Wissensmodellierung

Diplomarbeit
zur Erlangung des Magistergrades der Philosophie
an der Fakultät für Human- und Sozialwissenschaften
der Universität Wien

eingereicht von

Nikolai Juršič

Wien, am 1. April, 2003

Alle Dinge nämlich, die mir einfallen,
fallen mir nicht von der Wurzel aus ein,
sondern erst irgendwo gegen ihre Mitte.
(Kafka)

Inhaltsverzeichnis

Einleitung

Die Menschheit beschäftigt sich seit je her mit der Speicherung von Wissen, egal ob auf Stein, Papier oder digitalen Medien. Jede Zeit scheint ihre spezifische Verbreitungsmöglichkeit gefunden zu haben.
Heute haben wir einen historischen Einschnitt erreicht. Dank Internet kann die Veröffentlichung von Inhalten gleich leicht geschehen, wie die Suche nach bestimmten Inhalten. Relativ schnell war jedoch ein Punkt erreicht, an dem durch die rasante Verbreitung von verteilten Datenobjekten von einer unbewältigbaren Flut an Information die Rede war. Gleichzeitig wurde Information bestimmender ökonomischer Faktor und mit spezieller Software bearbeitbar. Die vorliegende Arbeit beschäftigt sich mit der Computer vermittelten Modellierung von referenzierbaren Wissen, anhand von Gesetzmäßigkeiten, wie sie bereits bei Sprache auftreten.
Im ersten Kapitel begreife ich den antiken Wissensbegriff in Kontrast zu einem neuzeitlichen Verständnis. Danach wird das Thema materialisiertes Wissen sein. Außerdem werde ich die Orientierungslosigkeit wegen den stark ansteigenden Bibliotheksbeständen des 19. Jahrhunderts beschreiben, die einige Gemeinsamkeiten mit der heutigen Situation im Internet teilt.
Im nächsten Abschnitt sollen Daten als Grundlage der Informations- oder Wissensmodellierung dargestellt werden. Es soll gezeigt werden, dass für die Manipulation großer Datenmengen das Verständnis ihrer Struktur von herausragender Bedeutung ist. Ebenso werden die Grundkonzepte heutiger Datenbankmodelle vorgestellt, neben der einfachen Organisationsform der Baumstruktur.
Der Schrift mit ihrer herausragenden Rolle der Wissensrepräsentation entspricht der elektronische Text in einer Computer vermittelten Realität, beiden gemein ist die Externalisierung von Sprache. Ein Ansatz künstlicher Metasprachen zur Textkodierung erweitert die natürliche Auszeichnungsfunktion von Schrift und Sprache durch deskriptives Markup. Mit der Erweiterung von Markup für inhaltliche Auszeichnungen wird es möglich, Texte durch ihre Struktur abstrakt zu modellieren.
Welche Möglichkeiten Hypertextsysteme bieten und was Hypertext ausmacht wird im nächsten Kapitel behandelt. Weiter soll die Flexibilität in der Modellierung herausgearbeitet werden, die elektronischer Text mit deskriptiv ausgezeichneter Struktur gewinnt. Ein im dritten Abschnitt vorgestellter Standard für eine unter anderem deskriptive Textkodierung war Ausgangspunkt für das heutige World Wide Web (WWW). Wieweit das WWW auf offene Standards aufsetzt, welchen Standardisierungen selbst Standards unterliegen und was offene Standards vor proprietären auszeichnet, wird im fünften Kapitel beantwortet.
In den letzten beiden Kapiteln wird die Metasprache XML [1] einmal als primäres Textkodierungssystem vorgestellt, wie auch als Möglichkeit einer standardisierten sekundären Strukturmodellierung. Weiter werden die Grundkonzepte der XML Topic Maps (XTM) dargestellt, eine bereits als ISO Standard geführte Grammatik zur sekundären Modellierung von abstrakten Einheiten, den To-

[1] eXtensible Markup Language

pics. Der Standard befindet sich im Anhang sowie eine Anwendung gemäß der Grammatik von XTM, die als Ausgangsbasis den Wittgenstein Nachlass heranzieht.
In diesem speziellen Fall soll dem Fehlen des aktiven Buchautors entgegen gewirkt werden durch die Möglichkeit des elektronischen Textes, flexibel mit seiner Struktur umzugehen. Daraus lassen sich verschiedene Sichtweisen in einem Text erzeugen, die unterschiedliche Ziele verfolgen und eigenen Gesetzen gehorchen.
Da Topic Maps eine Ausprägung semantischer Netze sind, können mit ihnen semantisch aufgeladene Navigationssysteme über beliebiges Datenmaterial erzeugt werden. Der Standard sieht ebenfalls die Verschmelzung mehrerer Topic Maps aus verschiedenen Domänen oder Wissensbereichen vor, so dass nach einiger Zeit ein allumfassendes begriffliches Netzwerk entstehen kann. Denkbar ist, dass das WWW in seiner heutigen Ausprägung um ein Netz von begrifflichen Bezügen angereichert wird, um die Exploration zusammenhängender Wissensbereiche zu ermöglichen.

1 Wissen

Die Informations- oder Wissensgesellschaft ist in aller Munde, das Verständnis derselben allerdings ist unklar. Mindestens genauso unscharf sind die Einzelbegriffe Information und Wissen, die aber dennoch in allen erdenklichen Bereichen häufig verwendet werden. Ein verhältnismäßig höheres Vorkommen dieser Begriffe lässt sich im Kontext einer Informations- und Kommunikationstechnologie beobachten, und ebensosehr scheint mit ihrer inflationären Verwendung eine proportionale Verständnislosigkeit einherzugehen. Wissen wurde zum Buzzword des eBussiness.
Etwas differenzierter sieht das eine Kulturwissenschaft, manchmal zwar bloß kulturpessimistisch und somit wenig hilfreich, ein anderes Extrem spiegelt sich in der Technikmanie. [Hartmann, 1999, 11] nennt es *die beliebte Identifizierung des technisch Machbaren mit dem sozialen Nutzen* und warnt vor einer *Gleichsetzung des optimierten Datenverkehrs mit einer Verbesserung der gesellschaftlichen Kommunikation.*
Der Wissensbegriff in der abendländischen Tradition von Platon über Kant wird gegen Glauben und Meinen abgegrenzt. Platons berühmte Definition lautet: ,,Wissen ist wahre, mit Begründung versehene Meinung." Gemeinsam ist dem Wissen, Glauben und Meinen das Für-wahr-Halten eines Sachverhaltes, voneinander abgrenzen lassen sich die drei Begriffe erst durch die jeweiligen Gründe des Für-wahr-Haltens.
Ein grundloses Für-wahr-Halten ist ein bloßes Glauben, Für-wahr-Halten mit nur wahrscheinlichen Gründen gilt als Meinung. Entgegen diesen stark subjektiven Ausprägungen liegt Wissen vor, wenn es gute Gründe gibt etwas für wahr zu halten. Diese Gründe sind in der Regel nachprüfbar und garantieren somit die Wahrheit des Gewußten. Sie sind im wesentlichen kultur- und zeitabhängig, stets einem Wandel unterzogen. Heute sind Erfahrung, logischer Beweis oder Autorität gute Gründe, etwas für wahr zu halten. Erfolgt eine Begründung des Wissens methodisch, sprechen wir von wissenschaftlichem Wissen.
In der Literatur eines Faches wie dem Informationsmanagement unterscheidet sich Wissen häufig in *implizites* (engl.: tacit knowledge) und *explizites* Wissen. [Voß, 2001] Letzteres ist gleichzeitig artikuliertes Wissen, lässt sich in den Medien speichern und kann unabhängig vom menschlichen Bewusstsein existieren. Implizites Wissen dagegen nicht, es steht vor der Artikulation und ist Wissen, das im Bewusstsein lokalisierbar ist.
Eine weitere Differenzierung lässt sich anhand einer Klassifikation von deklarativem (,,wissen, was") und prozeduralem Wissen (,,wissen, wie") machen. Deklaratives Wissen lässt sich mit einem durch die menschliche Sprache dargestellten Wissen gleichsetzen, prozedurales Wissen ist dann die psychologische Repräsentation des Wissens einzelner Menschen, wie die Fähigkeit des Laufens, Tanzens oder Schwimmens. [Voß, 2001, 10] meint, dass der Mensch Informationen deklarativ aufnimmt, die Speicherung nach einer Verarbeitung aber prozedural erfolgt. Somit wird es für den Menschen immer problematischer sich über ein Gebiet zu artikulieren, je mehr er darüber weiß.
Wissenschaftler haben sich immer schon mit Wissen beschäftigt, einmal mehr

mit der Entdeckung und Erforschung von neuem Wissen, einmal mit der Analyse seines Wesens. Die philosophische Tradition konzentriert sich auf diesen Diskurs mit ihrem Teilgebiet der Epistemologie. Eine moderne und kontroverse Betrachtung der Epistemologie über Wissen liefert Wilfrid Sellars, hier zitiert nach [Fried and Süßmann, 2001][116]:

> Charakterisieren wir eine Episode oder einen Zustand als einen des *Wissens*, so geben wir keine empirische Beschreibung dieser Episode oder dieses Zustandes; vielmehr plazieren wir sie im logischen Raum der Gründe, des Rechtfertigens und der Möglichkeit zu rechtfertigen, was jemand sagt.

Siedeln wir eine Episode oder einen Zustand im Raum der Gründe an, heißt das nicht auch gleichzeitig, eine empirische Beschreibung davon abzugeben. [Fried and Süßmann, 2001] sehen darin eine Anfälligkeit der Epistemologie für einen naturalistischen Fehlschluß. Durch die Entwicklung der modernen Wissenschaften hin zu den Naturwissenschaften hat sich der Begriff des Wissens in einem philosophischen Diskurs vom logischen Raum der Natur und ihren Ursache-Wirkungs-Relationen entfernt und ist zu einem Subjektbegriff geworden.
Um aber den Begriff Wissen für unsere Belange besser verständlich zu machen, ist es sinnvoll das Phänomen zu verstehen, das die Bezeichnung unserer Gesellschaft als eine Wissensgesellschaft prägte.

1.1 knowledgeable society

Der amerikanische Soziologe Robert E. Lane prägte den Begriff der *Wissensgesellschaft* (engl.: knowledgeable society) im Jahre 1966, popularisiert hat ihn jedoch Daniel Bell 1973 in seinem Buch *The Coming of Post-lndustrial Society* [Nelkin, 1987]. Seither bürgerte sich *Wissensgesellschaft* als Trendbegriff in allen möglichen Bereichen ein, egal ob in Wissenschaft, Politik oder Marketing.
Nehmen wir diese Bezeichnung unserer Gesellschaft ernst, sind wir einem Strukturwandel verfallen, die Wissensgesellschaft löst die Industriegesellschaft ab. Andererseits lässt sich Wissen immer schon als konstitutiv für eine kulturelle Einheit erkennen, und unter diesem Gesichtspunkt ist jede Gesellschaft eine Wissensgesellschaft.

> Der Wissensbestand einer Gruppe bildet das Grundelement ihrer Einheit und ihrer Persönlichkeit, und die Weitergabe dieses geistigen Kapitals ist die notwendige Voraussetzung für das materielle und soziale Überleben. [Leroi-Gourhan, 1995, 323]

Oberflächlich betrachtet ist der Grundbaustein der Wissensgesellschaft das Wissen, so wie die Industrialisierung der Grundbaustein der Industriegesellschaft war oder der Ackerbau der der Agrargesellschaft. Jedoch muss Wissen hier eine andere Funktion haben, denn nicht erst seit der zweiten Hälfte des 20. Jahrhunderts spielt Wissen eine tragende gesellschaftliche Rolle.

Eine neue Betrachtung von Wissen gibt es tatsächlich im Feld der Ökonomie. Hier hängt die Wertschöpfung wesentlich vom Wissen ab, Wissen wird somit zum strategischen Gut. Gleichzeitig sind Wissen und Bildung nicht mehr alleine Aufgabe des Staates, große Unternehmen haben Forschungs- und Entwicklungsabteilungen sowie Einrichtungen zur innerbetrieblichen Weiterbildung. Wissen wird zum Kosten-Nutzen-Faktor.
Durch die Entwicklung der Informations- und Kommunikationstechnologien, beginnend in den 1980er Jahren, bekommt Wissen einen neuen Stellenwert, Wissen erscheint vorerst als Information und das fundierte Für-wahr-Halten gewisser Gründe tritt in den Hintergrund. Wissen ist Anteil einer nicht endenden Informationsflut, die durch Computer und Vernetzung zeit- und ortsunabhängig wird und sich mit Meinen und Glauben zunehmend unter dem Deckmantel der Information vermischt.
Der Begriff Information erfährt ebenfalls eine Neuorientierung anhand der in den 30er und 40er Jahren des 20. Jahrhunderts publizierten Arbeiten von Alan Turing, Norbert Wiener und John von Neumann. Im Gegensatz zu Wissen wird Information in der Kybernetik und der Informationstheorie unabhängig vom menschlichen Bewusstsein erfasst, Information muss nicht mehr verstanden werden, das Hauptaugenmerk liegt auf der Messbarkeit der Information, um einen Nutzen festzumachen.
Das anfangs ausgemachte Für-wahr-Halten verschwindet immer mehr zu Gunsten einer selbständig kreierten Informationseinheit, die als Wissen verkauft wird. Wissen wird *bearbeitbar*, losgelöst vom ursprünglichen Träger Bewusstsein wird Wissensmanagement eine eigenständige Disziplin. Wissen, betrachtet als spezielle Art von Information, wird dank Informations- und Kommunikationstechnologien auf der Grundlage der Digitalisierbarkeit zur Handelsware, jedoch völlig unabhängig von der klassischen Unterscheidung zwischen Meinen, Glauben und Wissen. Wissen wird durch die Neuen Medien, und zwar in den digitalen Speichermedien, deren Kapazität als praktisch unendlich aufgefaßt werden kann, materialisiert.

1.2 externalisiertes Wissen

Daniel Bell betrachtete Wissen in seinem oben erwähnten Werk als verdinglicht, in den vergangenen 70er Jahren jedoch bloß in gedruckter Form. Das Wissen wandelte sich in der modernen Gesellschaft von einer imaginären Sache, lokalisierbar im Gedächtnis, zu einer gegenständlichen Sache. Eine chronologische Betrachtung des Wissensbegriffs lässt einen gewissen Trend erkennen, der immer mehr das verdinglichte oder externalisierte Wissen in den Vordergrund stellt. Egal ob niedergeschriebenes, aufgemaltes, ausgedrucktes, digital kodiertes oder übertragenes Wissen, so unterschiedlich diese scheinen, haben sie doch eines gemeinsam, einen materiellen Aggregatzustand.

> Bis zur Entstehung der Druckkunst in China und im Westen besteht zwischen mündlicher und schriftlicher Überlieferung keine eindeutige Trennung. Die Masse des Wissens ist in die mündlichen und techni-

> schen Praktiken integriert; die Spitze des Wissens, seit der Antike in ihrem Rahmen unverändert, wird handschriftlich niedergelegt und auswendig gelernt. [Leroi-Gourhan, 1995, 326]

Das Ende der Möglichkeit einer Aufnahme des Weltwissens für ein menschliches Gedächtnis wird üblicherweise mit der Gründung der Encyclopædia Britannica im 18. Jahrhundert zusammengelegt.

> The first edition of the Britannica was published one section at a time, in ,,fascicles", over a three-year period, beginning in 1768. The three-volume set was completed in 1771 and quickly sold out.
> Encouraged by the success of the first edition, the publishers issued the second edition in 10 volumes (1777-84). The third edition, completed in 1797 and the first to include articles by outside contributors, comprised 18 volumes; the fourth, completed in 1809, boasted 20. [Britannica, 2003]

Spätestens die dritte Ausgabe mit ihren 18 Bänden und mehreren *outside contributors* konnte unmöglich von einem einzigen Menschen inhaltlich erfaßt werden. Ab diesem Zeitpunkt lässt sich eine Dialektik feststellen, die sich seit der Moderne immer stärker ausprägt; je schneller das materielle Wissen ansteigt, umso schwieriger wird es, im klassischem Modus zu wissen. Eine gefüllte Bibliothek alleine macht jemanden nicht zum Wissenden, erst durch geistige Aufnahme und Verarbeitung des materialisierten Wissens kommt es in die Köpfe.

1.3 Orientierungssysteme

> Das 18. Jahrhundert markiert in Europa das Ende der antiken Welt im Druckwesen und in den Techniken. Es bietet uns den größten Traditionsreichtum und zugleich die ersten Ansätze jener Transformation, aus der unser heutiger Zustand hervorgegangen ist. In den Büchern verschlingt das soziale Gedächtnis in wenigen Jahrzehnten die gesamte Antike, die Geschichte der großen Völker, Geographie und Ethnographie einer Welt, die endgültig Kugelgestalt angenommen hat, Philosophie, Recht, Wissenschaften, Künste, Techniken und eine aus zwanzig verschiedenen Sprachen übersetzte Literatur. Der Strom wird sich bis zu uns noch verbreiten, aber berücksichtigt man die Proportionen, so hat es zu keinem anderen Zeitpunkt in der menschlichen Geschichte eine so schnelle Ausdehnung des kollektiven Gedächtnisses gegeben. [Leroi-Gourhan, 1995, 327]

Das Problem Wissen verlagert sich weiter und die Publikationsfülle wächst unaufhörlich. So kommt der Bibliothekar Fremont Rider 1944 zur verblüffenden Erkenntnis, dass sich alle sechzehn Jahre der universitäre Bibliotheksbestand verdopple. Wir wissen heute, dass Riders statistischer Beweis nicht stimmte, aber dennoch war er es, der die Karteikarte (engl.: microcard) zum Standard für ein Orientierungsystem in den Bibliotheken einführte, um dem trotz allem problematisch erscheinenden Publikationswachstum Herr zu werden. [Molyneux, 1996]

> Das kollektive Gedächtnis erreichte im 19. Jahrhundert einen solchen Umfang, daß man von einem individuellen Gedächtnis nicht länger erwarten konnte, den Inhalt der Bibliotheken in sich aufzunehmnen. Es erwies sich als notwendig, das im gedruckten Gehirn der Gemeinschaft erstarrte Denken durch ein zusätzliches Netz zu organisieren, auf das sich ein überaus vereinfachtes Bild des Inhaltes projizieren ließ. Vor allem war es unerläßlich, daß die Zellen dieses neuen Netzes unbegrenzt erweitert und durch geeignete Umorganisation jeder möglichen Suchanordnung im dokumentarischen Material angepaßt werden konnten. [Leroi-Gourhan, 1995, 329]

Bis zur Einführung von Karteikartensystemen wurden im 18. und teilweise noch im 19. Jahrhundert lediglich Notizbücher und Buchkataloge für eine Orientierung in den Bibliotheken verwendet.
Heute sind Datenbanksysteme zum Standard in jeder größeren Bibliothek geworden, die manchmal sogar über eine Internetsuchmaske von überall aus der Welt nach den gewünschten Kriterien abgefragt werden können. Dies übertrifft wohl einen Traum von Goethe, der sich für einen Abgleich der Aufzeichnungen der Bücherbestände in den Bibliotheken zu Jena und Leipzig einsetzte.

1.4 Wissensmanagement

Die computerunterstützen Verfahren und Handhaben von einerseits elektronischen Texten, die online publiziert werden und andererseits Referenzsystemen, die in der Fülle der bereits gedruckten Wissensbestände durch ausgeklügelte Suchverfahren Erleichterung bringen, werfen ein neues Licht auf den Begriff der Wissensgesellschaft. Ebenso bieten die Neuen Medien, insbesondere die Informations- und Kommnunikationstechnologien, eine neue Möglichkeit, Orientierungssysteme in die Praxis umzusetzen.
Das Internet soll hier besonders als größte Datenbank der Welt erwähnt werden, das, ähnlich wie im 18. Jahrhundert, eine noch nie dagewesene Publikationsfülle ermöglicht. Die stark vereinfachte Produktion von Texten durch den Computer und der Wegfall von früheren Filtertechniken durch Verlage und Druckkosten lassen gerade im Internet die Publikationsfülle ungebremst und ohne Qualtitätssicherung ansteigen. Die Texte sind jetzt nicht mehr in Büchern, sondern in elektronische Formen verpackt. Was dieser Umstand für die Praxis der Orientierung in materialisiertem Wissen bedeutet, ist heute noch nicht absehbar.
Zahlreiche Initiativen streben mit dem Konzept der maschinell lesbaren Metadaten eine Generierung des unorganisierten Internets zu einem strukturierten Inhaltsangebot an. [Berners-Lee, 1998]
Leroi-Gourhan nimmt diesen ,,chaotischen" Zustand, den wir heute im Internet vorfinden, gedanklich schon vorweg, nur beschreibt er ihn noch an Hand der Printmedien:

> Wir haben im Bereich des Gedruckten diesen seit zwei Jahrhunderten erreichten Stand nicht überwunden, und wie auf allen Gebieten hat sich die Spitze der Entwicklung verschoben, sie liegt nun nicht

mehr im Bereich der Bücher, der freilich als dokumentarische Infrastruktur fortexistiert, sondern in Dokumentationselementen, die von jedem Kontext befreit sind. [Leroi-Gourhan, 1995, 328 ff]

2 Daten

In der Welt existieren Dinge und Handlungen. Die menschliche Sprache legt ein Netz über die Welt, die diese Ansicht als plausibel erscheinen lässt. Dinge haben Eigenschaften und stehen in verschiedensten Relationen zueinander, Handlungen können Dinge erst erschaffen, sie zerstören oder verändern.
Im Bereich des Computers können wir eine Analogie einführen: Im Begriffsnetz der Informatik gibt es zwei Pfeiler, Daten und Algorithmen. Daten sind wie Dinge, Handlungen wie Algorithmen. Eigenschaften von Daten nennt man Attribute. Algorithmen erschaffen, verändern oder zerstören Daten.
Die Strukturanalogie zwischen Computermodell und Wirklichkeit lässt die Universalität des Computers erahnen und prädestiniert ihn für eine Modellierung eines wohldefinierten Ausschnitts der Welt. [Rechenberg, 2000]
Aus dieser Sicht eines Informatikers sind Daten maschinenlesbare Anordnungen von Zeichen, die einer bestimmten Zeichenmenge angehören, gemeinsam ist allen Zeichenmengen, dass sie auf das kleinste Alfabet {0, 1}, das sinnvolle Aussagen zulässt, reduziert werden können. Diese Umwandlung der Zeichenmengen für bspw. eine digitale Textrepräsentation gehorcht bestimmten Regeln der Datenkodierung. [Voß, 2001]

2.1 Datenstrukturen

Datenstrukturen lassen sich auf verschiedenen Ebenen beschreiben, die unterste Ebene im Bereich der Informatik (engl.: computer science) ist die des Binärcodes, jedoch bleibt diese Ebene dem Anwender meist völlig verborgen.
Hier geht es lediglich um Datenstrukturen, die sich im Textbereich finden lassen. Es geht hier in einem besonderen Maße um Texte und somit um Daten und Datenstrukturen, die für Texte und alle erdenklichen Arten von Textbeschreibungen typisch sind. [Abiteboul et al., 1999], [Buneman, 2000] Unter diesem Gesichtspunkt erscheinen vorerst die folgenden drei abstrakten Datenstrukturen als relevant.

2.1.1 unstructured data

Unstrukturierte Daten folgen keinerlei definierter Struktur, ihrem Erzeuger sind keinerlei Einschränkungen bezüglich jedweder Struktur vorgegeben. Diese Art ist wohl die am häufigsten verwendete und begegnet uns gewöhnlich als reiner Text oder in Form von Multimediafiles.

2.1.2 structured data

Daten sind strukturiert, wenn sie in ein vorher wohldefiniertes Schema eingefügt werden können und dieses eindeutig ist. Die Daten werden nach Datentypen und Struktur des Schemas gespeichert, das Schema selbst wird in den meisten Fällen separat vom Datenbestand gehalten.
In der Datenbankwelt wird das Schema auch Datenmodell genannt, und es obliegt dem Datenbankprogrammierer, dieses Datenmodell zu erstellen, nach dem

die Anwender der Datenbank später Daten eingeben bzw. auslesen. Bei der Erstellung der Struktur sollten alle späteren Datentypen und -vorkommen auf einem abstrakten Niveau bereits bekannt sein, da später nur Daten aufgenommen werden können, deren Struktur vorher exakt festgelegt wurde.
Dokumente werden als strukturiert bezeichnet, wenn eine formale Beschreibung der Struktur mit dem Dokument verknüpft ist und die Struktur des Dokuments der Strukturbeschreibung der Grammatik folgt. Da diese Art von Dokumenten auch im Internet immer mehr an Gewicht gewinnt, wird sich da die Textanalyse in Zukunft mit Bedingungen erweitern lassen, die bis jetzt nur Datenbanken vorbehalten waren.

2.1.3 semistructured data

Semistrukturierte Daten haben im Gegensatz zu unstrukturierten Daten eine maschinell verwertbare Struktur, ein Schema zur Strukturdefinition ist aber nicht vorhanden. Ohne solche Grammatik sind Dokumente semistrukturiert. Dieser Umstand differenziert semistrukturierte Daten von strukturierten und unstrukturierten, eine genaue Abgrenzung ist jedoch erst nach genauer Analyse möglich.[2]

2.2 Datenbanken

Ich möchte hier mehrere Verfahren besprechen, die große Mengen von Daten mittels eigens entwickelten Programmen verwaltbar machen, wie Suchfunktionen, besondere Eingabemöglichkeiten neuer Daten und die Speicherung dieser Daten. Alle Verfahren, die diese geringe Gemeinsamkeit haben, möchte ich hier kurz als Datenbank bezeichnen. Sollte aus dem Kontext keine eindeutige Differenzierung hervorgehen, verwende ich jeweils den spezielleren Namen einer gewissen Datenbankausprägung.
Die schon kurz erwähnte Datenspeicherung und Verwaltung der Daten übernimmt ein komplexes Programm, das Datenbankverwaltungssystem (engl.: data base management system, DBMS). Erst die Datenbank (engl.: data base) gemeinsam mit dem DBMS ergeben das funktionsfähige Datenbanksystem.

2.2.1 unformatierte Datenbanken

Das wesentlichste Merkmal einer unformatierten Datenbank oder eines Dokumentenmanagementsystems (DMS) ist der unstrukturierte Aufbau der Dokumente; zum Beispiel liegen Dokumente als natürlichsprachige Texte (ASCII, formatierter Text, Postscript etc.), HTML-Seiten, E-Mails oder Verzeichnisstrukturen vor. Diese Arten von Datenbanken sind von besonderem Interesse, seit Unternehmen angefangen haben, Wissen zu modellieren. Jedoch ist jeweils durch die Unstrukturiertheit des Datenmaterials ein Analyseprozess und eine anschließende Aufbereitung mit Metadaten notwendig, um Struktur miteinzubringen. [Voß, 2001, 277] Denn erst die Aufbereitung des unstrukturierten Materials mit

[2] Das passiert im Abschnitt 6.4.

maschinenlesbaren Datenelementen kann die Funktionen gewährleisten, die ein DMS ausmacht.

2.2.2 relationale Modelle

Der Name relationale Datenbank folgt aus dem Umstand, dass alle Handlungen, die mit dieser Datenbank durchgeführt werden können, auf der relationalen Algebra beruhen. Das relationale Modell ist auch das theoretisch fundierteste einer Datenbankausprägung, dieser Umstand ist wohl der mathematischen Exaktheit der Operatoren der relationalen Algebra zu verdanken, die wiederum eine Erweiterung von Mengen-Operatoren darstellen. Diese Erweiterung sieht so aus, dass nicht Mengen, sondern Relationen manipuliert werden und dass das Ergebnis der Relationen-Operation wieder eine Relation ergeben muss.
Ein großer Vorteil der relationalen Datenbanken ist die Möglichkeit zur Normalisierung des Datenbestandes. Das bedeutet, mehrfach verwendete Einzeldaten werden nur ein einziges Mal gespeichert. Der entscheidende Vorteil liegt darin, dass eine Änderung eines Datums nur an einer Stelle erfolgen muss und der Datenbestand niemals widersprüchlich werden kann.
Die weite Verbreitung dieses Datenbanktyps hatte die Standardisierung einer eigenen Manipulationsprache zur Folge, der Standard Query Language (SQL), definiert in ISO/IEC 9075.

2.2.3 objektorientierte Modelle

Bei diesen Modellen werden Daten als Objekte behandelt, wie es aus dem objektorientierten Paradigma bekannt ist. Die Handlungen, die über Objekte ausgeführt werden, sind nicht mehr auf die relationale Algebra beschränkt, wie bei relationalen Datenbanken, sondern Einschränkungen ergeben sich rein aus dem objektorientierten Paradigma bzw. der konkreten Ausprägung des DBMS.
Dieser Umstand macht diese Form von Datenbanken sehr weit einsetzbar, die Grenze der Einsetzbarkeit ist von vorneherein nicht eingeschränkt und macht eine sehr wirklichkeitsnahe Datenmodellierung möglich.

2.2.4 objekt-relationale Modelle

Diese Hybridform aus den oben beschriebenen Ansätzen stellt eine Erweiterung des relationalen Modells durch objektorientierte Ansätze dar, die sich meist auf die Modellierung von abstrakten Datentypen beschränkt, jedoch hängt das genaue Verhältnis beider Ansätze vom jeweiligen Anbieter des DBMS ab.

2.3 Datenspeicherung

Bei Datenbanken übernimmt in der Regel das DBMS zusammen mit Methoden des Betriebssystems die Speicherung der Daten, die dann meist im Binärformat auf einem physikalischen Speichersystem festgehalten werden. Die erforderlichen Methoden zum Lesen und Schreiben werden vom jeweiligen Betriebssystem (operating system, OS) bereitgestellt. Binäre Daten haben die Eigenschaft, sich

nur mit einem speziellen Programm darstellen zu lassen, hier mit dem DBMS. Eine direkte Manipulation der Daten auf Rohdatenebene mittels Systemsoftware ist somit nicht möglich, diese Funktion übernimmt ebenfalls das DBMS.
Im Gegensatz hierzu stehen Textformate, die auf viele verschiedene Arten schließlich auf der Festplatte des Computers gespeichert werden können. Eine für strukturierte Dokumente sehr typische Art der Organisation auf physikalischen Speichermedien ist die Organisationsform des Baumes.

> Die Elemente oder Blöcke einer Datei können auch in einer als *Baumstruktur* (engl. tree) bezeichneten Anordnung gespeichert werden. Besondere Bedeutung für die Speicherung größerer und unregelmäßig wachsender Datenmengen haben so genannte *balancierte Bäume*, kurz *B-Bäume* genannt, erlangt. [Gaus, 2000, 35]

Bei strukturierten Dokumenten ist es möglich, die Daten auch ohne Programm, mit dem das Dokument erstellt wurde, lesbar darzustellen. Dazu reicht ein einfacher Texteditor, der bei allen Betriebssystemen bereits vorinstalliert ist, vorausgesetzt es wird ein Zeichensatz verwendet, mit dem der Texteditor arbeiten kann. Daraus ergibt sich ein wesentlicher Vorteil dieser Art von Dokumenten als Strukturmöglichkeit über Plattformgrenzen hinweg, denn binäre Formate sind immer an ein Programm gebunden, das wiederum vom jeweiligen OS abhängig ist.
Computer *verstehen* prinzipiell keine Symbole oder Zeichen, sie handeln bloß auf Ebene von langen Zeichenketten des Binärecodes, der rein auf Zahlen beruht. In der Informatik heißen Daten *codiert*, wenn jedes Symbol mit einer eindeutigen Nummer repräsentiert ist. Werden vom Texteditor nun Zeichen dargestellt, die wir als Text interpretieren, so liegt dahinter die Decodierung der Zahlencodes in Zeichen eines Zeichensystems (character set) zur Darstellung und nach dem Editieren dieser Datei werden alle Zeichen wieder in den Binärcode des Computers codiert.
Heute besitzen Texteditoren die Fähigkeit, verschiedene Zeichensätze zu verwenden. HTML akzeptierte anfänglich nur den ISO 8859-1 Zeichensatz, oft der Einfachheit halber bloß als Latin-1 bezeichnet. Mit Latin-1 können fast alle westeuropäischen Sprachen dargestellt werden, mit ISO 8859-2 hingegen, kurz Latin-2, werden die auf dem lateinischen Alphabet aufgebauten slawischen Sprachen in Mitteleuropa unterstützt. Heute gibt es Erweiterungen des Zeichensatzes ISO 8859 bis Latin 15. Diese Fülle an verschiedenen Zeichensätzen ist notwendig, wenn das World Wide Web seinem Namen gerecht werden soll.

2.4 Metadaten

Ganz allgemein trifft das meiste, was oben bereits über Daten gesagt wurde, ebenso auf Metadaten zu. Informell betrachtet sind die Karteikartensysteme der Bibliothekare die ersten standardisierten Metadaten. In Bibliotheken werden Titel, Autor, Editor, ISBN etc. in strukturgleicher Form archiviert, um die Orientierung in der ständig wachsenden Anzahl der Publikationen zu erleich-

tern.
Tim Berners-Lee definiert Metadaten für seine Vision des Semantic Web:

> Metadata is machine understandable information about web resources or other things. [Berners-Lee, 1997]

Weiter stellt er fünf Axiome für Metadaten auf:

1. Metadata is data.
2. Metadata may refer to any resource which has a URI. [3]
3. Metadata may be stored in any resource no matter to which resource it refers.
4. Metadata can be regarded as a set of assertions, each assertion being about a resource (A u1 ...). Assertions which state a named relationship between two resources are known links (A u1 u2). Assertion types (including link relationships) should be first class objects in the sense that they should be able to be defined in addressable resources and referred to by the address of that resource A in { u }
5. The development of new assertion types and link relationships should be done in a consistent manner so that these sort of assertions can be treated generically by people and by software.

[3] URI steht für Uniform Resource Indicator und wird im Abschnitt 6.6.1 näher bestimmt.

3 Markup

Gewöhnlicherweise zeichnen Studenten ihre Lernunterlagen aus, meist mir Texthervorhebungen wie Unterstreichungen etc. Eine andere Möglichkeit zur Auszeichnung besteht durch Textmarker, mit denen meist in leuchtenden Farben bestimmte Textbereiche hervorgehoben werden. Mit dem Erfolg, bei nochmaliger Durchsicht sofort einen Fokus auf den wichtigen Textbereich zu erhalten; beim Einsatz mehrerer Farben könnten verschiedene Typen von Auszeichnungen auftreten. Somit bekommen in den Lernunterlagen verschiedene Textfärbungen verschiedene Auszeichnungstypen zugeordnet.
Leider ist diese intuitive Art von Textauszeichnung (engl.: to mark up) heute mit dem elektronischen Text nicht so einfach möglich, der zur maschinellen Verarbeitung formale Textauszeichnungen benötigt, die meist mühsam in den eigentlichen Text integriert werden müssen. Der Sinn von Markup liegt darin, dem eingentlichen Textdokument auf eine bestimmte Weise dessen Bedeutungsgehalt zu erweitern, der außerhalb des eigentlichen Textes liegt.
Die Bedeutsamkeit von Markup kann leicht veranschaulicht werden, bspw. mit dem Lautlesen eines Textes; dabei wird Markup zwar „verarbeitet", das heißt einzelne Wörter werden selbstverständlich mit Pausen voneinander getrennt, Fragezeichen und Satzenden bewirken Frequenzmodulationen der Stimme. Diese Handlungsanweisungen für den Leser sind so sehr textimmanent, dass wir uns Text und Sprache ohne Markup nur schwer vorstellen können.

3.1 Markup ála Coombs et al.

Markup[4] bezeichnet heute meist die Formatierungs- und Strukturierungssprachen, die mit computerunterstützter Textverarbeitung entstanden sind. Nach diesem Abschnitt soll die Möglichkeit gegeben sein, Daten (Texte) nach beliebigen Strukturen zu modellieren, abhängig von der späteren Verwendung. Der erste Satz aus dem TLP würde ohne jegliches Markup so aussehen:

> dieweltistalleswasderfallist

3.1.1 punctuational

Dieses Markup, das in der Sprache als Pause oder als Frequenzmodulation auftritt, ist in einem geschriebenen Text in Form von Leerzeichen zwischen einzelnen Wörtern und Satzzeichen vertreten. Hier noch einmal den ersten Satz, nun mit punctuational markup:

> Die Welt ist alles, was der Fall ist.

[4] Die nachfolgende Kategorisierung verschiedener Markuptypen ist im Wesentlichen [Coombs et al., 1987] entnommen.

3.1.2 presentational

Presentational Markup, wie der Name schon sagt, betrifft rein die Beschreibung der Präsentation des Textes für Papier oder Bildschirm. Horizontale und vertikale Abstände zwischen einzelnen Textteilen, Seitenumbrüche, Numerierungen etc. sind Beispiele für diese Art von Textauszeichnung. Vorstellbar wäre auch jeweils eine neue Seite anzufangen, wenn ein neues Kapitel beginnt oder Kapitelüberschriften mit Abständen zum restlichen Text mehr abzugrenzen. Der genaue Einsatz des presentational Markups hängt vom jeweiligen Textautor bzw. Texteditor ab.

3.1.3 procedural

In vielen Textverarbeitungssystemen wurde presentational Markup durch procedural Markup ersetzt. Das bedeutet für die Autoren, dass sie sich gewisser Kommandos bedienen, die ebenso in den Text geschrieben werden, die das Textverabeitungssystem aber vom Normaltext unterscheiden kann, um darauf mit spezifischen Textformatierungen zu antworten.
Diese Form der Auszeichnung ist hard- und softwareabhängig, jede Software verwendet ihre eigene Markupnotation.

3.1.4 descriptive

Um procedural von descriptive Markup zu unterscheiden, lässt sich sagen, dass das erste bereits als Anweisung für eine konkrete Textformatierung beschrieben wurde. Descriptive Markup hingegen beschreibt einen bestimmten Textteil, sagt, was er ist oder zu welchem Typ von Textelementen er gehört, und nicht mehr. Textelemente können allen erdenklichen Typen angehören, im Detail: Absätze, Überschriften, Indexeinträge, Glossareinträge etc. Mit diesem Mittel ist es nun möglich inhaltliche Teile des Dokuments so zu beschreiben, dass sie für die verschiedensten Arten der Weiterverabeitung ausgezeichnet sind.

3.1.5 referential

Referential Markup referenziert auf Elemente, die sich meist außerhalb des Dokuments befinden. Das jeweilige Textverarbeitungsprogramm verwendet sie für vorher festgelegte Anwendungsfälle. Das könnten einmal aus praktischen Gründen der automatische Austausch von Abkürzungen mit ihrer langen Form sein, oder einfach externe Daten, die es zu aktualisieren gilt.

3.1.6 metamarkup

Mit Metamarkup wird die Möglichkeit bezeichnet, die es dem Autor gestattet, die Interpretation des Markups festzulegen, Markup auf einer Metaebene zu beschreiben. Im Bereich des descriptive Markup werden mit Metamarkup die einzelnen Elementtypen festgelegt, nach denen später einzelne Textstellen ausgezeichnet werden.

3.2 Standard General Markup Language

Computer und seine Anwendungen stellen die Infrastruktur bereit, um Daten in einer Menge zu verarbeiten, wie es ohne diese Techniken nicht möglich wäre. Riesige gekoppelte Satellitenantennen, Very Large Arrays (VLA), gerichtet ins All, empfangen Frequenzen zwischen 300 and 50,000 MHz und speichern diese Datenansammlungen unentwegt, um nach Strukturen zu suchen.[5]
Dokumente und Texte werden seit Jahren fast ausschließlich am Computer erzeugt. Das geschah bis Mitte der 80er Jahre des vorigen Jahrhunderts mit dem Schwerpunkt auf der Präsentation des Dokuments auf Papier oder dem Bildschirm selbst. TEX ist eine solche Formatierungssprache, mit der hervorragende Ergebnisse erzielt werden können, sofern es um die Formatierung gedruckter Texte geht. Die Textstruktur selbst bzw. die Bedeutung einzelner Textabschnitte sind so aber nicht maschinell nutzbar, ebenso ist ein formatierter Text mit TEX ausschließlich für eine Anwendung gedacht und nicht generalisierbar. Denselben Sourcecode bspw. für mobile computing, für Intranets, Bücher oder Internetkataloge zu verwenden ist nicht möglich mit reiner Formatierungsauszeichnung des Textes.
Der erste Schritt für eine Verbesserung dieser Situation war Ende 1960 durch zwei Projekte gegeben, die durch die Idee des generischen (engl.: generic) Markups Dokumente mit logischen, rein inhaltsbeschreibenden Tags (to tag = markieren, auszeichnen; tag = Auszeichnungselement) versahen. Somit wurde es möglich physikalisch verteilte Dokumente z.B. zusammenzufügen oder auch mit unterschiedlichen Programmen ein und dasselbe Dokument zu verarbeiten.
Der Präsident der Graphic Communication Association (GCA) präsentierte 1967 seine Idee von der Trennung zwischen Inhalt und Formatierung eines Dokuments. Das Projekt GenCode der GCA beschäftigte sich mit dem generischen Markup, um verschiedene Dokumente und unterschiedliche Dokumenttypen gemeinsam zu verarbeiten und ggfs. neu zusammenzustellen. Ein weiteres Projekt namens Generic Markup Language (GML)[6], initiiert von IBM, befasste sich mit generischen Tags, um damit meist technische Dokumente mit unterschiedlichen Programmen auf verschiedenen Computersystemen zu formatieren, zu editieren und zu durchsuchen.
Aufbauend auf dem großen Erfolg von GML startete das American National Standards Institute (ANSI) ein Projekt unter der Leitung von Charles Goldfarb zur Entwicklung eines Standards zur generischen Textauszeichnung. Die Standard General Markup Language (SGML) war im Entstehen. Institutionen wie die Association of American Publishers (AAP) oder das U.S. Department of Defense arbeiteten bereitwillig mit SGML und entwickelten SGML-Anwendungen, als 1986 die International Standards Organization (ISO) einen Standard für SGML (ISO 8879:1986) ratifizierte.
Doch was ist SGML eigentlich? SGML ist *die* Metasprache für die Definition von Markupsprachen, das heisst SGML stellt lediglich eine Menge von Regeln

[5] Struktur wäre ein Indiz für außerirdisches Leben.
[6] Das Acronym GML stimmt auch mit den Anfangsbuchstaben seiner Entwickler überein: Goldfarb, Mosher und Lorie.

zur Definition von Markupsprachen bereit, ist selbst daher keine Markupsprache. SGML Dokumente sind reine Textdokumente kombiniert mit Tags, die vorher in einer Document Type Definition (DTD) beschrieben werden und die für die Strukturauszeichnung des Dokuments herangezogen werden. Die DTD beschreibt die Grammatik, die syntaktischen Regeln und die hierarchische Struktur einer Dokumentklasse.
SGML hat aber auch Nachteile, aufgrund der großen Macht von SGML ist es praktisch unmöglich die Sprache effizient einzusetzen, lediglich Global Players können die nötigen Ressourcen aufbringen. Um das auszugleichen, ohne die Vorteile von SGML zu verlieren, entwickelte das World Wide Web Consortium (W3C) 1998 eine überschaubare und leicht anwendbare Untermenge von SGML, die Extended Markup Language (XML). Im Grunde ist jedes XML-Dokument auch ein SGML-Dokument, mit dem Untschied, dass XML mit seiner Einfachheit es jedem gestattet Anwendungen zu entwickeln. Auch sind die komplexen, sehr großen SGML-Anwendungen praktisch untauglich für verteilte Netzwerkanwendungen.

4 Text

Ganz allgemein ist Text eine nach gewissen Regeln geordnete Menge von Zeichen mit kommunikativer Funktion. Die Regeln sind auch bekannt als Grammatik und die Zeichenmenge ist ebenso wie die Grammatik kulturabhängig und ständigem Wandel unterzogen. Die kommunikative Funktion lässt sich fassen als eine gewisse Gedächtnisfunktion von Mitteilungen, die Texte gegenüber Sprache besitzen.
In Platons *Phaidros* wird Schrift als das Gedächtnis schmälernd und sinnverstellend abgelehnt, der Sinn der Schrift ist nicht mehr eindeutig in Erfahrung zu bringen. Die Möglichkeit der Rückfrage bleibt im Text verborgen. So sind die platonischen Dialoge ihrer Form nach dem Ideal einer mündlichen Kultur verpflichtet, mit einem unmittelbaren Sprecher, der Rede und Antwort steht.
Hier geht es hauptsächlich um die Form elektronischer Texte und ihre Eigenart.

4.1 elektronischer Text

Ein wichtiger Faktor, den alle anderen Textsorten nicht aufweisen, muss bei elektronischen Text beachtet werden. Zu seiner Erzeugung und Präsentation wird der mediale Ort des Computers mit Hard- und Software verwendet. Natürlich lassen sich Repräsentationen von Texten mittels Drucker auf Papier bannen, jedoch ist das nur *eine* Form der Repräsentation. Im Computer selbst werden die Transistoren im flüchtigen Arbeitsspeicher nach einem Muster, das den Text in einer Weise repräsentiert, angesteuert und strukturiert. Auch im Festspeicher der elektronischen Rechenanlage wird auf der magnetisierbaren Oberfläche der Harddisk eine dem Text entsprechende Struktur aufgebracht. Diese materiellen *Textstrukturen* können von hier aus weiter transformiert werden oder in eine andere materielle Struktur umgeformt werden; auf Bildschirm oder Drucker. Alle diese Strukturen können wir als Text ansehen, schließlich handelt es sich immer noch um eine nach gewissen Regeln geordnete Zeichenmenge.
Wichtig bei einem elektronischen Text ist seine Kodierung, denn von ihr und ihrer Ausprägung hängt ab, welche Sichtweisen auf den Text später realisierbar werden. So ist eine Darstellung einer Liste aller englischen Fachtermini aus einem italienischen Forschungsbericht eine Sichtweise auf den elektronischen Text. Wird dieses Feature jedoch vorher bei der Textproduktion nicht berücksichtigt, können die entsprechenden Fachtermini auch nicht entsprechend ausgezeichnet werden, um sie später auf Wunsch in einer Liste darzustellen.
Für Hartmann spiegelt sich ein Wechsel in der Reproduktion gesellschaftlichen Wissens, *in der es immer weniger um Textverarbeitung geht als um entsprechendes Daten- und Informationsmanagement.* [Hartmann, 2000, 20]

4.1.1 Hypertext

Ted Nelson prägte den Begriff *Hypertext* als eine Art von *non-sequential writing*, eine die herkömmliche Art der Texterzeugung erweiternde Textsequenz. Konkret erweitert Hypertext den elektronischen Text um die Möglichkeit der

Referenzierung und des Verweises.

> Im Lauf der Jahrhunderte, die Homer und den Großen Yu von den ersten Druckwerken des Okzidents und des Orients trennen, bildete sich mit der wachsenden Masse an aufgezeichneten Tatsachen auch ein Begriff von Referenz und Verweisung heraus, aber die Schriften bleiben allenthalben noch kompakte Folgen, die durch Siglen und Randbemerkungen rhythmisiert werden, an denen der Leser sich dann in der Art eines primitiven Jägers orientiert, d.h. eher entlang einer Bahn als auf einer Ebene. Die Umwandlung der Wortfolge in ein System von Orientierungsfeldern ist noch nicht erfolgt. [Leroi-Gourhan, 1995, 326]

Eine seit Anfang der Geistesgeschichte schlummernde Kulturtechnik findet endlich ihre Realisierung in einer der ganzen Welt auf einen Schlag gleichzeitig zur Verfügung stehenden Technik der elektronischen Textproduktion bzw. Rezeption.

> Hypertext is the presentation of information as a linked network of nodes which readers are free to navigate in a non-linear fashion. [Keep et al., 2000]

Unter Hypertext wird heute in den meisten Fällen die Hypertext Markup Language (HTML) verstanden. Dies ist, sieht man Hypertext als Konzept und nicht als Abkürzung für HTML, natürlich nicht immer richtig, jedoch die weitverbreiteste Form von Hypertext. HTML ist schlechthin die *killer application* von SGML und der auslösende Moment für den gewaltigen Durchbruch des World Wide Web (WWW).
In HTML werden Referenzierungen oder Hyperlinks wie alle Auszeichnungen mittels Tags erzeugt, geschieht dies außerhalb des eigentlichen Textes, muss die externe Adresse des zu referenzierenden Objekts in den Tag miteingeschrieben werden. Hyperlinks innerhalb eines HTML-Dokuments werden mittels SGML eigenen `ID/IDREF`-Attributen realisiert. Bei Referenzierungen außerhalb des Dokuments muss dies mit einem Mechanismus geschehen, der vom Internet-Browser selbst zur Verfügung gestellt wird, denn SGML alleine kann dies nicht erfüllen. Jedoch war die Entwicklung *um* SGML nie wirklich still, so wurden stets Erweiterungen entwickelt, wie die Hypermedia/Time-based Structuring Language (HyTime) und ist in ISO 10744:1992 definiert.
Das Besondere von HyTime ist, dass Elemente auch außerhalb ein und desselben SGML-Dokuments erreicht werden können. Dies macht und bietet somit unabhängig vom SGML-internen `ID/DREF-Attribut` zusätzlichen Bonus im Bereich von plattformübergreifender und systemunabhängiger Bereitschaft.
Ein schwerer Nachteil von HTML war es schließlich, dass Mechanismen, die im Standard nicht explizit ausgewiesen, jedoch im Laufe der Zeit erwünscht waren, durch proprietäre Lösungen der Browser-Hersteller selbst ausgeglichen wurden. Dieser Umstand hatte den unliebsamen Effekt, dass ein einmal erzeugtes HTML-Dokument in jedem auf dem Markt erhältlichen Browser anders

dargestellt wurde.
In HyTime, mit seinen ausgeklügelten Verlinkungsmechanismen, ist es nun bspw. möglich, jedes erdenkliche Objekt zu verlinken. Bestimmte Bytesequenzen bei unstrukturierten Datenobjekten wie Musik- oder Bilddateien können mit diesem Standard exakt angesprochen werden. Eine Weiterentwicklung und einige weiterführende Standards werden wir später als Satellitentechnik von XML kennenlernen.

4.2 Textkodierung

Die Praxis der elektronischen Texterzeugung seit der Entwicklung von SGML hat die Möglichkeit gezeigt, die Struktur von dem Layout des Dokuments klar zu trennen. Hieraus ergibt sich eine Einteilung nach diesen beiden Kriterien, die ein Text bedingt, nicht wie zuvor nach Differenzierung der erdenklichen Textauszeichnungen selbst. Die einfache Anwendung von HTML war sicherlich einer der ausschlaggebenden Momente ihres Erfolges, jedoch sind gerade bei HTML inhaltliche Textauszeichnung und Auszeichnung zur Textformatierung nicht getrennt, was eine der großen Schwächen von HTML ist. Bei neuer entwickelten Markupsprachen wird genau auf diese Trennung großer Wert gelegt, um die Vorteile von generischem Markup auch voll nutzen zu können. Die Tatsache, dass diese Trennung durch die Vermischung beider Markupkonzepte nicht möglich ist, führt zu einer weiteren Betrachtung von Markup im Bereich der Textkodierung.

4.2.1 Textstruktur

Generisches Markup ermöglicht es, jedes beliebige Analyseverfahren einer Textgenese als Strukturauszeichnung in den Text miteinzufließen lassen. Die Ebene der Strukturauszeichnung kann verschieden feinmaschig sein, eine abstrakte Strukturauszeichnung könnte auf der Ebene von einzelnen Kapiteln bestehen, das Gegenteil wäre auf Ebene der Phoneme geglückt. Somit können beliebige Sichtweisen auf ein und dasselbe Datenmaterial erzeugt werden, sofern die gewünschte Sichtweise mittels Strukturauszeichnung auch gesondert kodiert wurde. Das Einbringen von Markup zur Auszeichnung der Textstruktur lässt sich als kognitiver Akt beschreiben. Dieser Prozess kann manchmal mehr Zeit und Energie in Anspruch nehmen als die eigentliche Textproduktion. Hinter ihm muss immer auch eine inhaltsbezogene Textanalyse stehen.

4.2.2 Textpräsentation

Die oben angesprochenen Sichtweisen auf Text, repräsentiert durch Strukturmarkup, können mittels von der Struktur völlig getrennter Repräsentationstechniken nach Belieben dargestellt werden. Außerdem sorgt ein weiterer Standard für die Transformation von strukturierten SGML-Dokumenten für eine bestimmte Verwendung in einem beliebigen Medium. Die Document Style and Semantic Specification Language (DSSSL; ISO 10179:1996) macht es bspw.

möglich, für eine SGML-Anwendung eine DSSSL-Anwendung zu entwickeln, die dann jede Instanz der SGML-Anwendung auf einem bestimmten Ausgabemedium, z.B. Papier oder CD-ROM, darstellt.
Die in Abschnitt 3.1.6 besprochenen Nachteile für SGML gelten nun auch für DSSSL, so hat das W3C einen den heutigen Ansprüchen gerecht werdenden Standard entwickelt, um Dokumente für Papier und Bildschirm darzustellen, die Cascading Stylesheet Language (CSS).
CSS wurde immer wieder weiterentwickelt, verändert und schließlich gibt es heute einen weiteren Standard, der die Charakteristika von CSS beinhaltet, sie aber um die Möglichkeit der Transformation von Dokumenten erweitert. Die eXtensible Stylesheet Language (XSL) stellt somit Sprachkonstrukte wie unter anderem Iterationen bereit, um Format- und Strukturtransformationen durchzuführen.

4.3 Ordered Hierarchy Of Content Objects

Die Frage, wie Text kodiert werden soll, um einmal seiner Natur sehr nahe zu kommen und ebenso die Möglichkeiten einer maschinellen Verarbeitung zu optimieren, mündet irgendwann in der Frage, was eigentlich die Natur des Textes sei. Es gibt naturgemäß die verschiedensten Betrachtungsweisen und Erklärungsansätze von Text, abhängig von den verschiedensten Kontexten, wie Medium, Funktion oder Form, um nur einige zu nennen. Im Kontext der elektronischen Textkodierung gab es in den 90er Jahren des 20 Jahrhunderts im Bereich der SGML-Community einen Versuch, sich der Natur des Textes anzunähern.
Ausgangspunkt war die Frage 'What is text, really?' und eine erste Antwort gleich vorweg 'A text is a OHCO.' [Renear et al., 1993] OHCO ist ein Acronym für Ordered Hierarchy of Content Objects, was soviel bedeutet wie, ein Text ist eine Anordnung von hierarchisch angeordneten Objekten einzelner Textteile. Problembestimmend war für diese Diskussion die Tatsache von *overlapping objects*, denn nach der OHCO-These kann ein Textelement immer nur einer logischen Einheit zugehören, nicht mehreren gleichzeitig. Das hat unter anderem zu immer genauer formulierten Versionen von OHCO-Thesen geführt, ich zitiere hier nur die letzte und dritte Version OHCO-3:

> For every distinct pair of objects x and y that overlap in the structure determined by some perspective P(1), there exists diverbatim perspectives P(2) and P(3) such that P(2) and P(3) are sub-perspectives of P(1) and x is a object in P(2) and not in P(3) and y is an object in P(3) and not in P(2). [Renear et al., 1993]

Die Textelemente (content objects) werden im Abschnitt 4.2.1 bereits als Strukturelemente ausgemacht und bei SGML mittels generischen Markup ausgezeichnet. Ich lasse mich hier nicht allzu genau auf diese Textdiskussion ein, ernte aber sehr wohl die Früchte dieser Debatte und setze sie für meine Zwecke ein. Einmal lässt sich Text bei Interesse an seiner Struktur und seiner Modellierung anhand der Struktur sehr brauchbar als OHCO behandeln. Weiters kommt es auf die jeweilige Sichtweise an, wie Textobjekte lokalisiert werden. Das heißt, divergente Sichtweisen erzeugen divergente OHCOs, jedoch können verschiedene

Sichtweisen auf einer Metaebene stets aufgelöst werden. Diese Einsicht ist, vor dem Hintergrund plausibel, dass es keine fix implementierte Strukturhierarchie von Objekten gibt, sondern eine solche immer erst ensteht, wenn jemand eine Betrachtung einer Analysesituation auf den Text legt.
Für diese Untersuchung ist es zweckmäßig, Texte wie OHCOs zu behandeln, denn mit SGML und weiteren sie flankierenden Standards ist diese Art der Textmodellierung das zum heutigen Zeitpunkt beste Verfahren mit elektronischem Text umzugehen, was unten noch zu zeigen sein wird.

5 offene Standards

Ein Standard lässt sich allgemein begreifen als eine hinreichend genaue Beschreibung, wie Handlungen regelmäßig ablaufen sollen. Standards sollen Handlungen vereinfachen, weil das Reflektieren über das 'Wie etwas getan werden soll' wegfällt. Im zwischenmenschlichem Zusammensein sind Anstandsformeln der Höflichkeit mit Standards vergleichbar, es ist 'vorgeschrieben' und den meisten Männern der Industrieländer bewußt, der Dame die Tür zu öffnen usw.
Eine genauere Beschreibung von Standards könnte so lauten:

> Standards are documented agreements containing technical specifications or other precise criteria to be used consistently as rules, guidelines, or definitions of characteristics, to ensure that materials, products, processes and services are fit for their purpose. [ISO, 2002]

Für auf Gewinn ausgerichtete Unternehmen ist es in einem besonderem Maße wichtig, Arbeits- und Lernaufwand zu rationalisieren und eine möglichst hohe Planungssicherheit für zukünftige Investitionen sicherzustellen. Normalerweise wird das erreicht, indem Standards eingeführt werden, für alle Beteiligten die im standardisierten Bereich Kompatibilität der Subsysteme bereitstellen. Auch Standards ändern sich von Zeit zu Zeit, wenn neue Erwartungen mit den bisherigen Mitteln nicht erreicht werden können, muss der Standard abgeändert werden. Wann und wie das passiert, weiß bei propritären Standards nur der Lizenznehmer, dieser Umstand bedeutet für andere Teilnehmer der Standardimplementierungen ein schwer zu kalkulierendes Risiko.
Das ist ein Grund, warum heute viele eigentlich konkurrierende Unternehmen in sogenannten strategischen Partnerschaften in gewissen Bereichen zusammenarbeiten. Bei diesen Situationen handelt es sich meist um De-facto-Standards, wie im Falle eines PC-Betriebssystems, den GSM und UMTS-Standard für die Mobilkommunikation usw. Manche Standards entstehen (als Standard) völlig ungeplant und setzen sich aufgrund ihrer praktischen Überlegenheit weltweit durch, ohne dass dies von vornehinein einkalkuliert wurde.
Die in dieser Arbeit behandelten Standards sind alle offener Natur, das heißt sie wurden eingeführt um allen Interessierten - in unserem Falle - den technischen Rahmen von Informations- und Kommunikationssystemen und seine Kompatibilität und Interoperabilität gleichermaßen zur Verfügung zu stellen.
Andere Zugänge zu Standardisierungsprozessen als die oben beschriebenen sollen hier betrachtet werden. Einer sind Institutionen, die gegründet wurden um Standards zu entwickeln. Die folgenden Organisationen können als maßgeblich betrachtet werden für offene internationale und vielfach etablierte Standards:

- International Organization for Standardization (ISO)
- International Electrotechnical Commission (IEC)
- American National Standards Institute (ANSI)
- World Wide Web Consortium (W3C)

- Internet Engineering Task Force (IETF)

Zwei solcher Institutionen möchte ich hier etwas genauer behandeln. Beide sind in einem herausragendem Maße relevant für das Thema dieser Arbeit, alle fünf sind direkt oder indirekt mit ihren Standards für das reibungslose Zusammenspiel aller Komponenten verantwortlich.

5.1 International Organization for Standardization

Die International Organization for Standardization (ISO) ist eine weltweite Föderation von über 140 Ländern, wobei sich in jedem Land eine ISO-Zweigstelle befindet. ISO wurde 1947 gegründet und besitzt heute 13.700 Standards.

> The mission of ISO is to promote the development of standardization and related activities in the world with a view of facilitating the international exchange of goods and services, and to develop cooperation in the sphere of intellectual, scientific, technological and economic activity. [ISO, 2002]

ISO's Arbeitsprozeß resultiert in internationalen Vereinbarungen, die schließlich als internationale Standards publiziert werden. Standards gibt es für die verschiedensten Dinge und Belänge, von der Stärke und Abmessungen von Telephon- und Bankomatkarten (ISO-Standard 7816) bis zu der von Kondomen (ISO-Standard 4074). Viele ISO-Standards, die weltweit zum Einsatz kommen, sind uns mehr oder minder unbekannt, wobei sich wiederkehrende gewerbliche Handlungen meist entsprechenden Standards verpflichten.

5.1.1 Standardisierungsprozeß der ISO

Internationale Standards der ISO werden durch die ISO technischen Komitees (technical committees, TC) und ihrer Subkomitees (subcommittees, SC) entwickelt, dafür durchlaufen sie sechs Stufen:

- Proposal stage
- Preparatory stage
- Commitee stage
- Enquiry stage
- Approval stage
- Publication stage

Diese Reihenfolge ist jedoch nicht zwingend, wenn bspw. ein von einer anderen Institution entwickelter Standard in den Standardisierungsprozeß der ISO aufgenommen werden soll, kann er im ,,Fast-track-procedure" einige Stufen überspringen.

Alle Standards werden mindestens alle fünf Jahre durch die verantwortlichen TCs/SCs überprüft, in der entscheidet eine Mehrheit der TC/SC, ob der jeweilige Standard wieder bearbeitet wird oder gleich bleibt.

5.2 World Wide Web Consortium

Das World Wide Web Consortium (W3C) wurde 1994 am Massachusetts Institute of Technology (MIT) in Boston gegründet mit dem selbstgesetzten Ziel: *Leading the Web to its Full Potential...* Direktor ist bis heute Tim Berners-Lee, der Erfinder des WWW. Das W3C besitzt drei Hosts: das MIT, das Institut National de Recherche en Informatique et en Automatique (INRIA) und Keio, die Universität von Tokio; zehn nationale Büros und über 420 Mitglieder.
Hauptprodukte des W3C sind sogenannte *Recommendations*, die im Internet als Standard für Protokolle und Anwendungen gelten. Ein die Arbeit des W3C beeinflussendes Ziel ist es, möglichst hohen Konsens der Recommandations zu erreichen, was durch ein festgesetztes Verfahren (Recommendation Process) bestimmt wird.

5.2.1 Recommendation Process im W3C

Das W3C bildet zu verschiedenen Gebieten spezielle Arbeitsgruppen, sofern dies von den Mitgliedern gewünscht wird. Die Arbeitsgruppen produzieren Arbeitsentwürfe (Working Drafts) für ihre jeweiligen Projekte, die als endgültig angesehener Stand der Spezifikation öffentlich als Empfehlungsanwärter (Candidate Recommendation) vorgestellt werden. In diesem Verfahren soll vor allem Außenstehenden die Möglichkeit der Prüfung und Implementierung gegeben werden, um eventuelle Probleme oder Ungereimtheiten so früh als möglich an die Arbeitsgruppe heranzutragen.
Nach Ablauf einer Begutachtungsfrist und eventueller Überarbeitung ist der nächste Schritt ein Empfehlungsvorschlag (Proposed Recommendation), dieser wird schließlich nach wiederholter Überarbeitungsphase im Status einer Empfehlung (Recommendation) erhoben.
Eine Empfehlung ist nicht so zwingend wie ein offizieller Standard, da es keinen Zertifizierungsprozesse und keine Lizenzvereinbarungen gibt. Somit bestehen auch keine rechtliche Möglichkeiten bei Nichteinhaltung einer Implementierung einer Recommendation.

6 primäre Strukturmodellierung

Unter primärer Srukturmodulierung lässt sich weitgehend alles verstehen, was in vorangegangenen Abschnitten über XML/SGML bzw. Markup gesagt wurde. Die direkte Textstrukturierung bildet das Grundinstrumentarium von XML und SGML und wird heute auch hauptsächlich in der Literatur behandelt. Hier soll es primär um XML gehen, auch das nächste Kapitel mit seiner sekundären Strukturmodulierung wird sich hauptsächlich auf XML als Darstellungsweise beschränken. Wäre eine ausschlaggebende SGML-Eigenart gegenüber XML zu erwähnen, werde ich das natürlich tun.

6.1 XML

Die eXtensible Markup Language (XML) beschreibt eine Klasse von Datenobjekten, kurz XML-Dokumenttypen und ist eine eingeschränkte Form von SGML, dadurch sind konforme XML-Dokumente immer auch konforme SGML-Dokumente, jedoch nicht umgekehrt. Anders als SGML ist XML ein vom World Wide Web Consortium (W3C) entwickelter Standard, dessen Arbeitsgruppe von Jon Bosak, Sun Microsystems, geleitet wurde.
XML folgte ganz klaren Entwurfszielen, eines war die einfache Nutzbarkeit im Internet, ähnlich wie bei HTML, das schnell über Netzwerke transportiert werden kann, ohne große Bandbreiten zu benötigen. Eine Eigenschaft, die SGML nicht besitzt.
Im Gegensatz zu HTML kann XML nicht nur Daten, sondern auch den logisch-strukturellen Zusammenhang dieser Daten transportieren. Mittels Markup wird einem gewissen Datenobjekt ein Name zugewiesen, aber wie auch bei SGML sagt das nichts über das Layout des Dokuments aus, und genau dieser Umstand grenzt HTML hier klar ab, dessen Standard sowohl generische als auch layoutorientierte Auszeichnungen besitzt.
Für die Präsentation von XML in beispielsweise einem Webbrowser ist deshalb immer eine zweite Methode zur layoutorientierten Darstellung notwendig, ähnlich wie das oben schon beschriebene Verfahren DSSSL. Aus diesem Grunde wird von XML oft als von einem reinem Datenformat für Datenaustausch gesprochen, dies ist aus einem bestimmten Blickwinkel auch vollkommen richtig, die volle Mächtigkeit zeigt XML jedoch erst mit seinen entwickelten oder noch in Entwicklung stehenden Satellitentechniken und teilweise schon vorhandenen standardisierten Dokumentstrukturbeschreibungen. Eine davon werde ich im nächsten Abschnitt genauer beschreiben.

6.1.1 Metasprache

Extensible Markup Language heißt auf Deutsch übersetzt soviel wie Erweiterbare Auszeichnungssprache, was aber so nicht ganz richtig ist, denn XML ist keine Auszeichnungssprache, sondern ein Regelwerk zur Schaffung von Auszeichnungssprachen, eine sogenannte Metasprache.

Die Auszeichnungssprachen, die mit XML erstellt werden können, haben verschiedene gleichrangige Bezeichnungen:

- XML-Anwendungen
- XML-Dokumenttypen
- XML-Vokabulare

In der Praxis wird meist kurz XML gesagt, ein Umstand, der vermieden werden sollte, solange Eindeutigkeit nicht vorliegt. Der Begriff XML-Anwendung ist ebenfalls mehrdeutig, da hierunter auch Programme verstanden werden können, die XML in irgendeiner Weise verarbeiten. Ich möchte mich auf die zweite Variante beschränken und diese mit XML-Dokument sogar etwas kürzen.

6.1.2 Markup

XML-Dokumente bestehen aus einer Folge von Zeichen und Markups. Das Markup befindet sich inmitten der Zeichen oder des elektronischen Textes und zeichnet bestimmte Bereiche der Zeichen mit einer Eigenschaft aus, die durch das Markup festgelegt wird. Dadurch können wir Textstücke beschreiben oder für eine gewisse Art der Verarbeitung gesondert kennzeichnen.
Das Markup wird als Zeichenkette auftretend als Tag bezeichnet. Ein Charakteristikum von XML ist, dass jedes *Start-Tag* mit einem zugehörigen *Ende-Tag* abgeschlossen werden muss. Dazwischen können textuelle Inhalte plaziert werden oder weitere Tags auftreten. Das bedeutet wir können zunächst zwei verschiedene Ebenen unterscheiden auf denen XML-Dokumente direkt strukturiert werden können. Einmal die Ebene der konkreten Daten oder Textfragmenten und die Ebene der abstrakten Einheiten, die den Daten Funktionen zuordnen oder sie gruppieren.
Die abstrakten Einheiten werden in XML als Elemente bezeichnet, die wiederum zwei verschiedene Typen besitzen können:

- Daten-Elemente
- Container-Elemente

Beide Arten können auch als Mischform auftreten und eine Differenzierung ist folgendermaßen begründbar, dass Daten-Elemente die konkreten Daten enthalten und Container-Elemente selbst wiederum Elemente enthalten, die Daten- oder Container-Elemente sein können. Mit dem Konzept der Container-Elemente ist es möglich, komplexe Strukturen in die Daten-Elemente zu modulieren. Ein dritter Typ sind die sogenannten leeren Elemente.
Alle Konstrukte, die in XML mit einer spitzen Klammer oder mit dem kaufmännischen Und beginnen, bezeichnen wir als Markup, alle anderen als Zeichendaten, Daten oder Text. Das Programm, das diese Unterscheidung formal zu treffen hat, um XML-Dokumente zu verarbeiten heißt Parser. Ein Parser ist ein Teil des

Prozessors, genauer des XML-Prozessors, dessen Aufgabe darin besteht XML-Markup und Daten in eine Folge von Tokens zu zerlegen, um diese dann weiter zu verarbeiten. Ein Token kann ein Start-Tag, ein Datenelement oder ein anderer Teil von Markup sein, der Dokumentteile auszeichnet. Diese Folge von Tokens wird danach an die nächste Komponente des XML-Prozessors weitergereicht.
Betrachten wir ein erstes Beispiel mit folgendem Ausschnitt eines möglichen XML-Dokuments:

```
<Vorname>Ludwig</Vorname>
```

Dabei handelt es sich um folgende unterschiedliche Tokens mit dazugehöriger Bedeutung:

`<Vorname>`	Start-Tag des Elements Vorname
`Ludwig`	Inhalt des Elements Vorname
`</Vorname>`	Ende-Tag des Elements Vorname
`<`	Öffnendes Begrenzungszeichen für das Start-Tag
`</`	Öffnendes Begrenzungszeichen für das Ende-Tag
`Vorname`	Elementname
`>`	Schließendes Begrenzungszeichen für ein Tag

Jedes Element kann über Eigenschaften verfügen, die mittels Attribute folgendermaßen realisiert werden können:

```
<name art="first">Ludwig</name>
```

`art`	Attributname
`=`	Zuweisungsoperator
`"`	Begrenzungszeichen für die Zeichenkette
`first`	Attributwert

Elemente können auch über mehrere Attribute verfügen, diese werden dann hintereinander mit Leerraum getrennt, ebenso innerhalb des Start-Tags aufgelistet.

6.1.3 Namen

Namen sind ein elementarer Bestandteil von XML. Elemente erhalten einen Namen, durch den es möglich wird, in ihnen enthaltenen Daten zu klassifizieren und zu beschreiben. Logischen Strukturen werden Elementtypnamen zugewiesen, wiederverwendbare Daten bekommen Entity-Namen und bestimmte Elemente erhalten IDs.
Bei der Vergabe von Namen ist bei XML gewissen Konventionen zu folgen, so müssen Namen mit einem Buchstaben oder einem oder mehreren Interpunktionszeichen beginnen Fortsetzen können sich Namen dann mit Buchstaben, Ziffern, Gedankenstrichen, Unterstrichen, Doppelpunkten oder Punkten. Namen

die mit den drei Buchstaben 'xml' beginnen, gelten als reserviert für bestimmte systemimanente Routinen. Leeräume dürfen nicht Bestandteil eines Namens sein.

6.1.4 Literaldaten

Im Gegensatz zu Namen sind Literaldaten die Daten, die innerhalb der Start- und Ende-Tags oder als Attributwert vorkommen können. Also alles was nicht unter die Kategorie Markup fällt, sondern gemeinhin als eigentlicher Text bezeichnet wird.

6.2 logischer Aufbau eines Dokuments

Der Aufbau eines XML-Dokuments gliedert sich einmal in einen Prolog und in eine Dokument-Instanz, wobei die Dokument-Instanz dasjenige bezeichnet, das sich bis jetzt als Markup und Text gezeigt hat. Die genaue Bezeichnung rührt daher, dass es eine Dokumenttyp-Definition (DTD) und ein XML-Dokument gibt, das in die Definition fällt und somit zur Dokument-Instanz wird.
Der Prolog lässt sich weiter aufgliedern in eine XML-Deklaration und eine Dokumenttyp-Deklaration.

6.2.1 XML-Deklaration

XML-Dokumente beginnen in der Regel mit der XML-Deklaration, die die Rahmenbedingungen des Dokuments angibt und die Versionsnummer enthält. Die XML-Deklaration ist an ihrer Form eindeutig von anderen Konstrukten zu unterscheiden:

```
<?xml version="1.0" encoding=UTF-8 standalone="yes" ?>
```

In diesem Beispiel wird davon ausgegangen, dass es sich um die Version 1.0 des XML-Standards handelt, die Kodierung UTF-8, eine Submenge von Unicode, verwendet wird und dass das XML-Dokument für sich selbst stehen kann, also von keinen externen Entitäten abhängt. Wobei noch zu sagen ist, dass alle Angaben in der XML-Deklaration optional sind, weil es sich um voreingestellte Standardwerte handelt.
Diese Eigenart eines XML-Dokuments wird auch Processing Instruction (PI) genannt und vom Parser einfach an die Anwendung weitergegeben ohne darauf zu reagieren.

6.2.2 Dokumenttyp-Deklaration

Bevor das erste Element des Dokuments erscheint, erfolgt die Dokumenttyp-Deklaration, die dafür sorgt, dass die Struktur der Dokument-Instanz gewissen Regeln enstpricht, die die DTD festlegt. Die Dokumenttyp-Deklaration verweist nun entweder auf eine externe DTD, wie es bei HTML üblich ist, oder die DTD befindet sich selbst intern im Rahmen der Dokumenttyp-Deklaration.
Die Anweisung für eine Dokumenttyp-Deklaration hat folgende allgemeine Form:

```
<!DOCTYPE name externer.zeiger [interne.untermenge]>
```

Der externe Zeiger wäre bei HTML 4.01:

```
PUBLIC "-//W3C//DTD HTML 4.01//EN"
```

Er verweist auf einen öffentlichen Deskriptor der DTD von HTML 4.01. Die eckigen Klammern stellen die Möglichkeit zur Verfügung eine DTD als interne Untermenge anzugeben, ohne dabei auf eine externe zuzugreifen.

6.2.3 Dokument-Instanz

Die Dokument-Instanz beinhaltet nun alle restlichen Elemente, Attribute, Entitäten und Zeichenfolgen, die Bestandteil des XML-Dokuments sind. Dieser Bereich des Dokumentes ist es auch, der sich als Baumform darstellen lässt. Wenn der Parser diesen Teil des Dokuments bearbeitet spricht man auch von Document Object Model (DOM), eine Repräsentationsform der Dokument-Instanz als Baumform im Speicher des Rechners. XML-Parser die nach dem DOM vorgehen, geben der Anwendung oder dem Programmierer die Möglichkeit das geparste Dokument mit Methoden zu bearbeiten, die auf die Strukturierung von Bäumen ausgelegt sind und sehr mächtig sein können.

6.3 DTD-Syntax

Der grundlegende Aufbau von XML-Dokumenten folgt dem oben beschriebenen Schema, jedoch haben wir noch keinerlei Struktur modelliert, obwohl das der eigentliche Zweck dieses Abschnitts ist. Wie schon erwähnt erfolgt die genauere Beschreibung des in der Dokument-Instanz vorkommenen Markups in der DTD. Hier soll diese nun etwas genauer betrachtet werden, um das Konzept zu verstehen, wie eine DTD Struktur und Ordnung in einem XML-Dokument organisiert.
Mit der DTD können Elementarten definiert werden, weiter kann festgelegt werden, welche Unterelemente in ihnen in welcher Anordnung und Zahl enthalten sein können. Ebenso können die schon kurz behandelten Attribute und deren Datentypen mit Attributwerten bereits voreingestellt werden. Die nun folgendenen Deklarationen können direkt in interne Untermengen des Prologs eingegeben oder extern abgelegt werden.

6.3.1 Elementdeklaration

Die Syntax der Elementdeklaration sieht für ein konkretes Beispiel so aus:

```
<!ELEMENT autor (#PCDATA)>
```

Diese Definition sagt aus, dass ein Element vom Typ `autor` als Inhaltsmodell Daten vom Typ `PCDATA` enthalten kann. Das Zeichen `#` zeigt lediglich an, dass es sich bei `PCDATA` um ein vordefiniertes Schlüsselwort handelt. Durch den Namen ist es nun möglich verschiedene Elementtypen zu definieren.

Der Datentyp kann noch andere Werte außer `PCDATA`[7] annehmen, zum anderen auch `EMPTY`, das würde bedeuten, dass es sich um ein leeres Element handelt. Weiters gibt es den Typ `ANY`, der beliebig viele Elemente jeglicher Art in einer nicht weiter spezifizierten Reihenfolge und Anzahl beinhalten kann. `CDATA` würde bedeuten, dass der Inhalt des Elements nicht geparst würde.
XML hätte nicht die Möglichkeit einer direkten Strukturmodellierung, wäre nicht die Organisation der Reihenfolge von Elementen durch die DTD gegeben. Im Prinzip legt das Inhaltsmodell fest, in welcher Weise im Container-Element andere Elemente zueinander in Beziehung gesetzt werden und welchen Status die einzelnen Elementtypen besitzen. Hierzu gibt es angelehnt an Reguläre Ausdrücke (engl.: regular expressions) drei Operatoren:

`A` A muss genau einmal auftreten (obligatorisch).
`A?` A kann einmal auftreten, oder aber auch ausgelassen werden (fakultativ).
`A+` A muss mindestens einmal, kann aber beliebig oft auftreten.
`A*` A kann einmal oder beliebig oft auftreten, kann aber auch ausgelassen werden.

```
<!ELEMENT titel (#PCDATA)>
<!ELEMENT stadt (#PCDATA)>
<!ELEMENT verlag (#PCDATA)>
<!ELEMENT jahr (#PCDATA)>
```

Hier wären nun vier Elemente mit bibliographischen Namen definiert, die jedoch keinerlei Strukturdefinition beinhalten, das geschieht jedoch mittels Container-Element:

```
<!ELEMENT buch (autor+, titel, verlag?, jahr?)>
```

Durch diese Deklaration kombiniert mit den Opertatoren ist es nun möglich, mehrere Autoren für ein Buch anzugeben, der Titel ist obligatorisch und die Verlags- und Jahresangabe ist nicht zwingend, kann also auch ausgelassen werden.
Für Beziehungen unter Elementen in Container-Elementen stehen zwei Konnektoren zur Verfügung:

`A, B` B folgt auf A
`A | B` A oder B

Eingesetzt können komplexere Strukturen erzeugt werden, wie z.Bsp.:

```
<!ELEMENT buch  (autor+,  titel,
(verlag? , jahr?) | (jahr?, verlag?))>
```

[7] PCDATA steht für 'parsed-character data' und bedeutet, dass der Inhalt dieser Datenelemente kein weiteres Element außer Text sein darf. Dies wird vom XML-Parser überprüft.

Diese Deklaration würde bedeueten das Container-Element `buch` müsste um der DTD zu genügen ihre Daten-Elemente folgend anordnen; Zuerst kommen mehrere oder nur ein Autor, dann der obligatorische Titel, weiter, dass die fakultativen Elemente Verlag und Jahr in beliebiger Reihenfolge auftreten können, also entweder Jahr auf Verlag oder Verlag auf Jahr.

6.3.2 Attributdeklaration

Attributdeklarationen bestimmen, welches Element welche Attribute haben kann, welchen Datentyp diese Attribute besitzen und ob und welche Vorgabewerte eingestellt werden sollen. Weiter ob die Angabe eines Attributwertes obligatorisch oder fakultativ sein soll. Die allgemeine Form einer Attributdeklaration lautet:

```
<!ATTLIST element.name (attribut.definition)>
```

Konkret könnten wir das Geschlecht einer Person mittels Attributdeklaration modellieren:

```
<!ELEMENT person (vorname+, nachname)>
<!ATTLIST person geschlecht (m | f) #REQUIRED>
```

Insgesamt sind drei Klassen von Attributtypen möglich:

1. String-Attribute, die aus beliebig vielen Zeichendaten bestehen.
2. Token-Attribute, deren Wert aus ein oder mehreren für XML relevanten Tokens bestehen.
3. Auzählungsattribute, deren Wert einer aus einer deklarierten Liste sein muss.

String Attribute könnten folgend deklariert werden:

```
<!ATTLIST buch isbnr CDATA>
```

Die Werte von String-Attributen sind Zeichenketten, wobei jedes Attribut, das in einem XML-Dokument ohne DTD vorkommt automatisch als String-Attribut behandelt wird.
Token-Attribute sind sehr mächtig, deshalb folgt eine übersichtlich Aufstellung aller ihr enthalten Datentypen:

`ID` Dieses Attribut dient als Identifikator für ein Element. Ein `ID`-Wert muss den Standardnamensregeln von XML entsprechen und eindeutig innerhalb eines Elements sein. Weiter ist `REQUIRED` oder `IMPLIED`[8] als Vorgabewert vorgeschrieben.

[8]Dieses Schlüsselwort bedeutet, dass der Parser das Fehlen eines Attributwertes ignoriert und der jeweiligen Aplikation das weitere Vorgehen überlässt. Hingegen ist bei `REQUIRED` die Angabe eines Attributets verpflichtet, der Parser würde das Fehlen also nicht einfach übergehen.

`IDREF` Das `IDREF`-Attribut ist ein Zeiger auf ein `ID`-Attribut eines anderen Elements, dessen Werte übereinstimmen müssen, somit lassen sich Referenzen modellieren. Auch dieses Attribut muss den Standardnamensregeln entsprechen.

`IDREFS` Der Wert dieses Attributs besteht aus mehreren Referenzen auf `ID`-Elemente, die durch Leerzeichen voneinander getrennt sein müssen.

`ENTITY` Das `ENTITY`-Attribut ist ein Zeiger auf ein externes Entity, das in einer DTD-Untermenge deklariert wurde.

`ENTITIES` Der Wert dieses Attributs besteht aus einem oder mehreren `ENTITY`-Typwerten, die wieder durch Leerzeichen voneinander getrennt werden.

`NMTOKEN` Der Wert dieses Attributs ist ein Name-Token-String, der aus einer beliebigen Kombination von Namenszeichen besteht.

`NMTOKENS` Dieser Wert besteht aus einem oder mehreren durch Leerzeichen getrennten `NMTOKEN`-Typwerten.

Aufzählungsattribute wurden oben bereits verwendet und werden durch eine deklarative Liste möglicher Werte charakterisiert, die jeweils einen gültigen Namen-Token (`NMTOKEN`) darstellen müssen. Auch hier können Vorgabewerte bestimmt werden.

6.3.3 Entity-Deklaration

Entities sind die kleinste physiche Einheit eines XML-Dokuments und werden in [Ray, 2001] als *Placeholders for Content* beschrieben, und genau das sind sie im Grunde auch. Entities werden prinzipiell zur flexibleren Dokumentorganisation verwendet und können von einfachen Abkürzungen innerhalb eines Dokuments bis zur Einbindung ganzer XML-Dokumente in einem XML-Haupt-Dokumente verwendet werden. Die einfachste Art der Entities ist das interne-Entity, das wie folgend definiert werden kann:

```
<!ENTITY dtd "document type definition">
```

Würde als Zeichenkette in einem XML-Dokument `&dtd` stehen, würde der Parser diese Abkürzung durch die Langform 'document type definition' austauschen. Die internen Entities grenzen sich von den externen Entities dadurch ab, da ein internes Entity in sich völlig abgeschlossen ist und kein eigenes Speicherobjekt benötigt. Externe Entities hingegen befinden sich an einer der Deklaration separaten Position und werden mittels Verweis referenziert.
Binäre Entities werden nicht geparst und sind primär dafür gedacht Binärdateien wie Musikfiles, Grafikdateien usw. in ein XML-Dokument einzubinden. Hinzu kommt, dass eine das binäre Format verarbeitende Applikation angegeben werden kann, um die binären Entities auch richtig darzustellen. Wird ein binäres Entity deklariert muss auf jeden Fall eine *Notation* angegeben werden. Eine Notation beschreibt das Dateninhaltsmodell nichtgeparster Daten und gibt

somit an, wie die binären Daten interpretiert werden sollen.
Parameter-Entities unterscheiden sich von den übrigen Entity-Arten, dass Verweise auf das Parameter-Entity nur innerhalb einer DTD erfolgen kann. Der hauptsächliche Verwendungszweck ist daher der der Rationalisierung und der Übersichtlichkeit durch Abkürzungen. Parameter-Entities können sowohl intern als auch extern realisiert sein.

6.4 well-formed documents

XML-Dokumente werden prinzipiell in zwei Gruppen geteilt, in wohlgeformte (engl.: well-formed) und in gültige (engl.: valid) Dokumente. Wobei Wohlgeformtheit eine schwächere Eigenschaft als Gültigkeit bzw. Validierbarkeit darstellt, da auch nicht gültige XML-Dokumente wohlgeformt sein können.
Im Gegensatz zu XML wird bei SGML-Anwendungen Validierbarkeit zum absoluten Maßstab erhoben. Dieser Umstand lässt sich leicht durch die Verbreitung des Internet erklären, denn gerade online-Anwendungen können genauso mit wohlgeformten Dokumenten (semistrukturierten Daten) sinnvolle Verarbeitungstechniken erreichen. [Lobin, 2000][70]
Formal kann von einem XML-Dokument behauptet werden, dass es wohlgeformt ist, wenn folgende Bedingungen eintreffen:

- Es enthält mindestens ein Element.
- Es besitzt ein ausgezeichnetes Wurzel- oder Dokumentelement, das alle übrigen Elemente beinhaltet.
- Es verfügt über einen einleitenden Prolog
- Alle auftretenden Elemente sind korrekt ineinander verschachtelt.
- Die Namen in den zusammengehörigen Start- und Ende-Tags stimmen überein.
- Attributnamen sind innerhalb eines Elements eindeutig.
- Entities werden deklariert, bevor sie benutzt werden.
- Die XML-Syntax nach [Bray et al., 1998] wird eingehalten.

6.5 valid documents

Jede Markup-Sprache, auch HTML, beruht auf einer DTD, die Autoren eines Dokuments mehr oder weniger zwingt, gewisse syntaktische Regeln und Strukturmodelle zu verwenden. Wenn maschinelle Verarbeitung gewährleistet werden soll, ist eine Vorwegnahme von restriktiven Regeln fast nicht zu umgehen. Im WWW mit HTML ist es einfach Dokumente zu erstellen, die DTD für HTML ist dementsprechend mit sehr wenigen Einschränkungen erstellt worden. Mitunter sicherlich ein Grund ihrer raschen Verbreitung, andererseits ergeben sich

gerade daraus erhebliche Mängel, die nun mit XML-basierten Markup-Sprachen ausgeglichen werden sollen.
Eine DTD legt also fest, welche Elemente in einem XML-Dokument vorhanden sein müssen, welche Elemente vorhanden sein können und wie sie zu anderen Elementen in Beziehung stehen sowie welche Attribute sie haben können oder müssen. Eine DTD charakterisiert somit eine Markup-Sprache und wird mit XML beschrieben. Weiter ist XML demnach eine Klasse von verschiedensten Sprachen, die jeweils durch eine eigene DTD definiert sind. HTML wäre eine dieser Sprachen, andere wichtige aber weniger bekannte sind MathML als mathematische Sprache, BSML (Bioinformatic Sequence Markup Language), das SOAP (Simple Object Access Protocol), die WML (Wireless Markup Language), das XML-Rpc, das Remote Procedure Call-Aufrufe und deren Formate definiert oder VXML (Visual XML) zur Beschreibung und Veröffentlichung von Web-Sites. Eine erschöpfende Aufzählung würde den Rahmen sprengen, auch werden in regelmäßigen Abständen neue XML-Vokabulare standardisiert.
Mit den bereits beschriebenden Möglichkeiten der DTD-Syntax kann nun eine Aussage darüber gemacht werden, ob ein XML-Dokument gültig (engl.: valid) bezüglich seiner DTD ist. Darüber enscheidet im Grunde eine Reihe von syntaktischen Bedingungen, die unten aufgelistet werden. Hiermit lässt sich sagen, ein XML-Dokument ist gültig, wenn es nachstehende Bedingungen erfüllt:

- Wenn die Dokumentinstanz wohlgeformt ist und sie in ihrem Aufbau der DTD mit allen eingebundenen Untermengen gemäß ihrer Einschränkungen entspricht.

6.6 Satellitentechnologien

XML als Regelwerk für die Beschreibung von Markup-Sprachen und als Instrument direkter Textstrukturierung wurde bereits vorgestellt. Auch der enorme Nutzen, der aus XML gezogen werden kann, will veranschaulicht sein, jedoch gibt es andere Spezifikationen, die XML erweitern bzw. die wirkliche Mächtigkeit von XML erst garantieren. [Ray, 2001]
Die Ganzheit der Satellitentechnologien ist für eine Person kaum mehr überschaubar und wächst ständig an, deshalb möchte ich hier bloß die Techniken behandeln, dessen Verständnis für die im nächsten Abschnitt erläuterten XML Topic Maps notwendig sind.

6.6.1 Request for Comments

Request for Comments: 2396, kurz RFC 2396, ist ein Standard der Internet Engineering Task Force (IETF), der den Aufbau des Uniform Resource Identifiers (URI) beschreibt.
URI ist ein der Uniform Resource Locators (URL) übergeordnete Beschreibung, wie auf Internetresourcen zugegriffen werden kann. URL wurde unter RFC 1738 standardisiert und mit RFC 2396 erweitert. [Berners-Lee et al., 1998] Der grundsätzliche Aufbau der URI ist identisch der URL, jedoch werden zusätzliche Möglichkeiten beschrieben, wie die Angabe einer Mailadresse.

6.6.2 XML Names

XML Names [Bray et al., 1999] oder XML-Namesräume (engl.: Namespaces) bieten die Möglichkeit Element- und Attributnamen eindeutig zu benennen. Hierzu werden die Element- und Attributnamen mit Namensräumen verknüpft, die durch einfache URI-Verweise identifiziert werden.
Die Überlegung entspringt dem Umstand, dass, wenn in XML-Dokumenten unterschiedliches Markup enthalten ist, das teilweise oder vollständig identische Tags verwendet, würde ein das Dokument verarbeitender Parser nicht zwischen zwei verschiedenen Markup-Konzepten unterscheiden können. Umgangen werden können derartige Namenskonflikte mittels XML Namespaces. Ein XML-Namensraum ist eine Zusammenstellung von Namen, welche durch einen URI-Verweis eindeutig identifiziert werden kann. URI-Verweise werden dann als identisch angesehen, wenn sie Zeichen für Zeichen genau gleich sind.

6.6.3 XMLBase

Mit XMLBase ist es, angelehnt an das HTML-Element `BASE`, möglich, einen Base-URI für ein Dokument anzugeben, um relative URIs zu externen Quellen aufzulösen. Dieses Verfahren ermöglicht somit die Basis einer URI, den Interfacenamen, wie z.Bsp.: www.philo.at einmalig anzugeben, und im weiterer Folge nur relativ mittels Resourcenbezeichner, wie 'wittgenstein.html' die Quellen zu referenzieren. XLink bezieht sich normativ auf XMLBase zur Interpretation relativer URI-Verweise. Die Syntax von XMLBase besteht allgemein aus folgendem XML-Attribut [Marsh, 1999]:

`xml:base`

6.6.4 XLink

Die XML Linking Language (XLink, [DeRose et al., 2001]) definiert Referenzierungskonstrukte, die XML unterliegen, ähnlich wie die unidirektionalen Hyperlinks von HTML. XLink ist jedoch funktional erweitert und es gibt in XML keine vordefinierten Tags, somit auch kein Tag für XML Links. Das bedeutet, dass bei XLink beliebige Elemente mit einer Link-Funktion ausgestattet werden können, dies geschieht mittels eines besonderen Attributes, das folgende Form aufweist:

`xlink:type`

Weitere Attribute dieses Standards sind:

`xlink:href` Nimmt als Attributwert die URL des Zielobjekts auf.

`xlink:show` Damit kann bestimmt werden, ob die Zielresource in das aktuelle Dokument eingefügt werden soll (`embed`) oder es ersetzt werden soll (`replace`), oder mittels (`new`) eine neue Ansicht erzeugt werden soll.

`actuate` Mit `onLoad` wird das Zielobjekt automatisch angezeigt, mit `onRequest` erst durch Aktion.

Der Attributwert `type` legt die Art der Referenz fest, diese kann zwei Werte annehmen, `simple` und `extended`;[9] Da XTM mit `simple`-Links auskommt, möchte ich auf die `extended`-Links nicht weiter eingehen.
Mit einfachen Links wäre folgendes Konstrukt möglich:

```
<institut> xmlns:xlink="http://www.w3.org/1999/xlink/namespace/"
   xlink:type="simple"
   xlink:href="http://www.univie.ac.at/philosophie/"
Philosophie Wien
</institut>
```

In einer DTD müssen auch XLink-Attribute deklariert werden, wobei in unserem Fall die Werte `simple` und `http://www.w3.org/1999/xlink/namespace/` in der DTD aus Übersichtlichkeit durch das Schlüsselwort `FIXED` vorgegeben werden könnten.

[9] Mit diesem Linktyp ist es möglich auf mehrere Dokumente zu verweisen und multidirektionale Links anzulegen.

7 sekundäre Strukturmodellierung

Im vorangegangenem Abschnitt ist das Grundinstrumentarium einer direkten Strukturierung mittels XML vorgestellt worden. Somit sind die Voraussetzungen für weitere Betrachtungen, die auf die primäre Modellierung von Textstrukturen aufbauen, geschaffen.
Mit *sekundärer Strukturmodellierung* möchte ich den Umstand betonen, dass es sich hier um Strukturierungstechniken handelt, die auf die Ebene der primären Strukturierung aufsetzen und so eine allgemeinere, indirektere und abstraktere Möglichkeit zur Modellierung von Datenstrukturen bieten, als es mit der Möglichkeit der oben beschriebenen Mittel der Fall ist. Diese Ebene der Modellierung gewinnt immer mehr an Bedeutung und wird nach [Lobin, 2000] die Zukunft der XML/SGML-basierten Informationsmodellierung entscheidend mitbestimmen.
Eine Methode, materialisiertes Wissen auf einer Metaebene zu strukturieren, ist die der Topic Maps, zu deutsch: Themenlandkarten. Ein Verfahren, dass der Verlag S. Fischer zur Publikation der Großen kommentierten Frankfurter Ausgabe (GKFA) der Werke Thomas Manns einsetzt, um damit eine CD-ROM-Ausgabe neben der ebenso erscheinenden Buchausgabe zu erreichen. Der Verlag hält noch bis 2025 die Exklusivrechte an Thomas Manns Werk und möchte zwischen 2001 und 2015 eine 58-bändige Referenzausgabe der Werke, Briefe und Tagebücher Thomas Manns veröffentlichen, die in jährlichen Teilabschnitten erscheinen soll. Eine derartiger Zeitraum ist im Bereich der elektronischen Publikation eine halbe Ewigkeit und will gut geplant sein. Der Verlag hat hierbei sowohl bei der primären als auch bei der sekundären Modellierung der Datenbasis auf SGML/XML gesetzt, um auf jeden Fall eine zukünftige Verarbeitung der Daten zu garantieren. [Müller and Schmidt, 2000]

7.1 Topic Map Standard

Der Topic Map Standard, verabschiedet unter dem Kürzel ISO 13250 im Sommer 1999, wurde von der JTC1/SC34 entwickelt. JTC1 steht für Joint Technical Committee 1, einem Unterkomitee der ISO, das für Bereiche der Informationstechnologie zuständig ist. Ein weiteres Unterkomitee ist das SC34, das Subcommittee 34, deren Aufgabenbereich als 'Document description and processing languages' bezeichnet wird. [Biezunski et al., 1999]
ISO Topic Maps beziehen sich auf HyTime und Architectural Form Definition Requirements (AFDR, [AFDR, 1997]) und sind als Meta-DTD definiert. Da im weiteren Verlauf im Detail nur auf das Pendant in XML eingegangen werden soll, möchte ich diesbezüglich nur auf vertiefende Literatur verweisen: [Lobin, 2000], [Widham and Mück, 2000] und [Biezunski et al., 1999].
Charles Goldfarb bezeichnete Topic Maps einmal als GPS System, Global Positioning System für das Web. Eine der einfachsten Anwendungen der Topic Maps ist ein Stichwortverzeichnis bzw. Index, der darüber Auskunft weiß, wo sich indizierte Begriffe befinden. Wer hier an den elektronischen Text und seiner Möglichkeit der einfachen Suche von Textmustervergleichen (engl.: pattern

matching) denkt, ist nur teilweise auf der richtigen Spur. Denn ein Index im gedruckten Buch kann nicht nur die exakte Wortgleichheit mit Vorkommen der Seiten angeben, auch Mehrinformation wie 'siehe auch', oder Seitenangaben zu Begriffen, die selbst gar nicht aufscheinen, nur indirekt behandelt werden, können angegeben werden. Das bringt wiederum den Umstand zum Vorschein, dass die Erstellung eines Index bzw. einer Topic Map ein Verstehen des inhaltlichen Aspekts voraussetzt und somit ein rein maschinelles Modellieren heute nicht als effektiv erscheinen lässt.
Topic Maps sind Metastrukturen für eine Wissensorganisation und äquivalent zum Buchindex, aber in der Manier des elektronischen Textes mit Hyperlink-Konzepten sind Topic Maps ein multidimensionales Orientierungssystem basierend auf semantischer Auszeichnung. Durch die sekundäre Strukturierung, die Topic Maps zugrunde liegt ist es möglich strukturierte Sichtweisen auf unstrukturierte Datenbasen zu schaffen, liegt die Datenbasis bereits in strukturierter Form vor, kann eine vollautomatische Erstellung einer Topic Map erfolgen. [Biezunski, 2000]
In [Biezunski et al., 1999] sind mögliche Verwendungsweisen für Topic Maps aufgeführt:

- To qualify the content and/or data contained in information objects as topics to enable navigational tools such as indexes, cross-references, citation systems, or glossaries.
- To link topics together in such a way as to enable navigation between them. This capability can be used for virtual document assembly, and for creating thesaurus-like interfaces to corpora, knowledge bases, etc.
- To filter an information set to create views adapted to specific users or purposes. For example, such filtering can aid in the management of multilingual documents, management of access modes depending on security criteria, delivery of partial views depending on user profiles and/or knowledge domains, etc.
- To structure unstructured information objects, or to facilitate the creation of topic-oriented user interfaces that provide the effect of merging unstructured information bases with structured ones. The overlay mechanism of topic maps can be considered as a kind of external markup mechanism, in the sense that an arbitrary structure is imposed on the information without altering its original form.

7.2 XTM

Im Dezember 2000 veröffentlichte die TopicMaps.Org Authoring Group die XML Topic Maps (XTM) Spezifikation in der Version 1.0. Die TopicMaps.Org Authoring Group ist eine unabhängige Autorengruppe, angeführt von Michael Biezunski und Steven R. Newcomb, beides Autoren des ISO-Standards 13250. Seit Oktober 2001 ist XML Topic Maps 1.0 offizieller Teil des ISO-Standards 13250.

XTM ist nicht nur einfach eine Portierung des ISO-Standards nach XML, sondern beinhaltet auch einige Erweiterungen und auch Erneuerungen, wichtig ist aber, dass sich XTM nur auf XML und seine Satellitentechnologien bezieht, wogegen der ISO-Standard auf AFDR, HyTime und SGML aufbaut. Eines der Ziele der Autorengruppe TopicMaps.Org bei der Formulierung der XTM-Spezifikation war das Ziel der Kompatibilität zu ISO 13250, XML und XLink. Somit können mittels Transformationsregeln die beiden Notationsarten ISO 13250 und XTM automatisch ineinander übergeführt werden. Durch die bereits fortgeschrittende Verbreitung von XML wurde es somit einfacher, Topic Map-Dokumente selbst zu erstellen bzw. Applikationen hierfür zu entwickeln, ebenso Ziele für das Design von XTM. Der Hauptgrund für eine Portierung in XML war jedoch eine uneingeschränkte Verwendung der XML Topic Maps im Internet. [Biezunski and Newcomb, 2001]
Formal ausgedrückt ist jedes XTM-Dokument, ein XML-Dokument, das genau eine Topic Map beinhaltet, dessen Wurzelelement das `topicMap`-Element ist.

7.3 TAO of Topic Maps

TAO steht für die drei Grundkonzepte von Topic Maps, 'T' für topic, 'A' für association und 'O' für occurrence. Eingeführt wurde dieses Acronym von Steve Pepper in [Pepper, 2002].

7.3.1 Topics

Topics sind das Fundament der Topic Maps und können alles Beschreibbare sein, eine Entität, ein Konzept, ein Text oder ein Philosoph des alten Griechenlands. Was der Autor einer Topic Map wirklich als Topic definiert, hängt von bloß von Faktoren ab wie den potentiellen Anwendern, dem zugrundeliegenden Resourcenpool und der zukünftigen Verwendung. Hinter jedem Topic steht ein Subject.

> In the most generic sense, a subject is any thing whatsoever, regardless of whether it exists or has any other specific characteristics, about which anything whatsoever may be asserted by any means whatsoever. [Biezunski et al., 1999]

In [Pepper, 2002] wird *subject* mit Platos Idee assoziiert und die Schatten in Platons Höhle werden mit den Topics verglichen.
Formal ausgedrückt in [Biezunski and Newcomb, 2001]:

> A topic is a resource that acts as a proxy for some subject; it is the topic map system's representation of that subject.

Der Akt der Erstellung eines Topics wird *reification* (Verdinglichen) genannt:

> The relationship between a topic and its subject is defined to be one of reification. Reification of a subject allows topic characteristics to be assigned to the topic that reifies it. [Biezunski and Newcomb, 2001]

Nach der Verdinglichung eines Subjekts ist es möglich in der Topic Map Behauptungen über das Subjekt stellvertretend über das Topic auszusagen.

> We say that the topic 'reifies' the subject - or makes the subject 'real' for the system. [Biezunski and Newcomb, 2001]

Jedes Topic ist einem oder mehreren Topic Types zugeordnet, so könnten der TLP vom Typ Diplomarbeit sein, Wittgenstein vom Typ Philosoph und Autor sowie Volksschullehrer. Die Verbindung zwischen Topic und Topic Types entsprechen einer Klasse-Instanz-Relation und die Topic Types sind wiederum als Topics zu deklarieren.
Durch dieses Konzept ist es unter anderem möglich, mittels logischen Schlussregeln, implizites Wissen einer Topic Map zu explizieren.

7.3.2 Occurrences

Jede Topic kann beliebig viele Resourcen aufweisen, die in irgendeiner Weise relevant sind. Jede dieser Resourcen werden Occurrences (Vorkommen) genannt und können in zweifacher Weise auftreten: einmal adressierbar mittels Linkmechanismus über eine URI (resource reference) oder aber in der Topic Map selbst als Textdaten (resource data). Occurences werden technisch gesehen mit XLink umgesetzt.
Das Pendant zum Topic Type ist hier der Occurrence Type, der ebenfalls eine Klasse-Instanz-Beziehung modelliert und so Aussagen über die Art der Verweise macht wie bspw. Arktiel, Erwähnung, Kommentar, Wörterbucheintrag, Brief, Tondokument, Bild etc.

7.3.3 Associations

Assoziationen beschreiben Beziehungen zwischen Topics und sind das Konzept, das Topic Maps gegenüber einfachen Indices so wirkungsvoll macht. Mittels Assoziationen Types können Relationen zwischen einem oder mehreren Topics modelliert werden. Über die Art der Relation im Sinne von Symmetrie, Transitivität oder Reflexivität wird hierbei aber noch keine Aussage gemacht.
Da einfache Assoziationen nun einmal Relationen ausdrücken, sind sie von Haus aus multidirektional: Wenn ,,Wittgenstein den TLP verfasste", dann folgt, dass ,,der TLP von Wittgenstein verfasst wurde". Um die Gerichtetheit von Assoziationen auszudrücken, gibt es das Konzept der Association Role (Assoziationsrolle), die selbst als Topic deklariert werden muss. So kann Wittgenstein die Association Role ,,Autor" in der Assoziation ,,verfasst von" zugewiesen bekommen, und der TLP die Rolle ,,Werk". Mit diesem Konzept können direktionale Modellierungen gebildet werden, wie: ,,TLP (Werk) verfasst von Wittgenstein (Autor)". An einer Relation können eine oder mehrere Association Roles beteiligt sein, die ihrerseits alle als Topic deklariert werden müssen.
Durch die Möglichkeit Associations zwischen Topics anzugeben und diese zu typen unterscheiden sich Topic Maps von normalen Hyperlinktexten erheblich, denn bei normalen Hyperlinksystemen ist eine Referenz immer vom Inhalt des

Zieles bzw. der Quelle abhängig. Bei Topic Maps ist dagegen eine semantische Modellierung von Referenzierungen zwischen einzelnen Topics durch Associations möglich, die völlig unabhängig von jedwelchen Textdateien oder Occurrences erstellt werden können und somit wie bei dem obigen Beispiel Fakten darstellen können, wie dass der TLP von Wittgenstein verfasst wurde.

7.3.4 XTM-DTD

Es gibt für die völlige Ausfaltung der Mächtigkeit von XML Topic Maps weitere Konzepte, die hier in ihrer ganzen Tragweite erklärt und beschrieben den Rahmen sprengen würden, deshalb möchte ich mich auf die Wiedergabe der DTD der XML-Topic Maps getreu der Topic Maps Authoring Group beschränken und auf vertiefende Literatur hinweisen: [Biezunski et al., 1999], [Lobin, 2000], [Biezunski and Newcomb, 2001] und [Widham and Mück, 2000].

A XML Topic Maps Documenttyp-Definition

```
<!-- XML Topic Map DTD   -->
<!-- file: xtm1.dtd -->
<!-- XML Topic Map (XTM) DTD, Version 1.0
This is XTM, an XML interchange syntax for ISO 13250 Topic Maps.
XML Topic Map (XTM)

Copyright 2000-2001 TopicMaps.Org, All Rights Reserved.
Permission to use, copy, modify and distribute the XTM DTD and its
accompanying materials for any purpose and without fee is hereby
granted in perpetuity, provided that the above copyright notice and
this paragraph appear in all copies. The copyright holders make no
representation about the suitability of the DTD for any purpose. It
is provided "as is" without expressed or implied warranty.

Editors:    Steve Pepper      <pepper@ontopia.net>
               Graham Moore      <gdm@empolis.co.uk>
Authors:   Murray Altheim    <altheim@eng.sun.com>
                Michel Biezunski  <mb@infoloom.com>
                Sam Hunting       <shunting@etopicality.com>
                Steven R. Newcomb <srn@coolheads.com>

Status:     Release
Version:    v1.0.1
Revision:   $ Id: xtm1.dtd,v 1.2 2001/02/08 16:03:12 pepper Exp $
PublicId:   "-//TopicMaps.Org//DTD XML Topic Map (XTM) 1.0//EN"
Revisions:
#2001-01-21: removed baseName from occurrence
#2001-02-02: made variantName optional in variant
#2001-02-02: changed ID to #IMPLIED on association
#2001-02-02: changed ID to #IMPLIED on resourceData
#2001-02-02: changed PLUS to REP on member
-->
<!-- Use this URI to identify the default XTM namespace:

         "http://www.topicMaps.org/xtm/1.0/"

     Used to identify the XLink namespace:

         "http://www.w3.org/1999/xlink"
-->

<!-- topicMap: Topic Map document element. -->

<!ELEMENT topicMap
```

```
   ( topic | association | mergeMap )*
>
<!ATTLIST topicMap
  id         ID        #IMPLIED
  xmlns      CDATA     #FIXED 'http://www.topicmaps.org/xtm/1.0/'
  xmlns:xlink  CDATA     #FIXED 'http://www.w3.org/1999/xlink'
  xml:base        CDATA     #IMPLIED
>

<!-- topic: Topic element -->

<!ELEMENT topic
   ( instanceOf*, subjectIdentity?, ( baseName | occurrence )* )
>
<!ATTLIST topic
   id              ID        #REQUIRED
>

<!-- instanceOf: Points to a Topic representing a class -->

<!ELEMENT instanceOf  ( topicRef | subjectIndicatorRef ) >
<!ATTLIST instanceOf
   id              ID        #IMPLIED
>

<!-- subjectIdentity: Subject reified by Topic -->

<!ELEMENT subjectIdentity
   ( resourceRef?, ( topicRef | subjectIndicatorRef )* )
>
<!ATTLIST subjectIdentity
   id              ID        #IMPLIED
>

<!-- topicRef: Reference to a Topic element -->

<!ELEMENT topicRef  EMPTY >
<!ATTLIST topicRef
   id              ID        #IMPLIED
   xlink:type      NMTOKEN   #FIXED 'simple'
   xlink:href      CDATA     #REQUIRED
>

<!-- subjectIndicatorRef: Reference to a Subject Indicator -->

<!ELEMENT subjectIndicatorRef  EMPTY >
```

```
<!ATTLIST subjectIndicatorRef
   id              ID        #IMPLIED
   xlink:type      NMTOKEN   #FIXED 'simple'
   xlink:href      CDATA     #REQUIRED
>

<!-- baseName: Base Name of a Topic -->

<!ELEMENT baseName  ( scope?, baseNameString, variant* ) >
<!ATTLIST baseName
   id              ID        #IMPLIED
>

<!-- baseNameString: Base Name String container  -->

<!ELEMENT baseNameString  ( #PCDATA ) >
<!ATTLIST baseNameString
   id              ID        #IMPLIED
>

<!-- variant: Alternate forms of Base Name -->

<!ELEMENT variant  ( parameters, variantName?, variant* ) >
<!ATTLIST variant
   id              ID        #IMPLIED
>

<!-- variantName: Container for Variant Name -->

<!ELEMENT variantName  ( resourceRef | resourceData ) >
<!ATTLIST variantName
   id              ID        #IMPLIED
>

<!-- parameters: Processing context for Variant -->

<!ELEMENT parameters  ( topicRef | subjectIndicatorRef )+ >
<!ATTLIST parameters
   id              ID        #IMPLIED
>

<!-- occurrence: Resources regarded as an Occurrence -->

<!ELEMENT occurrence
   ( instanceOf?, scope?, ( resourceRef | resourceData ) )
>
```

```
<!ATTLIST occurrence
   id              ID        #IMPLIED
>

<!-- resourceRef: Reference to a Resource -->

<!ELEMENT resourceRef  EMPTY >
<!ATTLIST resourceRef
   id              ID        #IMPLIED
   xlink:type      NMTOKEN   #FIXED 'simple'
   xlink:href      CDATA     #REQUIRED
>

<!-- resourceData: Container for Resource Data -->

<!ELEMENT resourceData  ( #PCDATA ) >
<!ATTLIST resourceData
   id              ID        #IMPLIED
>

<!-- association: Topic Association  -->

<!ELEMENT association
   ( instanceOf?, scope?, member+ )
>
<!ATTLIST association
   id              ID        #IMPLIED
>

<!-- member: Member in Topic Association -->

<!ELEMENT member
   ( roleSpec?, ( topicRef | resourceRef | subjectIndicatorRef )* )
>
<!ATTLIST member
   id              ID        #IMPLIED
>

<!-- roleSpec: Points to a Topic serving as an Association Role .. -->

<!ELEMENT roleSpec  ( topicRef | subjectIndicatorRef ) >
<!ATTLIST roleSpec
   id              ID        #IMPLIED
>

<!-- scope: Reference to Topic(s) that comprise the Scope -->
```

```
<!ELEMENT scope  ( topicRef  | resourceRef | subjectIndicatorRef )+ >
<!ATTLIST scope
   id              ID        #IMPLIED
>

<!-- mergeMap: Merge with another Topic Map -->

<!ELEMENT mergeMap  ( topicRef | resourceRef | subjectIndicatorRef )* >
<!ATTLIST mergeMap
   id              ID        #IMPLIED
   xlink:type      NMTOKEN   #FIXED 'simple'
   xlink:href      CDATA     #REQUIRED
>

<!-- end of XML Topic Map (XTM) 1.0 DTD -->
```

B Topic Map zum Beginn Wittgensteins Manuskript 115

```
<?xml version="1.0"?>
<!DOCTYPE topicMap PUBLIC
        "-//TopicMaps.Org//DTD XML Topic Map (XTM) 1.0//EN"
        "file://usr/local/home/gromit/xml/xtm/xtm1.dtd">
<topicMap
        xmlns = 'http://www.topicmaps.org/xtm/1.0/'
        xmlns:xlink = 'http://www.w3.org/1999/xlink'>

<topic id="bild_sagt_sich">
 <baseName>
  <baseNameString>Bild sagt sich selbst</baseNameString>
 </baseName>
</topic>

<topic id="wiedererkennen">
 <baseName>
  <baseNameString>Wiedererkennen</baseNameString>
 </baseName>
</topic>

<topic id="wohlbekannt">
 <baseName>
  <baseNameString>wohlbekannt</baseNameString>
 </baseName>
</topic>

<topic id="sprachvereinbarung_off">
 <baseName>
  <baseNameString>Sprachuebereinkommen ausschalten</baseNameString>
 </baseName>
</topic>

<topic id="multiplizitaet_d_wohlbekanntseins">
 <baseName>
  <baseNameString>Multiplizitaet des Wohlbekanntseins</baseNameString>
 </baseName>
</topic>

<topic id="wiedererkennen_im_genrebild">
 <baseName>
  <baseNameString>Wiedererkennen im Genrebild?</baseNameString>
 </baseName>
```

```
</topic>

<topic id="erinnerungsbild">
 <baseName>
  <baseNameString>Erinnerungsbild</baseNameString>
 </baseName>
</topic>

<topic id="spezifisches_erinnerungsbild">
 <baseName>
  <baseNameString>spezifisches Erinnerungsbild</baseNameString>
 </baseName>
</topic>

<topic id="zur_methode">
 <baseName>
  <baseNameString>zur Methode: seltsame Beleuchtung
                               durch Philosophie</baseNameString>
 </baseName>
</topic>

<topic id="erzaehlung">
 <baseName>
  <baseNameString>Erzaehlung</baseNameString>
 </baseName>
</topic>

<topic id="voll_gestrichen">
 <baseName>
  <baseNameString>Paragraph vollstaendig gestrichen</baseNameString>
 </baseName>
</topic>

<topic id="teil_gestrichen">
 <baseName>
  <baseNameString>Paragraph teilweise gestrichen</baseNameString>
 </baseName>
</topic>

<topic id="umformulierung">
 <baseName>
  <baseNameString>Neuformulierung durch Wittgenstein</baseNameString>
 </baseName>
</topic>

<topic id="alt">
```

```
 <baseName>
  <baseNameString>alte Formulierung</baseNameString>
 </baseName>
</topic>

<topic id="neu">
 <baseName>
  <baseNameString>neue Formulierung</baseNameString>
 </baseName>
</topic>

<topic id="ms115">
 <baseName>
  <baseNameString>das Manuskript 115</baseNameString>
 </baseName>
 <occurrence>
  <resourceRef xlink:href="http://helmer.hit.uib.no/wab/"/>
 </occurrence>
</topic>

<topic id="wittgenstein"/>
 <baseName>
  <baseNameString>Ludwig Wittgenstein</baseNameString>
 </baseName>
</topic>

<topic id="autorschaft"/>
 <baseName>
  <baseNameString>Autorschaft</baseNameString>
 </baseName>
</topic>

<topic id="werk"/>
 <baseName>
  <baseNameString>Schriftwerk</baseNameString>
 </baseName>
</topic>

<topic id="autor"/>
 <baseName>
  <baseNameString>Autor</baseNameString>
 </baseName>
</topic>

<topic id="p1">
 <baseName>
```

```
  <baseNameString>Paragraph 1</baseNameString>
 </baseName>
 <instanceOf>
  <topicRef xlink:href="#bild_sagt_sich"/>
 </instanceOf>
 <occurrence>
  <resourceRef xlink:href="http://lw.philo.at/ms_115.html#p1"/>
 </occurrence>
</topic>

<topic id="p2">
 <baseName>
  <baseNameString>Paragraph 2</baseNameString>
 </baseName>
 <instanceOf>
  <topicRef xlink:href="#wiedererkennen"/>
  <topicRef xlink:href="#voll_gestrichen"/>
 </instanceOf>
 <occurrence>
  <resourceRef xlink:href="http://lw.philo.at/ms_115.html#p2"/>
 </occurrence>
</topic>

<topic id="p3">
 <baseName>
  <baseNameString>Paragraph 3</baseNameString>
 </baseName>
 <instanceOf>
  <topicRef xlink:href="#wiedererkennen"/>
  <topicRef xlink:href="#teil_gestrichen"/>
 </instanceOf>
 <occurrence>
  <resourceRef xlink:href="http://lw.philo.at/ms_115.html#p3"/>
 </occurrence>
</topic>

<topic id="p4">
 <baseName>
  <baseNameString>Paragraph 4</baseNameString>
 </baseName>
 <instanceOf>
  <topicRef xlink:href="#wohlbekannt"/>
  <topicRef xlink:href="#teil_gestrichen"/>
 </instanceOf>
 <occurrence>
  <resourceRef xlink:href="http://lw.philo.at/ms_115.html#p4"/>
```

```
 </occurrence>
</topic>

<topic id="p5">
 <baseName>
  <baseNameString>Paragraph 5</baseNameString>
 </baseName>
 <instanceOf>
  <topicRef xlink:href="#wohlbekannt"/>
 </instanceOf>
 <occurrence>
   <resourceRef xlink:href="http://lw.philo.at/ms_115.html#p5"/>
 </occurrence>
</topic>

<topic id="p6">
 <baseName>
  <baseNameString>Paragraph 6</baseNameString>
 </baseName>
 <instanceOf>
  <topicRef xlink:href="#wohlbekannt"/>
  </instanceOf>
 <occurrence>
   <resourceRef xlink:href="http://lw.philo.at/ms_115.html#p6"/>
 </occurrence>
</topic>

<topic id="p7">
 <baseName>
  <baseNameString>Paragraph 7</baseNameString>
 </baseName>
 <instanceOf>
  <topicRef xlink:href="#wohlbekannt"/>
 </instanceOf>
 <occurrence>
  <resourceRef xlink:href="http://lw.philo.at/ms_115.html#p7"/>
 </occurrence>
</topic>

<topic id="p8">
 <baseName>
  <baseNameString>Paragraph 8</baseNameString>
 </baseName>
 <instanceOf>
  <topicRef xlink:href="#wohlbekannt"/>
  <topicRef xlink:href="#voll_gestrichen"/>
```

```
 </instanceOf>
 <occurrence>
  <resourceRef xlink:href="http://lw.philo.at/ms_115.html#p8"/>
 </occurrence>
</topic>

<topic id="p9">
 <baseName>
  <baseNameString>Paragraph 9</baseNameString>
 </baseName>
 <instanceOf>
  <topicRef xlink:href="#sprachvereinbarung_off"/>
 </instanceOf>
 <occurrence>
  <resourceRef xlink:href="http://lw.philo.at/ms_115.html#p9"/>
 </occurrence>
</topic>

<topic id="p10">
 <baseName>
  <baseNameString>Paragraph 10</baseNameString>
 </baseName>
 <instanceOf>
  <topicRef xlink:href="#sprachvereinbarung_off"/>
 </instanceOf>
 <occurrence>
  <resourceRef xlink:href="http://lw.philo.at/ms_115.html#p10"/>
 </occurrence>
</topic>

<topic id="p11">
 <baseName>
  <baseNameString>Paragraph 11</baseNameString>
 </baseName>
 <instanceOf>
  <topicRef xlink:href="#sprachvereinbarung_off"/>
 </instanceOf>
 <occurrence>
  <resourceRef xlink:href="http://lw.philo.at/ms_115.html#p11"/>
 </occurrence>
</topic>

<topic id="p12">
 <baseName>
  <baseNameString>Paragraph 12</baseNameString>
 </baseName>
```

```
 <instanceOf>
  <topicRef xlink:href="#sprachvereinbarung_off"/>
 </instanceOf>
 <occurrence>
  <resourceRef xlink:href="http://lw.philo.at/ms_115.html#p12"/>
 </occurrence>
</topic>

<topic id="p13">
 <baseName>
  <baseNameString>Paragraph 13</baseNameString>
 </baseName>
 <instanceOf>
  <topicRef xlink:href="#sprachvereinbarung_off"/>
 </instanceOf>
 <occurrence>
  <resourceRef xlink:href="http://lw.philo.at/ms_115.html#p13"/>
 </occurrence>
</topic>

<topic id="p14">
 <baseName>
  <baseNameString>Paragraph 14</baseNameString>
 </baseName>
 <instanceOf>
  <topicRef xlink:href="#multiplizitaet_d_wohlbekanntseins"/>
 </instanceOf>
 <occurrence>
  <resourceRef xlink:href="http://lw.philo.at/ms_115.html#p14"/>
 </occurrence>
</topic>

<topic id="p15">
 <baseName>
  <baseNameString>Paragraph 15</baseNameString>
 </baseName>
 <instanceOf>
  <topicRef xlink:href="#multiplizitaet_d_wohlbekanntseins"/>
 </instanceOf>
 <occurrence>
  <resourceRef xlink:href="http://lw.philo.at/ms_115.html#p15"/>
 </occurrence>
</topic>

<topic id="p16">
 <baseName>
```

```
  <baseNameString>Paragraph 16</baseNameString>
 </baseName>
 <instanceOf>
  <topicRef xlink:href="#multiplizitaet_d_wohlbekanntseins"/>
 </instanceOf>
 <occurrence>
  <resourceRef xlink:href="http://lw.philo.at/ms_115.html#p16"/>
 </occurrence>
</topic>

<topic id="p17">
 <baseName>
  <baseNameString>Paragraph 17</baseNameString>
 </baseName>
 <instanceOf>
  <topicRef xlink:href="#wiedererkennen_im_genrebild"/>
 </instanceOf>
 <occurrence>
  <resourceRef xlink:href="http://lw.philo.at/ms_115.html#p17"/>
 </occurrence>
</topic>

<topic id="p18">
 <baseName>
  <baseNameString>Paragraph 18</baseNameString>
 </baseName>
 <instanceOf>
  <topicRef xlink:href="#wiedererkennen_im_genrebild"/>
 </instanceOf>
 <occurrence>
  <resourceRef xlink:href="http://lw.philo.at/ms_115.html#p18"/>
 </occurrence>
</topic>

<topic id="p19">
 <baseName>
  <baseNameString>Paragraph 19</baseNameString>
 </baseName>
 <instanceOf>
  <topicRef xlink:href="#wiedererkennen"/>
 </instanceOf>
 <occurrence>
  <resourceRef xlink:href="http://lw.philo.at/ms_115.html#p19"/>
 </occurrence>
</topic>
```

```
<topic id="p20">
 <baseName>
  <baseNameString>Paragraph 20</baseNameString>
 </baseName>
 <instanceOf>
  <topicRef xlink:href="#wiedererkennen"/>
 </instanceOf>
 <occurrence>
  <resourceRef xlink:href="http://lw.philo.at/ms_115.html#p20"/>
 </occurrence>
</topic>

<topic id="p21">
 <baseName>
  <baseNameString>Paragraph 21</baseNameString>
 </baseName>
 <instanceOf>
  <topicRef xlink:href="#wiedererkennen"/>
 </instanceOf>
 <occurrence>
  <resourceRef xlink:href="http://lw.philo.at/ms_115.html#p21"/>
 </occurrence>
</topic>

<topic id="p22">
 <baseName>
  <baseNameString>Paragraph 22</baseNameString>
 </baseName>
 <instanceOf>
  <topicRef xlink:href="#erinnerungsbild"/>
 </instanceOf>
 <occurrence>
  <resourceRef xlink:href="http://lw.philo.at/ms_115.html#p22"/>
 </occurrence>
</topic>

<topic id="p23">
 <baseName>
  <baseNameString>Paragraph 23</baseNameString>
 </baseName>
 <instanceOf>
  <topicRef xlink:href="#erinnerungsbild"/>
 </instanceOf>
 <occurrence>
  <resourceRef xlink:href="http://lw.philo.at/ms_115.html#p23"/>
 </occurrence>
```

```
</topic>

<topic id="p24">
 <baseName>
  <baseNameString>Paragraph 24</baseNameString>
 </baseName>
 <instanceOf>
  <topicRef xlink:href="#erinnerungsbild"/>
 </instanceOf>
 <occurrence>
  <resourceRef xlink:href="http://lw.philo.at/ms_115.html#p24"/>
 </occurrence>
</topic>

<topic id="p25">
 <baseName>
  <baseNameString>Paragraph 25</baseNameString>
 </baseName>
 <instanceOf>
  <topicRef xlink:href="#erinnerungsbild"/>
 </instanceOf>
 <occurrence>
  <resourceRef xlink:href="http://lw.philo.at/ms_115.html#p25"/>
 </occurrence>
</topic>

<topic id="p26">
 <baseName>
  <baseNameString>Paragraph 26</baseNameString>
 </baseName>
 <instanceOf>
  <topicRef xlink:href="#spezifisches_erinnerungsbild"/>
 </instanceOf>
 <occurrence>
  <resourceRef xlink:href="http://lw.philo.at/ms_115.html#p26"/>
 </occurrence>
</topic>

<topic id="p27">
 <baseName>
  <baseNameString>Paragraph 27</baseNameString>
 </baseName>
 <instanceOf>
  <topicRef xlink:href="#spezifisches_erinnerungsbild"/>
 </instanceOf>
 <occurrence>
```

```
  <resourceRef xlink:href="http://lw.philo.at/ms_115.html#p27"/>
 </occurrence>
</topic>

<topic id="p28">
 <baseName>
  <baseNameString>Paragraph 28</baseNameString>
 </baseName>
 <instanceOf>
  <topicRef xlink:href="#spezifisches_erinnerungsbild"/>
 </instanceOf>
 <occurrence>
  <resourceRef xlink:href="http://lw.philo.at/ms_115.html#p28"/>
 </occurrence>
</topic>

<topic id="p29">
 <baseName>
  <baseNameString>Paragraph 29</baseNameString>
 </baseName>
 <instanceOf>
  <topicRef xlink:href="#spezifisches_erinnerungsbild"/>
 </instanceOf>
 <occurrence>
  <resourceRef xlink:href="http://lw.philo.at/ms_115.html#p29"/>
 </occurrence>
</topic>

<topic id="p30">
 <baseName>
  <baseNameString>Paragraph 30</baseNameString>
 </baseName>
 <instanceOf>
  <topicRef xlink:href="#zur_methode"/>
 </instanceOf>
 <occurrence>
  <resourceRef xlink:href="http://lw.philo.at/ms_115.html#p30"/>
 </occurrence>
</topic>

<topic id="p31">
 <baseName>
  <baseNameString>Paragraph 31</baseNameString>
 </baseName>
 <instanceOf>
  <topicRef xlink:href="#zur_methode"/>
```

```
 </instanceOf>
 <occurrence>
  <resourceRef xlink:href="http://lw.philo.at/ms_115.html#p31"/>
 </occurrence>
</topic>

<topic id="p32">
 <baseName>
  <baseNameString>Paragraph 32</baseNameString>
 </baseName>
 <instanceOf>
  <topicRef xlink:href="#bild_sagt_sich"/>
 </instanceOf>
 <occurrence>
  <resourceRef xlink:href="http://lw.philo.at/ms_115.html#p32"/>
 </occurrence>
</topic>

<topic id="p33">
 <baseName>
  <baseNameString>Paragraph 33</baseNameString>
 </baseName>
 <instanceOf>
  <topicRef xlink:href="#erzaehlung"/>
 </instanceOf>
 <occurrence>
  <resourceRef xlink:href="http://lw.philo.at/ms_115.html#p33"/>
 </occurrence>
</topic>

<topic id="p34">
 <baseName>
  <baseNameString>Paragraph 34</baseNameString>
 </baseName>
 <instanceOf>
  <topicRef xlink:href="#erzaehlung"/>
 </instanceOf>
 <occurrence>
  <resourceRef xlink:href="http://lw.philo.at/ms_115.html#p34"/>
 </occurrence>
</topic>

<topic id="p35">
 <baseName>
  <baseNameString>Paragraph 35</baseNameString>
 </baseName>
```

```
 <instanceOf>
  <topicRef xlink:href="#erzaehlung"/>
 </instanceOf>
 <occurrence>
  <resourceRef xlink:href="http://lw.philo.at/ms_115.html#p35"/>
 </occurrence>
</topic>

<association>
 <instanceOf>
  <topicRef xlink:href="#umformulierung"/>
 </instanceOf>
 <member>
  <roleSpec>
    <topicRef xlink:href="#alt"/>
  </roleSpec>
    <topicRef xlink:href="#p4"/>
    <topicRef xlink:href="#p5"/>
 </member>
 <member>
  <roleSpec>
    <topicRef xlink:href="#neu"/>
  </roleSpec>
    <topicRef xlink:href="#p20"/>
    <topicRef xlink:href="#p21"/>
 </member>
</association>

<association>
 <instanceOf>
  <topicRef xlink:href="#autorschaft"/>
 </instanceOf>
 <member>
  <roleSpec>
    <topicRef xlink:href="#autor"/>
  </roleSpec>
    <topicRef xlink:href="#wittgenstein"/>
 </member>
 <member>
  <roleSpec>
    <topicRef xlink:href="#werk"/>
  </roleSpec>
    <topicRef xlink:href="#ms115"/>
 </member>
</association>
```

```
</topicMap>
```

Literatur

[Abiteboul et al., 1999] Abiteboul, S., Buneman, P., and Suciu, D. (1999). *Data on the Web. From Relations to Semistructured Data and XML.* Morgan Kaufmann, California.

[AFDR, 1997] AFDR (1997). Architectural Form Definition Requirements (AFDR). `http://www.ornl.gov/sgml/wg8/docs/n1920/html/clause-A.3.html`.

[Berners-Lee, 1997] Berners-Lee, T. (1997). Metadata Architecture. `http://www.w3.org/DesignIssues/Metadata.html`.

[Berners-Lee, 1998] Berners-Lee, T. (1998). Semantic Web Road map. `http://www.w3.org/DesignIssues/Semantic.html`.

[Berners-Lee et al., 1998] Berners-Lee, T., Fielding, R., and Masinter., L. (1998). Rfc 2396: Uniform Resource Identifiers (URI): Generic Syntax. Technical report, Internet Engineering Task Force. `http://www.ietf.org/rfc/rfc2396.txt`.

[Biezunski, 2000] Biezunski, M. (2000). Topic Maps at a glance. `http://www.infoloom.com/tmsample/bie0.htm`.

[Biezunski et al., 1999] Biezunski, M., Bryan, M., and Newcomb, S. (1999). ISO/IEC FCD 13250:1999 - Topic Maps. Technical report, International Organization for Standardization. `http://www.ornl.gov/sgml/sc34/document/0058.htm`.

[Biezunski and Newcomb, 2001] Biezunski, M. and Newcomb, S. (2001). XML Topic Maps (XTM) 1.0. Specification, TopicMaps.Org Authoring Group. `http://www.topicmaps.org/xtm/1.0/`.

[Böszörményi and Weich, 1995] Böszörményi, L. and Weich, C. (1995). *Programmieren mit Modula-3. Eine Einführung in stilvolle Programmierung.* Springer, Berlin, Heidelberg, New York.

[Bray et al., 1999] Bray, T., Hollander, D., and Layman, A. (1999). Namespaces in xml. Recommendation, World Wide Web Consortium. `http://www.w3.org/TR/1999/REC-xml-names-19990114/`.

[Bray et al., 1998] Bray, T., Paoli, J., and Sperberg-McQueen, C. (1998). Extensible Markup Language (XML) 1.0. Technical specification. `http://www.w3.org/TR/1998/REC-xml-19980210/`.

[Britannica, 2003] Britannica (2003). History of Encyclopædia Britannica and Britannica.com. `http://corporate.britannica.com/company_info.html`.

[Buneman, 2000] Buneman, P. (2000). Semistructured data. `http://www.cis.upenn.edu/~db`.

[Coombs et al., 1987] Coombs, J. H., Renear, A. H., and DeRose, S. J. (1987). Markup Systems and the Future of Scholarly Text Processing. `http://www.oasis-open.org/cover/coombs.html`.

[DeRose et al., 2001] DeRose, S., Maler, E., Orchard, D., and Trafford, B. (2001). XML Linking Language (XLink) Version 1.0. Recommendation, World Wide Web Consortium. `http://www.w3.org/TR/xlink/`.

[DSSSL,] DSSSL. Informaion Processing - Processing Languages - Document Style Semantics and Specification Language (DSSSL). `http://www.ornl.gov/sgml/wg8/`.

[Eco, 1994] Eco, U. (1994). *Einführung in die Semiotik*. Fink, München.

[Floridi, 1999] Floridi, L. (1999). *philosophy and computing*. Routledge, London.

[Fried and Süßmann, 2001] Fried, J. and Süßmann, J. (2001). *Revolutionen des Wissens. Von der Steinzeit bis zur Moderne*. Beck, München.

[Gaus, 2000] Gaus, W. (2000). *Dokumentations- und Ordnungslehre Theorie und Praxis des Information Retrieval*. Springer, Berlin and Heidelberg and New York.

[Hartmann, 1999] Hartmann, F. (1999). *Cyber.Philosophy: Medientheoretische Auslotungen*. Passagen, Wien.

[Hartmann, 2000] Hartmann, F. (2000). *Medienphilosophie*. WUV, Wien.

[Hjelm, 2001] Hjelm, J. (2001). *Creating the Semantic Web with RDF*. John Wiley and Sons, New York and Chichester and Weinheim.

[ISO, 2002] ISO (2002). What is ISO? `http://www.iso.org/`.

[Keep et al., 2000] Keep, C., McLaughlin, T., and Parmar, R. (2000). Hypertext. `http://www.iath.virginia.edu/elab/hfl0037.html`.

[Lassila and Swick, 1999] Lassila, O. and Swick, R. R. (1999). Resource Description Framework (RDF) Model and Syntax Specification. Technical specification, World Wide Web Consortium. `http://www.w3.org/TR/REC-rdf-syntax/`.

[Leroi-Gourhan, 1995] Leroi-Gourhan, A. (1995). *Hand und Wort Die Evolution von Technik, Sprache und Kunst*. Suhrkamp, Frankfurt am Main.

[Lobin, 2000] Lobin, H. (2000). *Informationsmodellierung in XML und SGML*. Springer, Berlin, Heidelberg, New York.

[Lyotard, 1999] Lyotard, J.-F. (1999). *Das postmoderne Wissen: Ein Bericht*. Passagen, Wien.

[Marsh, 1999] Marsh, J. (1999). Xml base (xbase). Recommendation, World Wide Web Consortium. `http://www.w3.org/TR/2001/REC-xmlbase-20010627/`.

[Mates, 1997] Mates, B. (1997). *Elemantere Logik Prädikatenlogik der ersten Stufe*. Vandenhoeck und Ruprecht, Göttingen.

[Molyneux, 1996] Molyneux, R. E. (1996). Fremont Rider's Legacy. `http://www.arl.org/newsltr/188/fremont.html`.

[Müller and Schmidt, 2000] Müller, C. and Schmidt, I. (2000). Planning a new type of literary edition: the Thomas Mann Project. `http://www.gca.org/papers/xmleurope2000/papers/s09-02.html`.

[Nelkin, 1987] Nelkin, D. (1987). Science, Technology and Public Policy. `http://depts.washington.edu/hssexec/newsletter/1997/nelkin.html`.

[Penrose, 1991] Penrose, R. (1991). *Computerdenken*. Spektrum der Wiss. Verlagsges., Heidelberg.

[Pepper, 1996] Pepper, S. (1996). Information Retrieval. `http://www.dcs.gla.ac.uk/Keith/Preface.html`.

[Pepper, 2002] Pepper, S. (2002). The TAO of Topic Maps. Finding the Way in the Age of Infoglut. `http://www.ontopia.net/topicmaps/materials/tao.html`.

[Ray, 2001] Ray, E. T. (2001). *Learning XML. Guide to Creating Self-Describing Data*. O'Reilly & Associations, Beijing and Cambridge, Farnbam and Köln and Paris and Sebastopoi and Taipei and Tokyo.

[Rechenberg, 2000] Rechenberg, P. (2000). *Was ist Informatik? Eine allgemeinverständliche Einführung*. Hanser, München, Wien.

[Renear et al., 1993] Renear, A., Mylonas, E., and Durand, D. (1993). Refining our Notion of What Text Really Is: The Problem of Overlapping Hierarchies. `http://www.stg.brown.edu/resources/stg/monographs/ohco.html`.

[Strawson, 1994] Strawson, P. F. (1994). *Analyse und Metaphysik. Eine Einführung in die Philosophie.* Deutscher Taschenbuch Verlag, München.

[van Rijsbergen, 1996] van Rijsbergen, C. (1996). Information Retrieval. `http://www.dcs.gla.ac.uk/Keith/Preface.html`.

[Voß, 2001] Voß, S. (2001). *Informationsmanagement*. Springer, Berlin, Heidelberg, New York.

[Widham and Mück, 2000] Widham, R. and Mück, T. A. (2000). *Topic Maps. Semantische Suche im Internet.* Springer, Berlin, Heidelberg, New York.

[Zemanek, 1993] Zemanek, H. (1993). *Informatik und Philosophie*. BI-Wiss.-Verl., Mannheim, Wien, Zürich.