die zweidimensionale Struktur von Tabellen (Zeilen, Spalten) um zwei weitere Dimensionen erweitert (vgl. Chang et al. 2006). Erstens wird eine Zeitdimension hinzugefügt, was zur Folge hat, dass Zellen mehrere Werte enthalten können, die durch einen Zeitstempel differenziert werden.[96] Zweitens werden Spalten in Spaltenfamilien gruppiert, wodurch die dynamische Erweiterung der Tabelle um weitere Spalten möglich wird.[97]

Abb. 17: Bigtable-Datenstruktur am Beispiel einer Webtable



Quelle: Chang et al. 2006: 206

Die Bigtable-Datenstruktur erläutern Chang et al. am Beispiel einer *Webtable*, womit sie auf ein zentrales Element der Google-Suchmaschine anspielen (vgl. Abb. 17). Obwohl das tatsächlich verwendete Informationsmodell ein Firmengeheimnis

Unterschied wird im Folgenden dargelegt, ist aber für den Vergleich dieser Suchtechnologien sekundär.

96 | Chang et al. beschreiben die erste Erweiterung von traditionellen zweidimensionalen Tabellen folgendermaßen: »A Bigtable is a sparse, distributed, persistent multidimensional sorted map. The map is indexed by a row key, column key, and a timestamp; each value in the map is an uninterpreted array of bytes. (row:string, column:string, time:int64) ® string« (Chang et al. 2006: 205).

97 | Durch die zweite Erweiterung werden Spaltenfamilien eingeführt, die der Gruppierung von Spalten dienen: »Column keys are grouped into sets called column families, which form the basic unit of access control. All data stored in a column family is usually of the same type (we compress data in the same column family together)« (Chang et al. 2006: 206).

ist, lässt sich an dem von Google-Mitarbeitern publizierten Beispiel der Webtable die Funktion verdeutlichen, die diese Datenbank für die Realisierung der Websuche hat.

Eine Zeile beinhaltet Informationen zu einer Webpage, welche durch ihre URL identifiziert wird. In der ersten Spalte der Bigtable wird der gesamte Inhalt der Webseite als HTML-Code gespeichert.[98] Die übrigen Spalten des Beispiels beinhalten die Links, welche von der jeweiligen Webpage aus auf andere Seiten verweisen. Jedes Linkziel, d.h. jede verlinkte URL, wird in einer eigenen Spalte gespeichert und die Tabellenzellen beinhalten den Ankertext des Links. Gruppiert sind diese Spalten zu der Spaltenfamilie »anchor«, welche die unterschiedlichen Spalten, die Links enthalten, zu einem Typ zusammenfasst und ihnen hierdurch eine Bedeutung zuweist. In einer solchen Datenstruktur lässt sich die ständig wandelnde Topologie des Web abbilden. Indem Google sämtliche von Webcrawlern, also Suchrobotern, gefundenen Webseiten in der Webtable speichert, wird das WWW in eine Datenbank transformiert und somit als Ganzes verwalt- und verarbeitbar.

Die Websuche von Google basiert maßgeblich auf der Versammlung des WWW in einer Datenbank. Für die Realisierung der Suchfunktionalität ist diese Übersetzung jedoch nur ein erster Schritt, sodass die Nutzer der Suchmaschine Google allenfalls mittelbar mit der Webtable-Datenbank interagieren. Entscheidend ist vielmehr, dass durch die Übersetzung der offenen Netzarchitektur des WWW in eine geschlossene Datenbankstruktur das Web als Ganzes handhabbar wird. Dies ist die Voraussetzung für die Berechnung des PageRank von Webseiten aus dem Grad ihrer Verlinkung mit anderen Webseiten, der, wie bereits dargelegt, als ein wichtiger Relevanz- bzw. Qualitätsindikator einer Webseite fungiert und die Anordnung von Suchergebnissen in der Ergebnisliste grundlegend beeinflusst. Das Informationsmodell, welches der Webtable zugrunde liegt, dient demzufolge nicht der semantischen Beschreibung der auf Webseiten enthaltenen Informationen, sondern bildet eine Metastruktur für die nachträgliche algorithmische Zuschreibung eines Relevanzwerts zu einer Webseite. Das Resultat der dem PageRank-Verfahren eingeschriebenen Hypothese ist, dass sich die Qualität einer Webseite in ihrer Linkpopularität, d.h. der Quantität ihrer Referenzen widerspiegelt, welche durch die Transformation des Web in eine Datenbank berechenbar wird.

Was die nutzerseitige Suche nach Informationen anbelangt, stehen bei Websuchmaschinen algorithmische Verfahren der Auswertung von Information und der Zuschreibung von Bedeutung im Vordergrund. Welche Ergebnisse in welcher Reihenfolge für Suchanfragen zurückgegeben werden, basiert demzufolge auf Algorithmen. Insofern lässt sich die Suche mit Websuchmaschinen als ein algorithmischer Selektionsprozess verstehen. Google bezieht eigenen Angaben zufolge derzeit nicht weniger als 200 Faktoren in die Bewertung von Webseiten ein, wobei die

98 | Neben HTML-Dateien indexiert Google eine Reihe weiterer *gebräuchlicher* Dateitypen, die im Web gespeichert und durch eine URL abgerufen werden können. Eine Liste findet sich in den *Google Webmaster Tools* (vgl. Google).

Algorithmen zudem häufigen Veränderungen unterliegen (vgl. Singhal 2012).[99] Wie genau sich die einzelnen Faktoren auf die Auswahl von Suchergebnissen auswirken ist geheim.[100] Infolgedessen ist nicht nur das Web als unsichtbarer Bestand von Informationen eine Black Box, sondern auch die Suche, welche in der unsichtbaren Tiefe des Computers, hier der Serverfarm von Google, situiert ist. Eine Konsequenz daraus ist, dass es für die Nutzer von Websuchmaschinen prinzipiell unmöglich ist zu entscheiden, ob Veränderungen in den Ergebnissen auf Änderungen im Web oder auf Revisionen der Suchalgorithmen zurückzuführen sind (vgl. Stalder/Mayer 2009: 126).

Zwischen den Betreibern und Nutzern von Suchmaschinen besteht aufgrund der strategischen Geheimhaltung der Suchalgorithmen eine Machtasymmetrie.[101] Dies ist insofern problematisch, als die algorithmisch hergestellte Relevanzordnung kein objektives Bild der »wirklichen« Relevanz von Informationen ist. Die Algorithmen von Websuchmaschinen konstruieren vielmehr Relevanz. Im Anschluss an die bereits diskutierte Beschreibung des konzeptuellen Datenbankschemas als eine Weise der symbolischen Formung bzw. vorlogischen Strukturierung von Wirklichkeit kann diese algorithmische Herstellung einer Relevanzordnung als alternative Weise der symbolischen Formung *in*, *mit* und *durch* Computer verstanden werden.[102] Während das konzeptuelle Schema ein Modell der Welt vorgibt, beruhen Suchmaschinen jedoch auf Verfahren der nachträglichen Ordnung von Informationen durch Algorithmen. In diese Algorithmen sind Hypothesen darüber eingeschrieben, was für Nutzer relevante Informationen sind und wie sich diese computertechnisch ausfindig machen lassen. Die Relevanz einer Webseite wird von Websuchmaschinen daher nicht gemessen, wie sich etwa die Länge einer Strecke

99 | Der Marketing-Softwareanbieter Moz unterhält eine Liste mit bekannt gewordenen Veränderungen am Suchalgorithmus von Google (vgl. Moz).

100 | Dies trifft nicht nur auf Google, sondern auf Suchmaschinenbetreiber im Allgemeinen zu. Welche Faktoren auf welche Weise in die Bewertung von Webseiten eingehen wird gemeinhin geheim gehalten.

101 | Websuchmaschinen im Allgemeinen und Google im Besonderen lassen sich zwar als Gravitationszentrum der Orientierung im Web begreifen, sie bilden aber kein absolutes Machtzentrum für die Ordnung von Information und Wissen. Die Macht von Suchmaschinen entfaltet sich Röhle zufolge in einem komplexen Akteur-Netzwerk von Suchmaschinenbetreibern, Inhalteanbietern, Nutzern und Suchmaschinenoptimierern mit komplexen relationalen Machtverhältnissen, die »durch Verhandlungen, Assoziationen und wechselnde Dynamiken der Reversibilität gekennzeichnet« (Röhle 2010: 229) sind. Obwohl der Beurteilung von Röhle prinzipiell zuzustimmen ist, gilt es festzuhalten, dass den Betreibern von Suchmaschinen in dem vielschichtigen Machtgefüge eine Sonderstellung zukommt, da sie einseitig die Bewertungsregeln verändern können, nach denen die Algorithmen der Suchmaschinen Inhalte im Web ranken.

102 | Siehe S. 236f.

mit einem Maßband messen lässt, sondern nach den Regeln des Algorithmus Webseiten zugeordnet. Infolgedessen haben Suchmaschinenbetreiber durch die Veränderung der Algorithmen einen erheblichen Einfluss darauf, was ihren Nutzern als relevante Information präsentiert wird und welche Informationen diese zur Kenntnis nehmen.

Ein Beispiel hierfür ist der aktuelle Trend zur Personalisierung von Suchergebnissen, die der Internetaktivist Eli Pariser (2011) als *Filter Bubble* bezeichnet und kritisiert hat. Das Versprechen der Personalisierung ist es, den Nutzern auf Grundlage ihrer Nutzungsgewohnheiten bessere, d.h. individuell relevantere Ergebnisse zu liefern. Hierin besteht nach Ansicht von Pariser jedoch auch eine Gefahr, da die Nutzer zunehmend in einer *Filter Bubble* gefangen werden, in der sie nur das finden, was ihren eigenen Überzeugungen, Ansichten, Einstellungen und Interessen entspricht. Die für die Nutzer unsichtbare Personalisierung verdeckt die Vielfalt an Informationen und Wissen sowie die Heterogenität von Positionen und verzerrt somit ihre Wahrnehmung: »[T]he filter bubble distorts our perception of what's important, true, and real« (Pariser 2011: 20).

In Anerkennung der Gestaltungsmacht von Suchmaschinenbetreiber haben Lucas Introna und Helen Nissenbaum bereits 2000 in dem vielbeachteten Artikel *Shaping the Web: Why the Politics of Search Engines Matters* die Offenlegung von Suchalgorithmen gefordert: »As a first step we would demand full and truthful disclosure of the underlying rules (or algorithms) governing indexing, searching, and prioritizing, stated in a way that is meaningful to the majority of Web users« (Introna/Nissenbaum 2000: 181). Dies ist nach Einschätzung von Bernhard Rieder die am häufigsten geäußerte Forderung im aktuellen Suchmaschinendiskurs (vgl. Rieder 2009: 152). Doch das Öffnen der Black Box würde die Suche im WWW nicht demokratischer, neutraler oder besser machen. Im Gegenteil, die Bekanntgabe der in Suchalgorithmen eingeschriebenen Bewertungsmechanismen würde die Machtasymmetrie nicht auflösen, sondern auf eine andere Ebene verschieben. Denn die Ergebnisse von Suchmaschinen lassen sich nicht nur von »innen« heraus durch die Anpassung von Algorithmen verändern. Sie können auch von »außen« durch die gezielte Gestaltung von Webseiten und durch die Beeinflussung der Webtopologie manipuliert werden. Für diese Art der äußeren Einflussnahme auf die Ergebnislisten von Suchmaschinen hat sich der Begriff *Suchmaschinenoptimierung* (*Search Engine Optimization*, kurz: *SEO*) durchgesetzt.[103] Mit dem Wissen um die Be-

103 | Seitens der Suchmaschinenbetreiber wurde die Unterscheidung von weißen und schwarzen Optimierungspraktiken eingeführt, welche jedoch ebenso ökonomisch motiviert sind. Weiße Suchmaschinenoptimierung kann zur Verbesserung der Suche von Websuchmaschinen beitragen, wenn z.B. Webseitenanbieter ihre Seiten technisch so aufbereiten, dass sie von Suchmaschinen besser verarbeitet werden können. Diese Optimierungspraxis ist von Suchmaschinenbetreibern erwünscht und wird unterstützt. Demgegenüber dient die schwarze Suchmaschinenoptimierung der gezielten Manipulation von Suchergebnissen. Aufgrund dessen enthalten die

wertungsprinzipien von Suchmaschinen ließe sich die Suchmaschinenoptimierung exakt operationalisieren und damit algorithmisieren, was zur Konsequenz hätte, dass Nutzer auf Suchanfragen fortan nur noch das finden, was für die Suchmaschine besonders gut durch SEO-Algorithmen optimiert wurde. An die Stelle des einen als Übermächtig empfundenen Suchalgorithmus träte eine Vielzahl miteinander konkurrierender SEO-Algorithmen, die an den Informationsbedürfnissen der Nutzer vorbei operieren, denn die Aufmerksamkeitslenkung der Nutzer ist für diese kein Hilfsmittel für den übergeordneten Zweck der Suche, sondern ein allein ökonomisch motiviertes Ziel. Der Zugewinn an Transparenz hätte zur Konsequenz, dass das Finden von Ressourcen für die Nutzer in der Suchpraxis schwieriger, wenn nicht sogar unmöglich wird.[104] Infolgedessen dient die Geheimhaltung von Suchalgorithmen nicht nur der Sicherung von Wettbewerbsvorteilen einzelner Suchmaschinenbetreiber, sondern auch dem weiteren Funktionieren der Websuchmaschine. Diesbezüglich stellte Eric Enge auf dem Blog *Search Engine Watch* treffend fest: »[L]inks were a better quality signal when the world didn't know that they were a signal. But, those days are gone« (Enge 2012).[105] Mittlerweile ist

---

Ergebnislisten von Suchmaschinen bei bestimmten Themen für die Nutzer nur sehr wenige brauchbare Resultate. Derartige Manipulationen versuchen Suchmaschinenbetreiber durch die kontinuierliche Anpassung ihrer Bewertungsalgorithmen zu unterbinden.

104 | Eine Demokratisierung der Suche lässt sich demzufolge nicht durch die Offenlegung der Suchalgorithmen erreichen, sondern setzt, wie Rieder dargelegt hat, »ein klares begriffliches Verständnis der Technologie, ebenso wie ein Überdenken unseres Verständnisses von Demokratie« (Rieder 2009: 168) voraus. Diese kreuzen/überlagern sich im Design von Suchmaschinen, das derzeit vom Prinzip der Nutzerorientierung geprägt ist. Demgegenüber macht sich Rieder für ein »gesellschaftlich orientiertes Design« (Rieder 2009: 157) von Suchmaschinen stark, welches auf der politischen Ebene durch Regulierungsmaßnahmen und die Schaffung von Anreizen sowie auf der medienpraktischen Ebene durch die Förderung experimenteller Suchtechnologien auf der Basis existierender Suchmaschineninfrastrukturen durchgesetzt werden könne (vgl. Rieder 2009: 162ff.). Wie Röhle zu bedenken gibt, sind die Erfolgsaussichten eines solchen Ansatzes gegenwärtig als relativ gering einzuschätzen: »Regulierung, Aufklärung oder die Forderung technischer Alternativen – bisher sind noch so gut wie alle Ansätze an der Bequemlichkeit der Nutzer gescheitert« (Röhle 2010: 234).

105 | Der Indikator, auf dem einst der Erfolg der Suchmaschine Google hauptsächlich gründete, ist durch die Bedeutung, welche Links für die Relevanzzuschreibung beigemessen wurde, unzuverlässig geworden und hat infolgedessen an Relevanz verloren (vgl. Wall 2011). Dennoch bildet die Linkpopularität noch immer einen Faktor bei der Bewertung von Webseiten durch Suchmaschinen. Es ist jedoch umstritten, wie stark der Grad der Verlinkung die algorithmische Auswahl der Suchergebnisse beeinflusst.

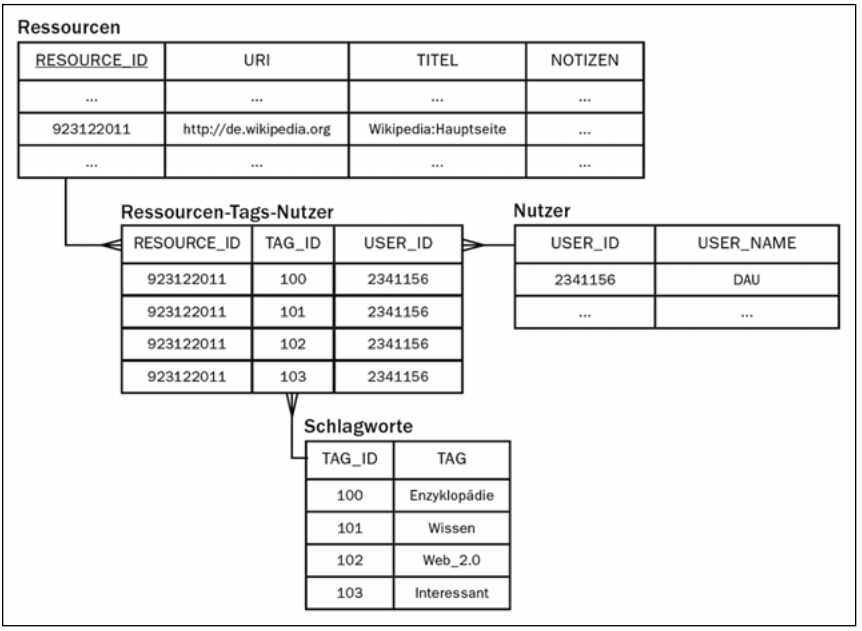
unklar, ob und in welchem Maß Google die Linktopologie des Web noch immer zur Relevanzbewertung von Webseiten heranzieht. Dementsprechend mag der PageRank-Algorithmus heute überholt sein. Das an diesem Beispiel erläuterte Prinzip der algorithmischen Herstellung von computer-lesbarer Signifikanz ist es jedoch nicht.

An Websuchmaschinen zeigt sich eine alternative Weise der operativen Verschaltung von Informationsmodellen und Algorithmen. Googles Webtable dient als Metastruktur der algorithmischen Zuschreibung von Bedeutung im Sinn von Relevanz. Zwar liegt auch der Webtable ein Informationsmodell zugrunde, welches die in der Datenbank gespeicherten Informationen als bestimmte Informationen adressierbar und damit verarbeitbar macht. Im Kontext von Websuchmaschinen ist dies jedoch nur die Voraussetzung für die anschließende algorithmische Auswertung der Informationen zum Zweck ihrer Relevanzbewertung. Demgegenüber wurde im vorangegangenen Unterkapitel zum relationalen Paradigma dargelegt, dass relationale Datenbanken digitale Informationen gemäß einem vordefinierten Informationsmodell verwalten. Auch wenn dies grundsätzlich zutreffend ist, gilt es die Gegenüberstellung von Datenbanken einerseits und Websuchmaschinen andererseits mit einer Einschränkung zu versehen. Zwar operieren relationale Datenbanken stets auf der Basis eines Informationsmodells. Dieses kann jedoch ähnlich wie bei Websuchmaschinen als Metastruktur für die algorithmische Zuschreibung von Bedeutung fungieren.

Welche Rolle Datenbanken in der medialen Praxis spielen, lässt sich demzufolge nicht allein an dem Datenmodell ablesen, auf dem eine Datenbank beruht. Vielmehr ist im Kontext partikularer Informationssysteme, z.B. Suchmaschinen, zu beobachten, wie Datenbanken darin operativ werden. Auch relationale Datenbanken können in Informationssystemen derart gebraucht werden, dass die in ihnen gespeicherten Informationen auf der Ebene des Informationsmodells der Datenbank, dem konzeptuellen Schema, zwar eine Bedeutung haben, diese aber von den Bedeutungen verschieden ist, die auf der Ebene des Informationssystems von Interesse sind und prozessiert werden. Ein Beispiel hierfür sind Social Tagging-Systeme, welche die nutzerseitige Beschreibung digitaler Objekte mit Schlagworten erlauben, ohne dass sich die Bedeutung der Schlagworte im Informationsmodell der Datenbank widerspiegelt. Hierbei bildet die relationale Datenbank nicht das Informationssystem, sondern ist Teil einer »*Infrastruktur der Bedeutung*« (Weinberger 2008: 205). Da im Rahmen eines Tagging-Systems der Sinn der von den Nutzern vergebenen Tags gegenüber der Datenbank nicht explizit gemacht wird, bedarf es algorithmischer Auswertungsverfahren, um den Tags eine Bedeutung beizumessen. Das bekannteste Beispiel hierfür sind Tag Clouds, in denen vergebene Schlagworte entsprechend der Häufigkeit ihrer Verwendung angeordnet bzw. dargestellt werden. Hieraus lassen sich Tendenzen in der Beschreibung digitaler Medienobjekte ablesen, thematische Zusammenhänge erschließen oder Interessen

von Nutzern verfolgen.[106] Entscheidend ist, dass die Unterschiede, die in einer Tag Cloud zur Darstellung kommen, nicht im konzeptuellen Schema der Datenbank modelliert sind, sondern das Ergebnis der regelgeleiteten Auswertung von Datenbankinformationen. Das Informationsmodell, welches Datenbanken in Tagging-Systemen zugrunde liegt, ist relativ einfach, wie der Informationsarchitekt Gene Smith in seiner praxisorientierten Einführung *Tagging: People-Powered Metadata for the Social Web* zeigt: Ressourcen, Tags und Nutzer werden jeweils in einer eigenen Relation modelliert, wodurch diese als Entitäten adressierbar werden (Abb. 18). Welche Schlagworte ein Nutzer für eine Ressource vergeben hat wird in der Relation *Nutzer-Ressourcen-Tags* gespeichert. Diese Relation enthält als Informationen die Schlüssel der drei anderen Relationen und stellt hierdurch eine Beziehung zwischen einem Nutzer einer Ressource und einem Tag her.

Abb. 18: Konzeptuelles Schema für ein kollaboratives Tagging-System



Quelle: Smith 2008: 141

Versieht ein Nutzer eine Ressource mit unterschiedlichen Tags, dann werden diese, wie im obigen Beispiel erläutert, als unterschiedliche Tupel gespeichert. Dieses Informationsmodell ermöglicht die Identifikation von Ressourcen, Nutzern sowie

106 | Das Erkenntnispotenzial einer Tag Cloud hängt davon ab welche Schlagworte in dieser zusammengezogen werden und was demzufolge den Referenzrahmen bildet. Für den Kontext der in diesem Kapitel verfolgten Argumentation ist diese Frage jedoch nebensächlich.

Schlagworten und erlaubt es dadurch, diese in ein Nähe- und Distanzverhältnis zueinander zu stellen. So können Nutzern auf Grundlage der von ihnen annotierten Ressourcen und vergebenen Tags nicht nur neue, ähnliche Ressourcen vorgeschlagen werden, sondern auch andere Nutzer, die ähnliche Interessen haben. Diese Ähnlichkeiten sind jedoch nicht in der Datenbank gespeichert, sondern können aus den Datenbankinformationen abgeleitet werden.[107]

Relationale Datenbanken, Websuchmaschinen und Social Tagging-Systeme lassen sich als Informationssysteme begreifen, die Bedeutung verarbeiten. Während in relationalen Datenbanksystemen der Gehalt von Information im Mittelpunkt steht, rücken Websuchmaschinen wie Google den Relevanzaspekt von Information ins Zentrum. Das Social Tagging basiert wiederum auf einer Sinnzuschreibung durch die Nutzer in Form von Schlagworten, die unterschiedlichen Auswertungsverfahren zugänglich sind. Tag Clouds wurden als Beispiel eines solchen Auswertungsrespektive Präsentationsmodus vorgestellt. Die Darstellung der Schlagworte entsprechend ihrer Häufigkeitsverteilung übersetzt die nutzerseitigen Beschreibungen in ein Relevanzdiagramm, welches eine Gewichtung zwischen den von den Nutzern vergebenen Schlagworten anschaulich macht. Hieran wird deutlich, dass die beiden Seiten von Bedeutung – Gehalt und Relevanz – für den computertechnischen Umgang mit Informationssammlungen nicht nur gleichermaßen wichtig sind, sondern auch, dass diese im Kontext spezifischer Informationssysteme häufig miteinander verschaltet sind. Zugleich erweist sich die Frage nach dem Unterschied zwischen Technologien des Sinns und Technologien der Relevanz in der digitalen Medienkultur als ungemein wichtig.[108] Dies wird im Folgenden am Beispiel des Semantic Web dargelegt.

Semantic Web: Herausforderungen einer Vision

An Websuchmaschinen wird mitunter kritisiert, dass diese weder in der Lage seien, den Sinn von Internetinformationen zu verstehen, noch den von Suchanfragen. Die

107 | Jenseits der beschriebenen Möglichkeiten zur Auswertung der gespeicherten Informationen sind in das konzeptuelle Schema jedoch auch Beschränkungen eingeschrieben, die sich direkt auf das Informationssystem auswirken. Das von Smith vorgeschlagenen Modell geht implizit davon aus, dass jede Ressource nur einen Titel hat. Demzufolge ist es im Rahmen eines Tagging-Systems, das auf einer solchen Datenbank beruht, nicht möglich, dass die Nutzer ihren Ressourcen einen eigenen Titel geben. Um dies zu ermöglichen, müsste das Schema um eine Relation erweitert werden, die neben dem gewählten Titel die Identifikationsnummer des Nutzer sowie der Ressource als Attribute enthält.

108 | Bedeutsam ist diese Frage nicht nur hinsichtlich der Beschreibung konkreter medialer Praxen, sondern vor allen auch auf der diskursiven Ebene, auf der die Vor- und Nachteile bestehender Informationssysteme ebenso verhandelt werden wie Visionen für die Zukunft digitaler Medientechnologien.

bereitgestellten Suchfunktionalitäten basieren auf relativ »einfachen statistischen Verfahren, die keinerlei Verständnis der Inhalte durch die Suchmaschine voraussetzen« (Dopichaj 2009: 113). Websuchmaschinen sind, so der Kern der Kritik, keine semantischen Technologien, die Informationen hinsichtlich ihres Gehalts differenzieren. Ob sich die Zeichenfolge *Golf* auf ein Automodell, eine Meeresbucht oder eine Sportart bezieht, ist für rein schlagwortbasierte Suchmaschinen gleichgültig. In Anbetracht der großen Menge an potenziell zugänglichen Informationen kann sich dies jedoch als problematisch erweisen. Seit Ende der 1990er Jahre wird daher verstärkt an Computertechnologien gearbeitet, die die automatische Verarbeitung von Sinn im WWW ermöglichen. Paradigmatisch dafür ist die Idee des Semantic Web, die von Tim Berners-Lee formuliert und als Vision des künftigen Web popularisiert wurde (vgl. Berners-Lee/Fischetti 1999; Berners-Lee et al. 2001).[109] Das Ziel des Semantic Web ist es, im Internet publizierte Informationen technisch nicht nur auf dem Niveau von Realität, sondern als Information *über* Realität verarbeitbar zu machen: »To date, the Web has developed most rapidly as a medium of documents for people rather than for data and information that can be processed automatically. The Semantic Web aims to make up for this« (Berners-Lee et al. 2001: 37). Konnte der Sinn von Internetinformationen bis dahin nur von menschlichen Akteuren erschlossen werden, soll dieser mithilfe des Semantic Web fortan auch von Computern verarbeitet werden können. Die Grundlage bildet die Explikation der Bedeutung von Information in Form von Metadaten, wodurch der implizite Sinn in explizite Informationen übersetzt wird. Um dies zu realisieren, wurden vom World Wide Web Consortium (W3C) eine Reihe von abstrakten Datenmodellen entwickelt, die es erlauben, unterschiedliche Sinndimensionen formal zu spezifizieren.[110]

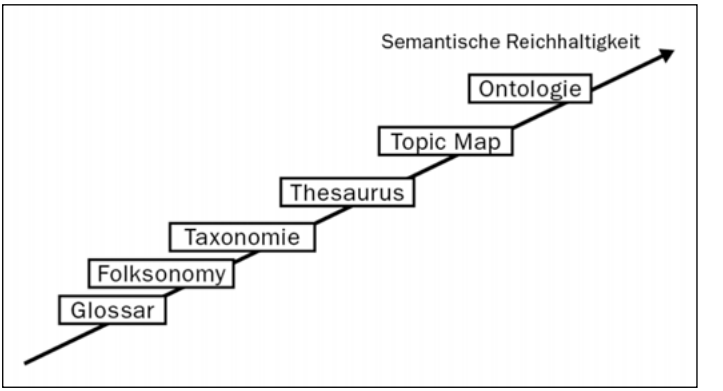
---

109 | Wie Blumauer und Pellegrini in ihrer Darlegung zentraler Begriffe und Unterscheidungen des Semantic Web feststellen, sind die Begriffe *semantische Technologie* und *Semantic Web* nicht miteinander gleichzusetzen. Beim Semantic Web handelt es sich um einen bestimmten Typus semantischer Technologien. Insofern dient die Bezeichnung semantische Technologien als Oberbegriff für Technologien des Sinns: »Während das Semantic Web im Kern auf Standards zur Beschreibung von Prozessen, Dokumenten und Inhalten sowie entsprechenden Metadaten – vorwiegend vom W3C vorgeschlagen – aufsetzt, und damit einen Entwurf für das Internet der nächsten Generation darstellt, adressieren semantische Technologien Herausforderungen zur Bewältigung komplexer Arbeitsprozesse, Informationsmengen bzw. Retrieval-Prozessen und Vernetzungs- oder Integrationsaktivitäten, die nicht nur im Internet, sondern auch innerhalb von Organisationsgrenzen in Angriff genommen werden« (Blumauer/Pellegrini 2006: 20).

110 | Ein Überblick über die Aktivitäten des W3C zum Semantic Web findet sich auf der Webseite des Konsortiums unter www.w3.org/standards/semanticweb/ (zuletzt aufgerufen am 14.09.2013).

Als Basistechnologien sind vor allem das Ressource Description Framework (RDF), das RDF-Schema (RDFS) und die Web Ontology Language (OWL) zu betrachten, welche einerseits die formale Deklaration des Sinns von Konzepten in einem Schema erlauben und andererseits die Beschreibung von Ressourcen gemäß der im Schema deklarierten Sinndimensionen.[111] Die vom W3C vorgeschlagenen Beschreibungs- und Sprachkonstrukte unterscheiden sich hinsichtlich ihrer Ausdrucksdrucksmächtigkeit, was zur Folge hat, dass sie die Formalisierung von Sinnzusammenhängen auf unterschiedlich komplexe Weise ermöglichen. Andreas Blumauer und Tassilo Pellegrini sprechen in diesem Zusammenhang von einer *semantischen Treppe*, welche von semantisch armen Glossaren bis hin zu semantisch reichhaltigen Ontologien reicht (vgl. Abb. 19). Auf der obersten Ebene von Ontologien soll es Computern mithilfe von festgelegten Regeln möglich sein, automatische Schlussfolgerungen anzustellen.

Abb. 19: Semantische Treppe



Quelle: Blumauer/Pellegrini 2006: 16

Ohne an dieser Stelle eine umfassende Analyse des Semantic Web zu leisten, werden im Folgenden zwei medienkulturwissenschaftlich relevante Aspekte beleuchtet. Eine erste aufschlussreiche Beobachtung erlaubt der Diskurs über das Semantic Web. Als Vision und Entwicklungsaufgabe hat es auf der einen Seite positive Resonanz gefunden. Hiervon zeugen beispielsweise das 2006 vom Bundesministerium für Wirtschaft und Technologie initiierte *THESEUS*-Forschungspro-

111 | In dieser Hinsicht ist die formale Explikation von Bedeutung im Semantic Web der Modellierung von Informationen in Datenbanken nicht unähnlich. Das konzeptuelle Schema ist mit den in RDFS und OWL formalisierten semantischen Modellen vergleichbar und die in der Datenbank gespeicherten Informationen mit den um semantische Metadaten erweiterten Ressourcen, wie Berners-Lee in einer Gegenüberstellung relationaler Datenbanken mit RDF herausgestellt hat (vgl. Berners-Lee 1998).

gramm, welches die Entwicklung semantischer Technologien insbesondere auch im Bereich des Semantic Web zum Ziel hat, sowie das von der Europäischen Union geförderte *LOD2*-Projekt, dessen Motto »Creating Knowledge out of Interlinked Data« (LOD2 2010) lautet.[112] Auf der anderen Seite wurde das Semantic Web von Kultur- und Medienwissenschaftlern einer zum Teil heftigen Kritik unterzogen (vgl. Shirky 2003, 2005; Cramer 2007; Weinberger 2008). Die vorgebrachten Einwände richten sich weniger auf konkrete Anwendungsszenarien, sondern beziehen sich grundlegender auf die Vision des Semantic Web im Allgemeinen. Mit dem Hinweis auf die soziale Konstruktion von Bedeutung und die Interpretationsoffenheit von Information wird angezweifelt, dass der Sinn von Internetinformationen durch Metainformationen eindeutig expliziert und so für die computertechnische Verarbeitung aufgeschlossen werden kann. Diese Kritik ist jedoch nicht unproblematisch, da jenseits des öffentlichkeitswirksamen Plädoyers für ein Semantic Web durch Tim Berners-Lee weithin umstritten ist, was das semantische Netz eigentlich genau sein soll. In dem 2008 erschienenen *Semantic Web Primer* verzeichnen die Autoren zwei teilweise gegenläufige Vorstellungen bzw. Entwicklungsrichtungen. Einerseits wird das Semantic Web als Datenweb verstanden und damit als spezifischer Teilbereich des WWW, wohingegen es andererseits als Versuch betrachtet wird, das Web insgesamt durch semantische Technologien zu verbessern, wie z.B. durch die Entwicklung von semantischen Suchmaschinen (vgl. Antoniou/van Harmelen 2008: 245f.).

In Anbetracht der Heterogenität des Semantic Web läuft die medien- und kulturwissenschaftliche Kritik an diesem Web teilweise ins Leere. Die formulierten Bedenken richten sich gegen eine abstrakte und vielgestaltige Utopie, die erst langsam konkrete Konturen annimmt (vgl. Pellegrini 2008). Infolgedessen kann jeder prinzipielle Einwand mit dem berechtigten, aber auch problematischen Hinweis entkräftet werden, dass die Kritik auf einem falschen Bild des Semantic Web beruhe (vgl. Van Dijck 2003; Hendler 2008; Antoniou/van Harmelen 2008: 247). Dennoch wäre es falsch, die von David Weinberger (2002, 2008), Clay Shirky (2003, 2005) und Florian Cramer (2007) gegenüber dem Semantic Web formulierten Bedenken leichtfertig zu ignorieren. Auch wenn ihre Argumentationen angreifbar sind, insofern sie sich auf das Semantic Web im Allgemeinen beziehen, kann ihre Kritik als Warnung vor einer Entgrenzung des Semantic Web verstanden werden. Drei eng miteinander zusammenhängende Dimensionen einer solchen Entgrenzung lassen sich unterscheiden:[113] die Universalisierung von Ontologien hin zu einem eindeutigen und allgemein gültigen Weltmodell; der Versuch, den Sinn aller digitalen

112 | Nähere Informationen zu den Zielen des Projekts und deren derzeitige Umsetzung finden sich auf der Webseite lod2.eu.

113 | Es wird an dieser Stelle nicht behauptet, dass die Befürworter des Semantic Web diese als Entgrenzungen charakterisierten Ziele verfolgen und an deren Umsetzbarkeit glauben. Vielmehr soll anerkannt werden, dass die abstrakte Idee eines semantischen Web die Gefahr birgt, dementsprechend verstanden zu werden.

Informationen durch semantische Metainformationen formal zu explizieren;[114] die Nachbildung des menschlichen Verständnis- und Interpretationsvermögens im Computer.[115]

An dieser Stelle soll exemplarisch nur auf den ersten Punkt näher eingegangen werden. Nach Ansicht von Cramer, Shirky und Weinberger beruht die Idee des Semantic Web auf der Utopie eines eindeutigen und allgemein gültigen Weltmodells, welches in einer Ontologie formal spezifiziert werden könne. Hiergegen sprechen sich diese Kritiker unisono aus.

Das Streben nach einer universellen Klassifikation kommt Cramer zufolge einem »technocratic neo-scholasticism«(Cramer 2007) gleich, der auf der gleichermaßen naiven und gefährlichen Annahme beruht, die Welt könne von einem einzigen universell gültigen Standpunkt aus beschrieben werden. Weinberger erkennt darin den »Traum des Rationalismus« (Weinberger 2008: 230), der auf dem Glauben beruht, ein »umfassendes System aufbauen und jede Mehrdeutigkeit aus ihm verdrängen« (Weinberger 2008: 230) zu können. Nach Ansicht von Shirky ist die Arbeit an einer universell gültigen und umfassenden Ontologie zum Scheitern verurteilt, da Metadaten stets eine Weltanschauung zum Ausdruck bringen: »Any attempt at a global ontology is doomed to fail, because meta-data describes a worldview« (Shirky 2003). Dass das Semantic Web eine universelle Ontologie voraussetzt bzw. auf die Entwicklung einer solchen Ontologie zuläuft, ist, wie Antonio und van Harmelen darlegen, jedoch ein weit verbreiteter Trugschluss, da die technische Entwicklung des Semantic Web von einem Pluralismus unterschiedlicher Ontologien mit begrenzter Reichweite gekennzeichnet ist, und zwar sowohl hinsichtlich des Gültigkeitsbereichs als auch hinsichtlich ihrer semantischen Reichhaltigkeit (vgl. Antoniou/van Harmelen 2008: 247).[116]

Dieser tatsächlichen Begrenztheit steht eine in den öffentlichen Darstellungen verbreitete Revolutionsrhetorik gegenüber, die im Semantic Web die nächste Evolutionsstufe der Menschheit erblickt: »[I]f properly designed, the Semantic Web can assist the *evolution* of human knowledge as a whole« (Berners-Lee et al. 2001: 43). Gegen diese Entgrenzung richten sich die zitierten Kritiker des Semantic

114 | In *Metacrap* weist Cory Docotrow (2001) auf die komplexen sozialen und technischen Prozesse der Erzeugung von Metadaten hin und kritisiert damit den unbedingten Glauben an die eindeutige semantische Annotierbarkeit von Informationen.

115 | Shirky zufolge steht das Semantic Web in der Tradition der *Künstlichen Intelligenz*-Forschung, deren Versprechen jedoch uneingelöst blieben (Shirky 2003).

116 | Antoniou und van Harmelen nennen vier weit verbreitete Trugschlüsse über das Semantic Web: 1. »The Semantic Web tries to enforce meaning from the top«; 2. »The Semantic Web requires everybody to subscribe to a single predefined meaning for the terms they use«; 3. »The Semantic Web requires users to understand the complicated details of formalized knowledge representation«; 4. »The Semantic Web requires the manual markup of all existing Web pages, an impossible task« (Antoniou/van Harmelen 2008: 247f.).

Web, wenn sie auf die prinzipielle Beschränktheit von Klassifikationssystemen und algorithmischen Auswertungsverfahren hinweisen, sowie wenn sie Zweifel daran anmelden, dass das gesamte WWW mit semantischen Metadaten erweitert werden kann. Produktiv gewendet kann diese Kritik als Hinweis für den künftigen Einsatz von Semantic Web-Technologien in konkreten medialen Praxen verstanden werden. Hierbei gilt es der Tendenz zur Entgrenzung des Semantic Web entgegenzuwirken, indem die durchaus noch näher zu bestimmenden Grenzen dieser Technologien des Sinns anerkannt werden und in die Gestaltung der auf diesen Technologien beruhenden Informationssysteme eingehen. Ein mögliches Ziel könnte es sein, die Kontingenz von Glossaren, Taxonomien, Thesauri und Ontologien in der Tiefe des Informationssystems nicht zu verbergen, sondern an den Benutzeroberflächen erfahrbar zu machen, sodass diese Erfahrung zum integralen Bestandteil der Interaktion mit semantisch annotierten Informationen wird. Dies führt zu der zweiten medienkulturtheoretisch bedeutsamen Beobachtung, welche die Probleme betrifft, die sich bei der praktischen Umsetzung des Semantic Web abzeichnen.

Neben den genannten prinzipiellen Bedenken wird häufig der pragmatische Einwand geäußert, dass es sich bei dem Semantic Web vor allem um einen theoretischen Diskurs handelt, der in der Medienpraxis jedoch weithin wirkungslos geblieben ist. »Man findet«, so stellt Weinberger kritisch fest, »[...] kaum Anwendungen des Semantic Web im großen Stil, die über die Forschungsphase hinausgehen würden« (Weinberger 2008: 233). Eine Ursache hierfür ist, dass RDF, RDFS und OWL nicht in erster Linie auf Anwendungen abzielen, sondern die semantische Auszeichnungen von Informationen ermöglichen, auf die in unterschiedlichen Anwendungskontexten zurückgegriffen werden können soll.[117] Die Publikation von semantisch annotierten Informationen im Internet und die Vernetzung dieser Informationen mithilfe von Links wird jedoch als Voraussetzung für die Implementierung von Semantic Web-Anwendungen betrachtet. Dementsprechend hat Tim Berners-Lee 2006 unter dem Titel *Linked Data* vier Regeln für die Publikation von strukturierten Daten im Internet vorgeschlagen (Berners-Lee 2006):

»1. Use URIs[118] as names for things
2. Use HTTP URIs so that people can look up those names.
3. When someone looks up a URI, provide useful information, using the standards (RDF*, SPARQL)
4. Include links to other URIs. [sic!] so that they can discover more things.«

117 | Das Semantic Web beruht auf der Idee der Entkopplung der im Web verfügbaren Informationen von ihrem Gebrauch im Rahmen bestimmter Anwendungen, d.h. unter anderem von ihrer Präsentation auf Webseiten. Die Parallele zum Problem der Datenunabhängigkeit, welches die Entwicklung von Datenbankmanagementsystemen angeleitet hat, ist unverkennbar.

118 | URI steht für *Uniform Resource Identifier* und dient der eindeutigen Identifikation von Ressourcen.

Das Ziel dieser *Best Practice*-Richtlinien ist es, die isolierten Inseln strukturierter Daten im Internet zu einem »Global Data Space« (Heath/Bizer 2011: 4) zusammenzuführen, den Berners-Lee als »web of data« bezeichnet und in dem der »unexpected re-use of information« (Berners-Lee 2006) möglich werde. Im Anschluss an die Formulierung der Linked Data-Regeln wurde im Januar 2007 vom W3C das Linking Open Data-Gemeinschaftsprojekt initiiert, welches die Bemühungen im Bereich von Linked Data koordiniert und einen Überblick über die derzeit verfügbaren Datenbestände gibt. Dies hat dazu geführt, dass die Zahl der frei zugänglichen Linked Data-Datenbestände von anfänglich zwölf auf 570 angestiegen ist (W3C 2014). Durch diese Entwicklung ist die oben genannte Voraussetzung für die Entwicklung von Endnutzeranwendungen im Semantic Web gegeben. Jedoch sind mit der zunehmenden Zahl an verfügbaren Linked Data-Datensätzen zwei neue Herausforderungen entstanden, deren Lösung noch aussteht.

Auf der Datenebene erweist sich die konsistente Vernetzung heterogener Datenbestände als Herausforderung. Im Rahmen eines Bestands werden Entitäten als Ressourcen behandelt, die durch einen *Uniform Resource Identifier* (URI) eindeutig identifiziert werden. Diese Ressourcen werden mithilfe von Prädikaten beschrieben, die ebenfalls eindeutig durch URIs adressiert werden können. Da denselben Ressourcen in verschiedenen Datenbeständen unterschiedliche Identifikatoren zugewiesen werden können und diese zudem häufig mithilfe verschiedener Vokabulare beschrieben werden, ist es sowohl auf der Ebene der Ressourcen als auch auf der Ebene der Vokabulare bedeutsam, Identitätsbeziehungen durch typisierte Links explizit zu machen (vgl. Heath/Bizer 2011: 22f.).[119] Um anzuzeigen, dass sich verschiedene URIs auf dieselbe Ressource beziehen und somit über dasselbe informieren, werden konventionell Identitätslinks – owl:sameAs – gebraucht, deren Bedeutung in der zweiten Version der Web Ontology Language (OWL 2) wie folgt definiert ist: »The property that determines that two given individuals are equal« (W3C 2009). Die formallogische Semantik des owl:sameAs-Prädikats orientiert sich an dem von Leibniz formulierten Prinzip der Ununterscheidbarkeit, welches besagt, dass zwei Entitäten x und y dann identisch sind, wenn sämtliche wahren Aussagen über x ebenfalls auf y zutreffen und umgekehrt sämtliche wahren Aussagen über y auch für x gelten (vgl. Halpin/Hayes 2010). Identität ist diesem Verständnis zufolge symmetrisch und transitiv. Dies hat, wie Halpin und Hayes darlegen, zur Konsequenz, dass derart formal-semantisch spezifizierte owl:sameAs-Links nicht als Identitätsbehauptungen begriffen werden, sondern als Faktum:

119 | Um die Vielfalt unterschiedlicher Beschreibungsvokabulare einzugrenzen, wird empfohlen, auf bereits etablierte Vokabulare zurückzugreifen, wie z.B. das Dublin Core-Metadatenvokabular, das Friend-of-a-Friend (FOAF)-Vokabular und das Semantically-Interlinked Online Communities (SIOC)-Vokabular (vgl. Heath/Bizer 2011: 24f. und 61f.).

»The real trick with *owl:sameAs* is that it works both ways: as it is both symmetric and transitive, so that anyone can link to your data-set with *owl:sameAs* from anywhere else on the Web without your permission, and any statement they make about their own URI will immediately apply to yours.« (Halpin/Hayes 2010)

Die formale Semantik des owl:sameAs-Prädikats abstrahiert Informationen von ihrem Entstehungskontext und suggeriert, dass die im *Web of Data* publizierten Informationen kontextunabhängig gültig sind. Explizit wird diese Annahme in der Interpretation von RDF-Tripeln, die als Fakten und nicht als Behauptungen begriffen werden: »OWL semantics treat RDF statements as facts rather than as claims by different information providers« (Heath/Bizer 2011: 23). Das Abstrahieren vom Entstehungs- und Publikationskontext einer Information stellt ein Problem dar, das in der Linked Open Data-Gemeinschaft zu einer »significant uncertainty« (Heath/Bizer 2011: 23) geführt hat. Zur Lösung des Problems geben Tom Heath und Chrstian Bizer in ihrer Monographie *Linked Data: Evolving the Web into a Global Data Space* Folgendes zu bedenken: »[K]eep in mind that the Web is a social system and that all its content needs to be treated as claims by different parties rather than as facts« (Heath/Bizer 2011: 23). Mit diesem Hinweis vermögen die Autoren die konstatierte Unsicherheit nicht auszuräumen, sondern gestehen allenfalls ein, dass sich die in der Interpretation von owl:sameAs-Links und RDF-Tripeln eingeschriebene Idee des Web of Data als einem Netz reiner Information an der sozialen Praxis bricht.

Dieses Spannungsverhältnis von Theorie und Praxis kann exemplarisch an zwei Problemen des Gebrauchs von Identitäts-Links aufgezeigt werden. Erstens ermöglicht die symmetrische und transitive Interpretation des owl:sameAs-Prädikats die gezielte Manipulation der Informationen, welche von unterschiedlichen Nutzern mit einer Ressource in Verbindung gebracht werden. Anders gewendet: Das Web of Data basiert auf der impliziten Annahme, dass alle Mitwirkenden wahre Informationen zur Verfügung stellen bzw. dass sie dies zumindest anstreben. Für *Bullshit*, der sich nach Ansicht des Philosophen Harry Frankfurt (2006) durch die Indifferenz gegenüber der Wahrheit von Aussagen auszeichnet, ist im Web of Data demzufolge kein Platz.[120] Das zweite Problem besteht in der uneindeutigen Verwendung des owl:sameAs-Prädikats, die dessen formaler Definition zuwiderläuft. Wie Halpin und Hayes in *When owl:sameAs isn't the Same* aufzeigen, wird dieses Prädikat in der Praxis dazu verwandt, um unterschiedliche Identitäts- und Ähnlichkeitsbeziehungen zwischen Ressourcen auszudrücken, die logisch nicht äquivalent sind. Hinsichtlich der formalen Definition von owl:sameAs stellt dies

120 | Eine Person redet Frankfurt zufolge immer dann *Bullshit*, wenn ihm bzw. ihr egal ist, ob das Gesagte wahr oder falsch ist: »Der Bullshitter hingegen verbirgt vor uns, daß der Wahrheitswert seiner Behauptung keine besondere Rolle für ihn spielt. […] Es ist ihm gleichgültig, ob seine Behauptungen die Realität korrekt beschreiben. Er wählt sie einfach so aus oder legt sie sich so zurecht, daß sie seiner Zielsetzung entsprechen.« (Frankfurt 2006: 62f.)

einen Missbrauch dar, der zwar konstatiert, aber wahrscheinlich nicht ausgeräumt werden kann. Daher gelte es nicht, zu versuchen die Bedeutung der Identitäts-Links definitorisch festzulegen, sondern umgekehrt auch als Effekt ihres Gebrauchs zu verstehen oder, wie Halpin und Hayes abschließend fordern, »as functions of their actual use« (Halpin/Hayes 2010).

Eine Perspektive für einen möglichen Lösungsansatz kann die medienphilosophische Reformulierung des Problems von Linked Open Data eröffnen. Die einzelnen Datenbestände im Web of Data enthalten in erster Linie keine Fakten, sondern Aussagen im Sinne Foucaults. Indem man jedoch versucht, diese als reine Fakten zu behandeln und sie so von ihrem Entstehungs- und Publikationskontext abzulösen, verlieren die Datenbestände ihren Aussagestatus. Eine Zeichenfolge konstituiert nach Ansicht Foucaults erst dann eine Aussage, wenn »sie zu ›etwas anderem‹ [...] eine spezifische Beziehung hat« (Foucault 1981: 129). Die Spezifik dieser Beziehung zu »etwas anderem« geht im Web of Data verloren, auch wenn die Semantik von Linked Data in RDF-Tripeln expliziert wird. Die Aussagefunktion fällt mit dem Gehalt einer Äußerung nicht in eins, sondern geht dieser voraus:

»Sie [die Aussage, M.B] ist vielmehr mit einem ›Referential‹ verbunden, das nicht aus ›Dingen‹, ›Fakten‹, ›Realitäten‹ oder ›Wesen‹ konstituiert wird, sondern von Möglichkeitsgesetzen, von Existenzregeln für die Gegenstände, die darin genannt, bezeichnet oder beschrieben werden, für die Relationen, die darin bekräftigt oder verneint werden. Das Referential der Aussage bildet den Ort, die Bedingung, das Feld des Auftauchens, die Differenzierungsinstanz der Individuen oder der Gegenstände, der Zustände der Dinge und der Relationen, die durch die Aussage selbst ins Spiel gebracht werden; es definiert die Möglichkeiten des Auftauchens und der Abgrenzung dessen, was dem Satz seinen Sinn, der Proposition ihren Wahrheitswert gibt.« (Foucault 1981: 133)

Bezogen auf den Linked Data-Kontext bedeutet dies, dass es keine reinen Informationen über Fakten oder sogar reine Fakten gibt, die im Web of Data zugänglich gemacht und miteinander vernetzt werden könnten. Es handelt sich stets um Aussagen, in deren Rahmen Zeichenketten (mediale Konstellationen) als faktische Informationen über etwas interpretiert werden.

Aussagen verweisen, wie Deleuze das foucaultsche Konzept rekonstruierend herausgearbeitet hat, auf drei Räume – einen *kollateralen*, einen *korrelativen* und einen *komplementären* Raum – die sie aufspannen und in denen sie sich gleichermaßen situieren (vgl. Deleuze 1992b: 14ff.). Den kollateralen Raum bilden die anderen Aussagen, die eine Aussage umgeben, ihr Vorausgehen und Nachfolgen. Als korrelativen Raum bezeichnet Deleuze »die Beziehung der Aussage [...] zu ihren Subjekten, ihren Objekten und ihren Begriffen« (Deleuze 1992b: 16). Der komplementäre Raum ist das nicht-diskursive Außen der Aussage, dass das technisch-materielle, institutionelle und ökonomische Milieu bildet, in dem die Aussage artikuliert wird. Durch die Unterscheidung des kollateralen, korrelativen und

komplementären Raums der Aussage sind nicht nur drei Beschreibungs- und Analysedimensionen von Aussagen benannt, sondern auch eine notwendige und hinreichende Bedingung für das Auftauchen von Aussagen. Diese müssen in den drei Räumen verortet werden können, um als Aussagen interpretiert werden zu können. Um Linked Data als Aussagen verstehen und gebrauchen zu können, müssen sie im Web of Data hinsichtlich dieser drei Räume prinzipiell verortet werden können.

Die Probleme, welche aus der Behandlung von Linked Data als Fakten herrühren, hängen eng mit einer zweiten praktischen Herausforderung zusammen, auf die abschließend noch einzugehen ist. Das Semantic Web im Allgemeinen und Linked Open Data im Besonderen basieren auf der Idee der Entkopplung der im Web verfügbaren Informationen von ihrem Gebrauch im Rahmen bestimmter Anwendungen, d.h. unter anderem von ihrer Präsentation auf Webseiten.[121] Die Frage, die sich hieran anschließt ist, wie derartige Anwendungen von Linked Data-Informationen aussehen und auf welche Weise sie mit Informationen umgehen können.

Heath und Bizer unterscheiden hier domänenspezifische Anwendungen von generischen Anwendungen (vgl. Heath/Bizer 2011: 85). Das charakteristische Merkmal domänenspezifischer Anwendungen ist, dass sie nur auf spezifische Linked Data-Ressourcen oder bestimmte Typen von Informationen im Web of Data zurückgreifen, um sie im Kontext der Anwendung zu präsentieren. Ein Beispiel hierfür ist das Musikportal der British Broadcasting Corporation (BBC).[122] Für jede Musikerin, jeden Musiker und jede Band, deren Musik auf einem der Radiosender der BBC gespielt wird, existiert im BBC Music-Portal eine Webseite, auf der sich Informationen darüber finden, in welchen Sendungen welche Musikstücke des Künstlers gespielt worden sind (vgl. Shorter 2008; Scott 2008; Kobilarov et al. 2009). Darüber hinaus enthält die Künstlerwebseite vielfältige weitere Informationen, wie z.B. biographische Daten, Links zu anderen Seiten u.v.m. Diese Informationen stammen zu einem großen Teil nicht von der BBC, sondern werden automatisch von anderen Quellen aggregiert und in die Seite integriert. Hierfür wird hauptsächlich auf die Linked Data-Bestände von DBpedia und der MusicBrainz-Datenbank zurückgegriffen, wodurch es beispielsweise möglich ist, die von der BBC gespielten Interpreten mit ihrem jeweiligen Wikipedia-Eintrag zu verknüpfen und infolgedessen die Künstlerbiographien automatisch von Wikipedia in das BBC Music-Portal zu integrieren.[123] Dass es sich um eine Linked Data-Anwendung handelt,

121 | Die Parallele zum Problem der Datenunabhängigkeit, welches die Entwicklung von Datenbankmanagementsystemen angeleitet hat, ist an dieser Stelle unverkennbar.

122 | Das Portal *BBC Music* ist unter www.bbc.co.uk/music (zuletzt aufgerufen am 10.09.2013) erreichbar.

123 | Das Projekt *DBpedia* (dbpedia.org/) extrahiert strukturierte Daten aus der Online-Enzyklopädie *Wikipedia* und stellt diese als Linked Data zur freien Verfügung. *MusicBrainz* (musicbrainz.org/) ist ein kollaboratives Projekt zur Versammlung von Musik-Metadaten, die gemäß den Linked Data-Richtlinien publiziert werden.

bleibt den Nutzern der Webseite verborgen. Das Web of Data wird in der Black Box domänenspezifischer Anwendungen unsichtbar.

Hierin besteht ein Unterschied zu generischen Anwendungen, die es den Nutzern ermöglichen, mit allen im Web of Data verfügbaren Informationen zu interagieren. Generische Anwendungen verbergen das Web of Data nicht in der Tiefe des Computers, sondern machen es durch ihre Benutzerschnittstelle an der Oberfläche sichtbar. Die Möglichkeit derartiger Anwendungen wird von Heath und Bizer mehrfach herausgestellt und sogar zu einem Wesenszug von Linked Data stilisiert: »The use of Web standards and a common data model make it possible to implement generic applications that operate over the complete data space. This is the essence of *Linked Data*« (Heath/Bizer 2011: 5). Den durchaus großen Erwartungen und Hoffnungen entsprechen erste prototypische Anwendungen, die die Interaktion mit dem Web of Data am Modell des Webbrowsers und am Modell von Websuchmaschinen umzusetzen versuchen (vgl. Heath/Bizer 2011: 86f.). Es ist jedoch fraglich, ob Browser und Suchmaschine die geeigneten Modelle für die Interaktion mit dem Web of Data darstellen. In Bezug auf den *Disco – Hyperdata Browser* (vgl. Bizer/Gauß 2007), der das hypertextuelle Navigationsprinzip des traditionellen Web direkt auf das Web of Data überträgt, stellen Heath und Bizer fest: »Structured data, however, provides human interface opportunities and challenges beyond those of the hypertext Web« (Heath/Bizer 2011: 86). Vor ähnlichen Herausforderungen stehen Linked Data-Suchmaschinen, wie z.B. *Sig.ma* (www.sig.ma), die den Nutzern die schlagwortbasierte Suche erlauben. Als Ergebnis werden nicht nur Links zu Ressourcen zurückgegeben, die dem Suchkriterium entsprechen, sondern auch ein Überblick über die dort verfügbaren Informationen. Angeordnet werden diese Informationen nach Attributen und nicht nach Relevanz. Darin besteht jedoch ein zentrales Leistungsmerkmal von Websuchmaschinen, die Orientierung im Web ermöglichen, indem sie die Aufmerksamkeit der Nutzer steuern. Ein Äquivalent bei Linked Data-Suchmaschinen gibt es noch nicht.

Weder das Modell des Browsers, noch das Modell von Websuchmaschinen lassen sich in einfacher Weise auf das Web of Data übertragen und so stellt die Frage nach einem brauchbaren und nutzerfreundlichen generischen Interface die zweite zentrale Herausforderung dar. Dies hat Ora Lassila, der neben Berners-Lee ebenfalls als ein Vordenker des Semantic Web zählt, bereits 2007 in einem Blogeintrag herausgestellt: »[M]ost of the remaining challenges to realize the Semantic Web vision have nothing to do with the underlying technologies involving data, ontologies, reasoning, etc. Instead, it all comes down to *user interfaces* and usability« (Lassilia 2007).

Wie lässt sich also die hier entwickelte Perspektive auf Datenbanktechnologien zusammenfassen? Ein angemessenes Verständnis der technischen Logik der Versammlung, Verwaltung und Verfügbarmachung von digitalen Informationen *in* und *mit* Computern verlangt eine differenzierte Betrachtung auf drei Ebenen: der Apparatur, der Architektur und der Verfahren. Ausgehend von der Betrachtung

der Festplatte als dem apparativen Horizont von Datenbanktechnologien wurden die Entwicklung von Architekturen für Datenunabhängigkeit sowie Verfahren des computertechnischen Umgangs mit *Bedeutung* im doppelten Sinn von *Gehalt* und *Relevanz* diskutiert. Es wurde deutlich, dass der Eindruck der Immaterialität digitaler Information unter anderem auf der zunehmenden Entkopplung der computertechnischen Verwaltung und Verarbeitung von Informationen beruht, die durch Datenbankmanagementsysteme geleistet wird. Diese materiellen Voraussetzungen der Immaterialität digitaler Informationen bleiben beim Gebrauch von Datenbanken jedoch gemeinhin verborgen. Aus dieser Unsichtbarkeit der materiellen Infrastrukturen resultiert die problematische Vorstellung, es handele sich um eine Eigenschaft von digitalen Informationen und nicht um ein Leistungsmerkmal von Informationssystemen. Einer solchen Naturalisierung digitaler Informationen kann durch die Betrachtung und Reflexion der architektonischen Gestaltung von Informationssystemen entgegengewirkt werden.

Die Kritik an Naturalisierungstendenzen eröffnet zugleich eine Analyseperspektive auf Verfahren des computertechnischen Umgangs mit Bedeutung. Sofern bedeutungstragende Informationen in DBMS automatisch verwaltet werden, beruht dies auf der Übersetzung von Information *über* Realität in Information *als* Realität. Durch die Spezifikation des Gehalts von Information im konzeptuellen Schema wird jedoch nie *die*, sondern stets nur *eine* Bedeutung formal expliziert. Das Gleiche gilt für Ontologien im Kontext des Semantic Web. Infolgedessen sind die in Datenbanken oder im Web of Data versammelten Informationen nie bloße Fakten, sondern Aussagen, die in spezifischen Kontexten getroffen werden und von einer bestimmten medientechnischen Konfiguration abhängen. Eine der größten Herausforderungen für die Realisierung der Semantic Web-Vision stellt, wie gezeigt wurde, die Bewahrung des Aussagecharakters von Informationen dar.

Den betrachteten Verfahren der Explikation des Gehalts von Information *für* Computer stehen in der digitalen Medienkultur algorithmische Verfahren der Zuschreibung von Bedeutung *durch* Computer gegenüber: Als Relevanzbewertungstechnologien schaffen Websuchmaschinen Orientierung, indem sie unüberschaubare Mengen an Informationen ordnen und hierdurch die Aufmerksamkeit der Nutzer auf Inhalte lenken, die für sie potenziell von größerem Interesse sind als andere. An die Stelle der vorgängigen Explikation von Informationen im konzeptuellen Datenbankschema oder in Webontologien tritt die nachträgliche Analyse durch Algorithmen. Dieser Unterschied ist jedoch allenfalls relativ, da auch Suchmaschinen stets auf Datenbanktechnologien angewiesen sind. Eine Analyse der technischen Bedingtheit der digitalen Medienkultur kann infolgedessen nicht bei der holzschnittartigen Gegenüberstellung von Daten(strukturen) und Algorithmen stehen bleiben, sondern muss beobachten, wie beide Seiten in konkreten Informationssystemen miteinander verschaltet sind, und danach fragen, wie sich dies auf je unterschiedliche Weise in mediale Praktiken mit digitalen Informationssammlungen einschreibt.