



Medienbausteine für web-basierte Lernsysteme

vom Fachbereich 20
der Technischen Universität Darmstadt
genehmigte Dissertationsschrift
von

Diplom-Wirtschaftsinformatiker Achim Steinacker

geboren am 23.05.1969 in Heilbronn

zur Erlangung des Grades eines
Doktor-Ingenieur (Dr.-Ing.)

Darmstadt 2001
Hochschulkennziffer D17

Erstreferent: Prof. Dr.-Ing. Ralf Steinmetz
Koreferent: Prof. Dr.-Ing. Wolfgang Effelsberg

Tag der Einreichung: 18. Oktober 2001
Tag der Disputation: 11. Januar 2002

Inhaltsverzeichnis

Inhaltsverzeichnis	iii
Abbildungsverzeichnis	vii
Tabellenverzeichnis	ix
1 Einleitung	11
1.1 Multimediales Lernen bei Bildungseinrichtungen	11
1.2 Virtuelle Marktplätze für Lernmaterialien	14
1.3 Anforderungen an multimediales Lernmaterial.....	16
1.4 Ziel und Aufbau der Arbeit	16
2 Metadaten für Web-Ressourcen	18
2.1 Einleitung	18
2.1.1 Definition von Metadaten.....	19
2.1.2 Zweck von Metadaten	20
2.2 Struktur von Metadaten	23
2.2.1 Strukturelle Ebene	23
2.2.2 Begriffliche Ebene.....	24
2.2.3 Definition Medienbaustein	24
2.3 Arten von Metadaten	25
2.3.1 Inhaltsabhängige Metadaten.....	25
2.3.2 Inhaltsunabhängige Metadaten.....	26
2.4 Beispiele für Metadaten.....	27
2.4.1 PICS.....	28
2.4.2 Dublin Core	30
2.4.3 MPEG-7.....	33
2.4.4 Warwick Framework	36
2.5 Generierung von Metadaten	37
2.5.1 Methoden zur Generierung von Metadaten.....	37
2.5.2 Extraktionsebenen	39
2.5.3 Generierungsmechanismen.....	39
2.6 Kodierung von Metadaten	42

2.6.1	Kodierung innerhalb der Ressource	43
2.6.2	Kodierung zusammen mit der Ressource	43
2.6.3	Kodierung unabhängig von der Ressource.....	44
3	Metadaten für multimediale Lernressourcen	48
3.1	Learning Objects Metadata (LOM)	49
3.1.1	Einbettung von LOM.....	49
3.1.2	Ziele von LOM.....	51
3.1.3	Elemente von LOM	52
3.1.4	Struktur von LOM	52
3.2	Programmgestützte Generierung von LOM-Metadaten	54
3.2.1	General	54
3.2.2	Life Cycle	56
3.2.3	MetaMetadata.....	56
3.2.4	Technical	57
3.2.5	Educational.....	57
3.2.6	Rights.....	58
3.2.7	Relation.....	58
3.2.8	Annotation	58
3.2.9	Classification	58
3.2.10	Zusammenfassung und Bewertung.....	58
3.3	Datenmodell für LOM.....	59
3.3.1	Struktur.....	59
3.3.2	Document Type Definition für LOM	60
3.4	Dynamische Medienbausteine.....	64
3.4.1	Parametrisierbare Medienbausteine.....	66
3.4.2	Schnittstelle für Medienbausteine	67
3.5	Automatische Komposition von Medienbausteinen.....	69
3.5.1	Eigenschaften einzelner Medienbausteine	70
3.5.2	Beziehungen zwischen Medienbausteinen	71
3.6	Bewertung von LOM.....	72
3.6.1	Anzahl der Metadatenelemente	72
3.6.2	Austauschbarkeit und Werkzeuge zur Erstellung und Suche.....	73
3.6.3	Vollständigkeit von LOM.....	73

3.7	Semantische Netze.....	73
3.7.1	Wissensrepräsentation	74
3.7.2	Einsatz von semantischen Netzen.....	77
3.7.3	Architektur.....	78
3.7.4	Kodierungsformate für semantische Netze.....	79
3.8	Einsatz von semantischen Netzen in Lernsystemen.....	80
3.8.1	Datenmodell	85
3.8.2	Kodierung	86
3.8.3	Architektur.....	88
3.8.4	Werkzeuge zur Erstellung der Metadaten	90
4	Autoren- und Suchwerkzeuge für LOM	94
4.1	LOM-Editor	94
4.1.1	Analyse und Design der Architektur	94
4.1.2	Verwendung vorgegebener Vokabulare	95
4.1.3	Oberfläche des Editors.....	98
4.2	Realisierung einer XML-basierten Suchmaschine für LOM.....	99
4.2.1	Auswahl der Datenbank.....	100
4.2.2	Suchoberfläche	103
4.2.3	Server.....	105
4.2.4	Vollständige Architektur der zu entwickelnden Applikation	109
5	Entwicklung eines Werkzeug zum Aufbau eines semantischen Netzes .	114
5.1	Werkzeuge für semantische Netze	114
5.1.1	Bewertung.....	116
5.1.2	Anforderungen.....	116
5.2	Architektur des ConceptSpace-Editors.....	117
5.3	Visualisierungen von Baumstrukturen	118
5.3.1	Microsoft Explorer	119
5.3.2	HAWK.....	120
5.3.3	The Hyperbolic Tree von inight	121
5.4	Visualisierungen von allgemeinen Netzen	122
5.4.1	Ilog.....	123
5.4.2	Brain von Natrifical	124

5.4.3	i2 Analyst Notebook.....	125
5.5	Zusammenfassung	127
5.6	Darstellung des Graphen	128
5.6.1	Restriktionen.....	130
5.6.2	Strategie für Zusammenfassungen von Knoten.....	131
5.6.3	Optimierung des Graphen.....	132
5.7	Funktionswertberechnung des Graphen	134
5.7.1	Topologischer Abstand eines Graphen.....	135
5.7.2	Einhaltung einer vorgegebenen Kantenlänge.....	135
5.7.3	Topologische Abstände nicht direkt verbundener Knoten	136
5.7.4	Überdeckungsfreie Anordnung der Knoten.....	136
5.7.5	Spezifikation der Lage einzelner Knoten durch geometrische Regeln	138
5.7.6	Stabilität des Layouts gegenüber kleinen Änderungen	139
5.7.7	Einbeziehung von Grundstrukturen.....	139
5.8	Implementierung.....	139
5.8.1	SFK.....	140
5.8.2	GraphVisualisierung.....	142
5.8.3	SchemaVisualisierung	144
5.8.4	Benutzeroberfläche des ConceptSpace-Editor	144
6	Zusammenfassung und Ausblick	146
Anhang A:	Document Type Definition für LOM	148
Anhang B:	Grundschemata der deutschen Übersetzung des LOM-Entwurfs.....	150
Literaturverzeichnis		176

Abbildungsverzeichnis

Abbildung 1	: Modell eines virtuellen Marktplatz für Lernmaterialien.....	14
Abbildung 2	: Zusammensetzung eines Medienbausteins (vgl.[130]).....	22
Abbildung 3	: Klassifikation von Metadaten	26
Abbildung 4	: Beschreibung eines Rating Services für PICS	28
Abbildung 5	: Beispiel für ein PICS-Label	28
Abbildung 6	: Beispiel für eine Dublin Core Beschreibung (vgl.[53]).....	31
Abbildung 7	: Komponenten des MPEG7-Systems (vgl. [71])	32
Abbildung 8	: Beispiel einer MPEG7-Beschreibung (vgl.[53]).....	34
Abbildung 9	: Zusammenhang zwischen Mediendaten und Metadaten	41
Abbildung 10	: Metadatenbeschreibung in HTML.....	42
Abbildung 11	: 5 Schichten-Modell der LTSA (vgl. [55])	48
Abbildung 12	: Komponenten eines web-basierten Lernsystems ([55]).....	49
Abbildung 13	: Baumstruktur der Learning Object Metadata	58
Abbildung 14	: Beispiel für eine LOM-Beschreibung mit XML.....	62
Abbildung 15	: Beispiel für einen dynamischen Medienbaustein	63
Abbildung 16	: SCORM API (vgl. [2]).....	67
Abbildung 17	: Architektur eines wissensbasierten Systems (vgl. [47])	78
Abbildung 18	: Beispiel für ein Inhaltsverzeichnis I	81
Abbildung 19	: Beispiel für ein Inhaltsverzeichnis II	81
Abbildung 20	: extrahiertes Inhaltsverzeichnis über Kompressionsverfahren	83
Abbildung 21	: extrahiertes Inhaltsverzeichnis über JPEG.....	84
Abbildung 22	: Kombination von LOM-Beschreibungen und semantischen Netzen.....	85
Abbildung 23	: Komponenten eines Lernsystems	89
Abbildung 24	: Architektur eines Lernsystems.....	91
Abbildung 25	: Abbildung des Objektmodells auf verschiedene Persistenzmechanismen .	94
Abbildung 26	: Auszug aus der Konfigurationsdatei für Vokabularwerte	96
Abbildung 27	: Anzeige der Vokabulare im Editor	97
Abbildung 28	: Benutzeroberfläche des Editors	98
Abbildung 29	: Collection-, DocType. und Node-Elemente des LOM Schemas	100
Abbildung 30	: Ergebnis einer Suchanfrage	101
Abbildung 31	: Front-End der Suchmaschine	103
Abbildung 32	: Vereinfachtes Verarbeitungsmodell von Cocoon ([87]).....	108
Abbildung 33	: Architektur der Suchmaschine	109

Abbildung 34	: Übersichtsresultat einer Suchanfrage	110
Abbildung 35	: Detailansicht einer Suchanfrage	111
Abbildung 36	: Baumansicht des Microsoft Explorers	118
Abbildung 37	: Präsentation der Suchergebnisse mit HAWK ([99])	120
Abbildung 38	: Navigation mit einer Hyperbolic-Tree Visualisierung	121
Abbildung 39	: Visualisierung mit Ilog-Bibliothek.....	122
Abbildung 40	: Kreuzungsfreie Visualisierung des ILog-Graph.....	123
Abbildung 41	: Visualisierung mit Nartifical Brain	124
Abbildung 42	: Visualisierung mit Analyst Notebook	125
Abbildung 43	: Visualisierung der Übersicht einer Analyst Notebook Struktur.....	126
Abbildung 44	: Zusammenfassung von Knoten zu virtuellen Knoten	128
Abbildung 45	: Stern- bzw. baumförmige Visualisierung eines Graphen.....	129
Abbildung 46	: Visualisierungen für 1:n:1-Beziehungen.....	129
Abbildung 47	: Sequenz einer Interaktion mit Zusammenfassung von Knoten.....	131
Abbildung 48	: Architektur des ConceptSpace-Editors	140
Abbildung 49	: GUI des ConceptSpace-Editors.....	144

Tabellenverzeichnis

Tabelle 1	: Beispiel für eine Metadatenbeschreibung	20
Tabelle 2	Kategorie Dynamic Metadata für LOM (vgl. [110])	66
Tabelle 3	Abbildung von Vokabularwerten auf Textbeschreibungen	96
Tabelle 4	Übersicht der XSL-Elemente	107
Tabelle 5	Kriterien zum Vergleich von Ontologie-Editoren (vgl. [36])	114
Tabelle 6	Kategorienschema des LOM-Entwurfs	150
Tabelle 7	Schema für LOM Datum	174
Tabelle 8	Schema für LOM Vokabular	175

1 Einleitung

Wo das multimediale Lernen beginnt und das reine Stöbern oder die Informationssuche im WWW aufhört, kann so wenig eindeutig definiert werden wie der Begriff Lernen selbst. Im Gegensatz zum Begriff des multimedialen Lernens ist das wachsende wirtschaftliche Potential, das dem Bereich des multimedialen Lernens zugeschrieben wird, unzweifelhaft. Eine Studie des Branchenverband Software and Information Industry Association (SIIA) prognostiziert einen Umsatz von 4.21 Billionen Euro alleine für den europäischen Markt des computerbasierten Lernens [27]. Nachfrager sind sowohl kleine Unternehmen, die standardisierte Schulungen für Mitarbeiter online und dadurch billiger durchführen, als auch große Unternehmen, die sich multimediale web-basierte Schulungen individuell zusammenstellen lassen und dann zur internen Weiterbildung verwenden. Unabhängig davon, wie realistisch die Prognose ist, sind Universitäten bisher wenig an den Verdienstmöglichkeiten beteiligt, die sich durch die multimediale Lehre ergeben. Der Grund hierfür ist, daß für die Universitäten bisher keine Geschäftsmodelle existieren, die eine kommerzielle Weiterverwertung des eigenen Materials ermöglicht. Dabei sind es sehr oft die Hochschulen, die innovatives multimediales Lernmaterial entwickeln und den pädagogischen und technischen Einsatz der computergestützten Lehre untersuchen. Die Möglichkeit einer Finanzierung für Universitäten als Anbieter von Schulungsmaterial ist daher mit ein Grund, warum entsprechende Projekte derzeit gefördert werden und das Interesse an einer Verwaltung, Verteilung und Wiederverwendung von Lernmaterial stark zunimmt [20].

1.1 *Multimediales Lernen bei Bildungseinrichtungen*

Eine der Grundaufgaben von Universitäten, Schulen und andere Bildungseinrichtungen ist es, Lernmaterialien zu erstellen und Studenten oder Schülern Wissen zu vermitteln. Es ist daher nicht verwunderlich, dass sich die Organisation dieser Einrichtungen durch den Einsatz elektronischer Lernmaterialien stark verändern wird. Während bei der klassischen Lernsituation das Wissen, das vermittelt wird, durch einen eindeutigen Wissensträger, dem Lehrer, "lokalisiert" ist, kommt es durch die Nutzung moderner Kommunikationstechniken zu einer Verteilung des Wissens, und damit auch zu einer verteilten Speicherung des Wissens [148].

Nach [137] führt dies dazu, dass Unterrichtseinheiten und Vorlesungen zukünftig mehr und mehr daraus bestehen werden Methoden und Techniken zu vermitteln, wie Lernende selbstständig Wissen erlangen und Zusammenhänge erkennen können; die Vermittlung der eigentlichen Inhalte wird zweitrangig. Unterstützung bei dieser neuen Art des Wissenserwerbs und damit auch Kompensation des Verlusts der persönlichen Betreuung kann durch Einbeziehung von Mitlernern und Tutoren erfolgen [147]. Dieser unter dem Namen Computer Supported Cooperative Learning (CSCL) bekannte Forschungsbereich soll in dieser Arbeit nicht weiter betrachtet werden. Ein aktueller Überblick über CSCL ist in [148] beschrieben, Untersuchungen über Datenmodelle und Architekturen für kooperative Lernszenarien werden in [134] oder [83] durchgeführt.

Verteilt gespeichertes Wissen kann aber nur dann erfolgreich zum Lernen eingesetzt werden, wenn es bei Bedarf auch gefunden werden kann. Zudem erfordert die Suche nach Lernmaterial

im Web detailliertere Suchkriterien als sie für herkömmliche Dokumente nötig sind. Während bei einer konventionellen Informationssuche der Schwerpunkt einer Suche nur darauf liegt, wo die gewünschte Information zu finden ist, ist es bei der Suche nach Lernmaterialien auch noch von Bedeutung, wie die Information, die in einem Dokument oder in einer multimedialen Präsentation enthalten ist, dem Benutzer präsentiert wird.

Eine Aufteilung des Wissens in verteilt gespeicherte Einheiten ermöglicht nicht nur die Individualisierung des Lernens, indem nur die Teile bearbeitet werden, die auf eine bestimmte Lernsituation oder das Vorwissen des Lerners abgestimmt sind. Es lassen sich zudem gezielt einzelne Teile leichter aktualisieren, ohne dass immer eine komplette Einheit physikalisch verfügbar sein und geändert werden muss. Ein weiterer wichtiger Aspekt, der sich aus der Aufteilung des Wissens in unabhängige Einheiten oder Module ergibt, ist die Möglichkeit einzelne Teile wiederzuverwenden. Dies ist nicht möglich, wenn die komplette Kurseinheit aus einer monolithischen Struktur besteht. Kommerzielle Autorensysteme für multimediales Lernmaterial wie Authorware Attain und Flash bieten zwar die Funktionalität an, externes Material in die eigene Präsentation mit einzubinden; Teile einer Authorwarepräsentation können hingegen nicht aus der vollständigen Anwendung herausgelöst und unabhängig wiederverwendet werden.

Selbst bei der alleinigen Verwendung von textbasierten Materialien ergibt sich durch die Wiederverwendung eine enorme Kostenersparnis, da die Konzeption und Erstellung von Schulungsmaterialien nur durch hochqualifizierte und entsprechend teuren Experten erfolgen kann. Der Versuch, die Anforderungen von Pädagogen und Lernpsychologen an computerbasierten Lernmaterialien, in Form von häufiger und gezielter Verwendung von multimedialen Techniken bei der Entwicklung des Materials umzusetzen, steigert die Bedeutung der Herstellungskosten der Schulungsmaterialien zusätzlich. Tschirz tritt in [19] (zitiert nach [113]) die Ansicht, dass aufgrund der hohen Kosten für die Produktion multimedialer Materialien die gesamte Entwicklung über die kommerzielle Weiterbildung laufen wird:

“Nur in der Weiterbildung lässt sich relativ schnell Geld verdienen. Die teure Produktion von Multimedia muss finanziert werden, also wird sie über die Weiterbildung laufen müssen. Man fängt mit der Weiterbildung an, um genügend Erfahrungen im Multimediageschäft zu kriegen und setzt die Produkte dann vielleicht langsam auch in den grundständigen Studienangeboten ein.”

Da für deutsche Hochschulen der kommerzielle Weiterbildungsmarkt bisher nur eine untergeordnete Rolle spielt, ist der konsequente Einsatz von Multimedia in Form von Schulungsmaterial und der Videoaufzeichnung von Vorlesungen an deutschen Universitäten im Allgemeinen noch nicht sehr weit verbreitet. Im Bereich der Informatik ist die Situation an deutschen und internationalen Universitäten anders. Dies liegt daran, dass Multimedia selbst ein Forschungsgebiet der Informatik ist und daher auch die Möglichkeiten und der Einsatz multimedialer Animationen in der Lehre wissenschaftlich untersucht und in der Praxis verwendet werden (siehe bsp. [111] oder [112]).

Ähnlich verhält es sich mit dem Einsatz von Tele-Teaching. Beginnend mit der regelmäßigen Übertragung und Archivierung von Vorlesungen und Seminaren wurde von den Universitäten Mannheim, Heidelberg und Karlsruhe seit 1995 die Möglichkeit für Informatik-Studenten

geschaffen, an Vorlesungen der jeweiligen anderen Universitäten teilzunehmen und Prüfungen abzulegen. Die Evaluation dieses Projekts ist mittlerweile abgeschlossen und im Rahmen der Virtuellen Hochschule Oberrhein (VIROR) in den Regelbetrieb übergegangen [141]. Zusammen mit anderen Projekten, wie der Einrichtung digitaler Hörsäle in Darmstadt oder anderswo, stehen damit stabile Methoden und Werkzeuge zur Verfügung, die die Erstellung und Übertragung von Vorlesungen, Seminaren und anderen multimedialen Lernmaterialien ermöglichen [91]. Neben der Möglichkeit für Studierende einer Universität Vorlesungen oder Seminare über Wissensgebiete zu belegen, die an der eigenen Universität nicht angeboten werden, eröffnet der Einsatz der Kommunikationstechnik beim Lernen zusätzlich den Austausch von Lernmaterial zwischen Universitäten. So können Professoren, die ihre Vorlesung inhaltlich abgestimmt haben, komplette Kurse oder abgeschlossene Teile austauschen und wiederverwenden.

Neben den technischen Voraussetzungen ist also eine Vereinheitlichung und Anerkennung der angebotenen Lerninhalte ein zweiter entscheidender Punkt für den Erfolg einer Wieder- und Weiterverwendung von multimedialen Lernmaterial. Kann eine Vorlesung, die in einem digitalen Hörsaal der TU Darmstadt gehalten wird, in Mannheim zwar empfangen werden, eine Prüfung oder die Anerkennung einer Teilnahme hingegen nicht möglich ist, wird die Bereitschaft der Mannheimer Studenten, die Vorlesung zu verfolgen, sehr gering sein. Während eine Anerkennung durch Credit-Point-Systeme und Evaluationsarchitekturen realisiert werden kann, erfordert eine Vereinheitlichung eines Wissensgebietes eine Abstimmung und Einigung darüber, aus welchen Teilen ein Wissensgebiet besteht und wie diese Teile zu strukturieren sind. Im Bereich der Informatik wird in Deutschland durch das Projekt ULI eine solche Vereinheitlichung angestrebt [140]. Dabei wird in diesem Projekt davon ausgegangen, dass der Austausch kompletter Vorlesungen nur in Einzelfällen sinnvoll ist; vielmehr wird angenommen, dass feingranularere, abgeschlossene und unabhängige Module, wie multimediale Animationen, Simulationen oder abgeschlossene Kapitel ausgetauscht und in den jeweiligen Vorlesungen verwendet werden. Schulmeister geht in [113] ebenfalls davon aus, dass die Einrichtung von virtuellen Studiengängen nur dann Sinn macht, wenn sie stark modularisiert sind. Auf diese Weise werden Bibliotheken frei zugänglicher multimedialer Bausteine entstehen, aus denen man sich nach Bedarf bedienen und die Bausteine in größere Einheiten einbinden kann, sodass trotz Wiederverwendung von Teilen individuelle Vorlesungen erstellt werden können. Da sich diese Bibliotheken von herkömmlichen digitalen Bibliotheken nur durch die Größe und die thematische Ausrichtung des Materials unterscheiden, sind auch hier die Probleme der langfristigen Archivierung und der Auszeichnung des Materials mit beschreibenden Daten von großer Bedeutung. Übersteigt die Anzahl multimedialer Lernbausteine, die von einer Bibliothek verwaltet werden, eine bestimmte Größe, ist ohne die Auszeichnung und separate inhaltliche Beschreibung der Module eine effektive Suche nach individuell angepaßten Bausteine nicht mehr möglich.

Die Schaffung virtueller Studiengänge und die Vereinheitlichung von Lernmaterial fördert nicht nur die Austauschbarkeit, sondern führt auch zu einer Konkurrenzsituation zwischen Universitäten. Multimediale Kurse, Prüfungen und Seminare sind durch die ständige Erhöhung der verfügbaren Bandbreite und die Entwicklung bei mobilen Kommunikationsgeräten überall und zu jeder Zeit verfügbar. Damit konkurrieren Universitäten nicht mehr nur national um Studenten, sondern sehen sich zunehmend einem internationalen Wettbewerb ausgesetzt. Internationale Universitäten bieten komplette Studiengänge online an, die keine Präsenzveranstaltungen mehr beinhalten und bei denen auch die Prüfung nur noch rechnerge-

stützt durchgeführt werden (beispielsweise das Massachusetts Institute of Technology (MIT) [88]). Dabei bleibt für Universitäten die Situation insoweit unverändert, dass immer noch die Qualität der Lehre als wichtiges Kriterium bei der Beurteilung einer Lehrveranstaltung verwendet wird. Wenn die Vermittlung der Inhalte allerdings nicht mehr durch Präsenzveranstaltung durchgeführt wird, sondern die Materialien nur noch online zur Verfügung gestellt werden wird bei virtuellen Studiengängen die Qualität des Lernmaterials entscheidend über den Erfolg einer Vorlesung sein. Entscheidend ist also auch in diesem Fall dass die Materialien zum einen in der zunehmenden Informationsflut überhaupt noch gefunden werden können und dass detaillierte Beschreibungen über die Inhalte vorliegen, damit potentielle Lerner vorher entscheiden können, ob sie einen komplette Kurs oder ein einzelnes Modul bearbeiten und damit auch u.U. bezahlen möchten. Eine Handelsplattform für Module kann durch die Errichtung virtueller Marktplätze für Lernmaterialien realisiert werden, die im nächsten Abschnitt beschrieben wird. Anhand der Beschreibung eines virtuellen Marktplatzes lässt sich auch zeigen, wie viele unterschiedliche Rollen an der Erstellung, Distribution und Bearbeitung des Materials beteiligt sind. Entsprechend vielfältig sind daher auch die Anforderungen, die die Rollen an die Handhabung des Lernmaterials stellen.

1.2 Virtuelle Marktplätze für Lernmaterialien

Offene Marktplätze, auf denen Universitäten kooperieren und konkurrieren, um individuell angepasste computergestützte Weiterbildung oder virtuelle Studiengänge anbieten zu können, werden mittlerweile nicht mehr nur als eine Vision gesehen [69], sondern werden in einfacher Form schon heute betrieben [129], [138], [139]. Dabei unterscheiden sich virtuelle Marktplätze von bereits bestehenden virtuellen Universitäten dadurch, dass auf ihnen Material von beliebigen Anbietern gehandelt wird und nicht nur von einer zentralen Einrichtung¹. Zum Aufbau virtueller Universitäten existieren mittlerweile kommerzielle Kurs- und Studenten- und Prüfungsverwaltungssysteme ([96],[26]). Die Rollen, die von diesen Systemen unterstützt werden, beschränken sich dabei auf die Universitäten als Anbieter kompletter Vorlesungen und auf Studenten als Konsumenten. Die Hauptfunktion eines Marktplatzes, den Handel mit Lerneinheiten unterstützten solche Systeme nicht.

Das Zusammenspiel verschiedener Anbieter und Nachfrager bei einem virtuellen Marktplatz erfordert ein Kooperationsmodell zwischen Anbietern von Lerninhalten, Händlern und Konsumenten. Ein einfaches Geschäftsmodell, dass die Kommunikation zwischen den auf dem Marktplatz agierenden Rollen unterstützt, ist in [68] beschrieben und besteht aus den nachfolgend aufgeführten 5 Schritten:

- 1 Ein Autor erstellt multimediales Lernmaterial und übergibt es an einen Rechteinhaber, bsp. an ein Online-Verlagshaus
- 1 Der Rechteinhaber testet und bewertet das Material und gibt es dann zusammen mit Zugriffsrechten und Preisen für die Nutzung an den Betreiber des virtuellen Marktplatzes
- 1 Der Betreiber macht das Lernmaterial auf dem Marktplatz verfügbar
- 1 Nachfrager, die entweder Endbenutzer oder Entwickler von Lernmaterialien sein können, haben Zugriff auf das Material und können das Material nach einem Kauf oder ohne

1. Einen ausführliche Auflistung bestehender virtueller Universitäten findet sich bsp. bei [113]

Unabhängig davon, welches der vorgestellten Modelle in der Praxis verwendet wird, wird auch in diesem Szenario deutlich, dass Lernmaterialien mit einer Vielzahl von zusätzlichen Informationen ausgestattet sein müssen, bevor sie auf einem virtuellen Marktplatz handelbar sind. Zudem erfordert die Vielzahl der Rollen, die von der Erstellung bis hin zur Auslieferung des Materials beteiligt sind, eine Standardisierung der Austauschformate der Materialien.

Diese Standardisierung muss dabei sowohl auf einer technischen, als auch einer inhaltlichen Ebene erfolgen. Während eine Festlegung, welche Inhalte in Lernmaterialien enthalten sein sollen, damit sie von verschiedenen Universitäten wechselseitig anerkannt werden, beispielsweise durch Projekte wie das im letzten Abschnitt beschriebene ULI, erfolgen kann, fehlt bisher auch ein Standard, wie diese Inhalte beschrieben oder die Beschreibungen ausgetauscht werden können.

1.3 Anforderungen an multimediales Lernmaterial

Informationssysteme mit denen ein personalisierter und zeitunabhängiger Zugriff auf Lernmaterialien sichergestellt werden kann, erfordern die Erfassung und das Management einer Vielzahl unterschiedlicher Informationen [145]. Zusammengefaßt ergeben sich durch die oben dargestellten Szenarien folgende technische Anforderungen an multimediales Lernmaterial.

- 1 Lernmaterial soll zu jeder beliebigen Zeit an jedem Ort verfügbar sein.
- 1 Lernmaterial soll aus unabhängigen modularen Einheiten bestehen, die wiederverwendet, beliebig kombiniert und zu größeren Einheiten oder Kursen zusammengestellt werden können.
- 1 Lernmaterial soll aufgrund von Bedürfnissen und Wünschen der Lernenden beispielsweise in Bezug auf Lernziel, Medienpräferenz, Lehrmethode und Schwierigkeitsgrad ausgewählt und individuell zusammengestellt werden können.
- 1 Lernmaterial, das Teil eines Kurses ist, soll unabhängig von den anderen Teilen ausgetauscht oder aktualisiert werden können.
- 1 Lernmaterial soll auf Eignung hin bewertet werden können, ohne über die Ressource selbst verfügen zu müssen.
- 1 Lernmaterialien sollen mit Zertifikaten oder Nutzungsinformationen versehen werden können.
- 1 Beschreibungen über Lernmaterialien sollen über eine standardisierte Kodierung zwischen beliebigen Anwendungen ausgetauscht werden können.

1.4 Ziel und Aufbau der Arbeit

Das Hauptziel der vorliegenden Arbeit besteht darin zu untersuchen, inwieweit heute bestehende Ansätze zur Beschreibung multimedialer Lernmaterialien geeignet sind, die oben aufgezählten Anforderungen zu erfüllen. Aufbauend auf diesen Untersuchungen, wird ein Datenmodell für web-basierte Lernsysteme vorgeschlagen. Zudem soll aufgezeigt werden, welche Konsequenzen sich durch die Verwendung von standardisierten Metadatenbeschreibungen

gen und die Aufgaben der beteiligten Rollen für die Architekturen von zukünftigen web-basierten Lernsystemen ergeben.

Kapitel 2 gibt einen Überblick über den aktuellen Stand im Bereich der Entwicklung von Beschreibungsformaten für allgemeine multimediale Ressourcen. Der derzeit vielversprechendste Ansatz zur Beschreibung von multimedialer Lernmaterialien, der Learning Object Metadata (LOM) -Entwurf der IEEE ist Inhalt des Kapitels 3. In den ersten Abschnitten von Kapitel 3 wird LOM detailliert beschrieben und auf die Möglichkeiten, die Beschreibung einer Lernressource mit LOM zu kodieren eingegangen. Insbesondere wird durch die Definition eines austauschbaren Beschreibungsformat die Entwicklung der in den weiteren Kapiteln beschriebenen Werkzeuge zur Entwicklung und Verwaltung einer Wissensbasis für multimediale Lernressourcen vorbereitet. Im Hinblick auf die große Menge an Information, die ein Autor bei der Auszeichnung einer Ressource mit LOM bei der Beschreibung zur Verfügung stellen muss, befasst sich ein signifikanter Teil der Untersuchung mit den Möglichkeiten, einzelne Elemente der Metadatenbeschreibung automatisch zu generieren.

Bei der Bewertung von LOM wird sich zeigen, dass der Entwurf für die manuelle Suche nach Ressourcen sehr gut geeignet ist. Es ist weiter möglich, eine multimediale Lernressource zu beurteilen, ohne über sie zu verfügen. Entsprechend kann also durch LOM eine Vorauswahl getroffen werden, wenn Bausteine gesucht werden, die keine kompletten Kurse darstellen, sondern Module, die in eine bestehende Struktur eingebunden werden, die aber vorher noch einer Evaluation unterliegen. Eine automatisierte Kombination von Bausteine zu einer größeren Einheit ist hingegen nicht möglich, da LOM nicht genug Informationen bereitstellt, um daraus semantische Strukturen abzuleiten oder zu entwickeln.

Aus diesem Grund werden in den sich anschließenden Abschnitten semantische Netze beschrieben, die zur Speicherung von Wissenstrukturen, die unabhängig von konkreten physikalischen Ressourcen existieren, verwendet werden. Zusätzlich wird gezeigt, dass nur durch eine Kombination von Metadatenentwürfen wie LOM und semantischen Netzen alle Voraussetzungen für beliebig wiederverwendbare und vor allem in beliebigen Kontexten einsetzbare Medienbausteine erfüllt werden können, die zum Einsatz solcher Bausteine in einem web-basierten Lernsystem benötigt werden.

In den folgenden Abschnitten werden dann Anforderungen beschrieben, die an Implementierungen eines web-basierten Lernsystem, das als Basis LOM und semantische Netze hat, gestellt werden. Aufbauend auf diesen Anforderungen wird dann ein Datenmodell und eine Architektur für eine Wissensbasis entwickelt, die als Basis für ein web-basiertes Lernsystem dienen kann. Im Gegensatz zu Ansätzen, die auf eine Vereinheitlichung der Datenmodelle abzielen [32], [9], wird in dieser Arbeit eine getrennte Kodierung und Speicherung der semantischen Modelle und der Beschreibungen der Lernmaterialien vorgeschlagen. Zusätzlich zur Basisarchitektur des Lernsystems werden Werkzeuge identifiziert, die zum Aufbau einer Wissensbasis benötigt werden.

Die Kapitel 4 und 5 beschreiben die Entwicklung und den Einsatz von Autorenwerkzeugen, mit denen der Aufbau einer Wissensbasis für die Verwaltung von multimedialem Lernmaterial möglich ist. Kapitel 6 fasst die Arbeit zusammen und gibt einen Ausblick über zukünftige Entwicklungen.

2 Metadaten für Web-Ressourcen

2.1 Einleitung

Hochschulen, die ihre multimedial aufbereiteten Kurse oder Animationen weiterverwenden oder externen Personen über das Web zur Verfügung stellen wollen, stehen vor dem Problem der Beschreibung der Materialien. Das Problem Informationseinheiten oder Module so zu beschreiben, daß sie einfach gefunden, bewertet, kombiniert und individuell verwendet werden können, ist nicht nur auf den Bereich des multimedialen Lernens beschränkt sondern existiert für alle im Web verfügbaren Informationen und ist dort unter dem Begriff "information overload" bekannt. Die Hauptgrund für den Erfolg des WWW- die Einfachheit mit der man HTML-Dateien erstellen, zur Verfügung stellen und mit bestehenden Informationen verbinden kann - stellt sich mittlerweile auch das Hauptproblem bei der Nutzung des WWW zur Suche nach Informationen dar. Vor der Nutzung des WWW als universellem Informationsspeicher waren die Hauptprobleme bei der Informationssuche festzustellen, ob die Information elektronisch verfügbar ist und wie der Zugriff erfolgen kann. Mittlerweile ist nahezu jede gewünschte Information über das Web erhältlich und sofort mit einem gängigen Web-Browser darstellbar; das Problem ist jetzt die Information in der unüberschaubaren und ständig wachsende Menge an Informationen und Dokumenten, die im Web zur Verfügung stehen, zu finden [128]. Diese Situation führt dazu, dass das viele Dokumente und damit auch Lernmodule in der Informationsflut versinken und nicht mehr gefunden werden. Das beste Beispiel hierfür sind die Ergebnisse auf eine Suchanfrage bei einer gängigen Suchmaschine

Wird als Suchbegriff ein weit verbreiteter Begriff wie beispielsweise das Bildkomprimierungsverfahren JPEG verwendet, so geht die Anzahl an Treffer, die die Suchmaschine zurückliefert, in die Tausende. Die Zahl der Dokumente, die sich direkt mit dem Thema JPEG beschäftigen, mag auch heute im Web noch überschaubar sein. Die Zahl der Dokumente, bei denen der Begriff lediglich verwendet wird, um im Kern aber andere Informationen bereitzustellen, führt dazu, dass die Resultatsmenge eine unübersichtliche Größe erreicht. Der Grund hierfür ist, dass die Suchmaschinen nur sehr einfache Indizierungsverfahren bei der Informationssammlung über Web-Ressourcen verwenden. In der Regel findet nur eine reine Volltextindizierung über HTML-Seiten statt [17]. Zusätzlich werden nur bestimmte Informationen der Seite ausgewertet. Eine grundlegende Filterung ist beispielsweise, dass nur die Teile einer HTML-Seite ausgewertet werden, die innerhalb von Paragraphen-Tags stehen. Würden diese Filter nicht verwendet, dann wäre jede Web-Seite, die ein Bild im JPEG-Format anzeigt, ein Resultat als Antwort auf eine Anfrage nach JPEG.

Dabei haben Texte den großen Vorteil, dass sie relativ leicht analysiert werden können. Es existiert eine Vielzahl von Algorithmen zur Textanalyse und vor allem zur effizienten Speicherung der Textindizes [41]. So kann beispielsweise bei der Indizierung von Dokumenten die Anzahl des Auftretens eines Begriffes, bei der Beurteilung über die Relevanz eines Dokumentes herangezogen werden. Web-Seiten, die aus einer Liste an Bildern bestehen, deren Format jeweils aber auch in einer textbasierten Beschreibung auf der Seite steht, können aber auch durch solche Verfahren nicht ausgefiltert werden können. Die grundlegende Limitierung dieser Analyseverfahren liegt also darin, dass es keine Möglichkeit gibt zu spezifizieren, welche Bedeutung ein Suchbegriff für ein Dokument, in dem er vorkommt, haben soll. Zudem können

alle Indizierungsverfahren nur bei Textdokumenten verwendet werden. Handelt es bei einer web-basierten Ressource um eine Kombination multimedialer Komponenten, so müssen völlig andere Analyseverfahren verwendet werden, da bei diesen Medien kaum untersuchte Elementarbausteine wie Wörter zur Verfügung stehen. Dabei wird unter einer web-basierten Ressource jegliche digitale Informationseinheit verstanden, auf die mit Hilfe eines gängigen Web-Browsers zugegriffen werden kann.

Eine Indizierung von Videomaterial kann durch das Hinzufügen einer Metaspur erfolgen, auf der die Beschreibungen der einzelnen Szenen gespeichert werden. Ein Bezug wird also über eine zeitbasierte Referenz zwischen Text und Inhalt (Audio und Video) hergestellt. Dieser Prozeß ist sehr zeitintensiv [39]. Er unterliegt zudem einer subjektiven Interpretation des Inhalts durch den Autor der Beschreibung.

Die auf eine automatische Bearbeitung ausgerichteten Ansätze werden unter dem Stichwort Content Processing zusammengefaßt [127]. Zur Analyse von Bild- und Filmmaterial greifen einige Systeme auf einfache visuelle Merkmale wie Farbe, Umrisse, Texturen und Bewegung zurück. Erfolgreich werden visuelle Indikatoren beispielsweise als Input für neuronale Netze zum Aufspüren von Gesichtern in Filmen verwendet. Diese Technik ist allerdings nur sehr eingeschränkt in der Lage, z.B. Personen zu identifizieren, und liefert nur einen vorverarbeitenden Schritt, einen Hinweis, an welcher Stelle im Videostrom gesprungen werden muss, um die beteiligte Person zu sehen. In [75] werden diese Techniken verwendet, um eine automatische Genreerkennung bei Videos zu erreichen. Auch im Bereich des digitalen Audio erweist sich die automatische Inhaltsanalyse mittels neuronaler Netze als ein Weg zur Vorverarbeitung. Hierbei werden aus natürlicher Sprache sogenannte Transcripts in Textform erstellt. Liegt eine Textfassung der Tonspur vor, so kann sie zur Recherche des Audioinhalts eines Films verwendet werden. Gemeinsam ist beiden Ansätzen, dass die elementaren Bestandteile auf denen die Inhaltsanalyse basiert vom System erst erlernt werden müssen. Eine Anzahl einfacher Audio- und Videoindikatoren lassen sich automatisch aus digitalen Daten extrahieren und erweisen sich dabei als sehr gute und wichtige Ergänzung zu textuellen Informationen. Diese Indikatoren werden jedoch bislang meist unabhängig voneinander zum Zugriff auf multimediale Daten verwendet. Zudem bieten diese einfachen Indikatoren nur wenig direkten Bezug zu semantischen Konzepten, wie sie für eine Suche benötigt werden. Neuronale Netze lernen zwar die geometrischen Regelmäßigkeiten, die ein Gesicht mit großer Wahrscheinlichkeit ausmachen, die Bedeutung des erlernten Konzepts "Gesicht" und damit die inhaltsbezogene Einbettung in seine im Bild oder im Film festgehaltene Umgebung fehlt. Es erweist sich daher von Nutzen, ein fundiertes Modell zur Abbildung der realen Welt zu verwenden, mittels dessen die aus Audio und Video extrahierten Informationen derart gruppiert werden können, daß eine Erkennung von Sachverhalten an der realen Welt orientiert erfolgen kann. Das Problem der Extraktion von Information aus den Rohdaten wird noch gravierender, wenn man die Struktur der im Web verfügbaren Ressourcen betrachtet. Üblicherweise bestehen web-basierte Ressourcen nicht nur aus einem Medium, sondern aus einer Kombination von Texten, Bildern, Videos oder auch Animationen, für die bisher überhaupt keine Analyseverfahren existieren. Die Lösung dieser Probleme stellen Metadaten zur Beschreibung der Ressourcen dar.

2.1.1 Definition von Metadaten

Die Wort Metadaten wird erst seit ungefähr 15 Jahre verwendet. Mit steigender Popularität des World Wide Webs hat der Begriff zunehmend an Bedeutung gewonnen. Metadaten sind defi-

niert als „**strukturierte Daten über Daten**“. [60] Sie stellen beschreibende Informationen über z.B. Ressourcen, Inhalte und Benutzerprofile zur Verfügung. Metadaten alleine stellen für den Benutzer keinen Nutzen dar. Sie werden es dann nützlich, wenn sich die gespeicherten Informationen auf ein konkretes Objekt beziehen. Diese sehr einfache Definition beinhaltet ein breites Spektrum an Möglichkeiten: von manuell generierten textlichen Beschreibungen einer Ressource bis hin zu maschinell generierten Daten, die nur von entsprechenden Softwareanwendungen ausgewertet werden können. Metadaten verwenden in der Regel mehr als ein Beschreibungselement für die Speicherung von Informationen über eine Ressource. Ein gängiges Beispiel hierfür ist eine Visitenkarte, die neben dem Namen des Besitzers auch noch Kontakt-Informationen wie seine Email-Adresse oder Telefonnummer enthält. Durch Visitenkarten ist neben den eigentlichen Informationen, zusätzlich auch noch die Struktur, in der die Information kodiert ist, festgelegt. Metadaten sind kein neues Konzept. Sie werden von Bibliotheken seit langem zur Katalogisierung von Dokumenten verwendet. [37] erweitert die obige Definition beispielsweise mit dem Zusatz: „Es handelt sich dabei typischerweise um bibliographische Daten“. Unter bibliographischen Daten werden dabei Informationen über den Autor, den Verlag, das Erscheinungsjahr und den Titel des Dokumentes verstanden. In Form von Karteikarten werden diese Informationen seit langem für die Suche nach Dokumenten verwendet.

2.1.2 Zweck von Metadaten

Metadaten können für unterschiedliche Zwecke verwendet werden. Eine Aufstellung des Verwendungszwecks von Metadaten für web-basierte Ressourcen findet sich in [63], die auszugsweise in der folgenden Aufzählung aufgeführt ist.

- 1 Zusammenfassung der Bedeutung des Dokuments
- 1 Realisierung einer Zugriffskontrolle auf das Dokument
- 1 Suchmöglichkeiten für Medien, die nicht direkt analysiert werden können.
- 1 Informationen über die Verwendung der Ressource, wie Hardware- oder Softwarevoraussetzungen
- 1 Informationen über den Status oder die Version eines Dokuments
- 1 Speicherung von Informationen über den Autor der Ressource oder anderen Kontaktinformationen für Anfragen bezüglich der Ressource
- 1 Informationen über Beziehungen einer Ressource zu anderen Ressourcen

Ein Beispiel einer Metadatenbeschreibung für ein gedrucktes Buch zeigt Tabelle 1

Metadatenelement	Wert
Titel	Multimedia-Technologie
Autor	Ralf Steinmetz

Tabelle 1 Beispiel für eine Metadatenbeschreibung

Metadatenelement	Wert
Verlag	Springer Verlag
ISBN-Nummer	3-540-67332-6
Erscheinungsjahr	2000

Tabelle 1 Beispiel für eine Metadatenbeschreibung

Die Auflistung zeigt die vielfältigen Anforderungen die an Metadaten gestellt werden. Welche direkten Eigenschaften einer Ressource überhaupt beschrieben werden können, hängt direkt davon ab, in welchem Medienformat die Ressource vorliegt. Während bei digitalen Bildern Farbverläufe, oder bei Videos Szenenfolgen, mit Hilfe von Metadaten beschrieben werden können, ist dies bei textbasierten Ressourcen nicht möglich. An dem einfachen Beispiel der Metadatenbeschreibung eines Bildes erkennt man auch daß bei gleichem Medienformat unterschiedliche Eigenschaften einer Ressource für die Beschreibung mit Metadaten in Frage kommen. Soll es anhand von Metadatenbeschreibungen möglich sein, für ein bestimmtes Bild, ähnliche Bilder im Sinne von ähnlichen Lichtverhältnissen oder ähnlichen Motiven zu finden, werden sehr detaillierte Eigenschaften wie Farbverläufe, Umrisse oder andere visuelle Merkmale benötigt. Geht es beispielsweise aber darum jugendgefährdende Bilder zu identifizieren, so sind diese detaillierten Informationen nicht hilfreich. In diesem Fall muß eine Einordnung des Bildes in ein bestimmtes Genre oder die Zuordnung einer Altersgruppe zu dem Bild erfolgen. Je nach Anforderung, die an eine Beschreibung einer Ressource gestellt wird, müssen also unterschiedliche Eigenschaften der Ressource beschrieben sein. Das Medienformat einer Ressource bestimmt aber nicht nur, welche Eigenschaften überhaupt beschrieben werden können sondern wirkt sich auch direkt auf die Möglichkeit der Kodierung und der Zuordnung einer Metadatenbeschreibung zur entsprechenden Ressource aus. (vgl. Abschnitt 2.6 "Kodierung von Metadaten"). So lassen sich Metadatenbeschreibungen, die sich auf textbasierten Ressourcen beziehen leichter direkt in einer Ressource speichern, als dies bei Bildern der Fall ist.

Die Auswahl, welche Eigenschaften durch Metadatenbeschreibungen erfaßt werden sollen, wird in einer strukturellen Ebene festgelegt (vgl. Abschnitt 2.2.1 "Strukturelle Ebene"). Dabei wird die Auswahl der Eigenschaften wie oben erwähnt durch das Medienformat der Ressource zwar beeinflußt und kann daher nicht unabhängig von den Medienformaten der Ressourcen getroffen werden, bedingt die Eigenschaften die durch Metadaten erfaßt werden können aber nicht vollständig. So können die in Tabelle 1 gezeigten Eigenschaften, wie im Beispiel gezeigt, für die Beschreibung von Büchern verwendet werden. Es ist aber ebenfalls problemlos möglich diese Merkmale Bildern oder Videos zuzuweisen. Die Auswahl welche Eigenschaften einer Ressource beschrieben werden in Verbindung mit dem Medienformat der Ressource bestimmen hingegen direkt um welche Art von Metadaten es sich handelt (siehe Abschnitt 2.3 "Arten von Metadaten"). Diese Kombination bestimmt daher auch, welche Eigenschaften direkt aus den Ressourcen extrahiert werden können und welche durch manuellen Aufwand ermittelt werden müssen.

Während auf der strukturellen Ebene festgelegt ist, welche Eigenschaften der Mediendaten beschrieben werden, wird auf der begrifflichen Ebene festgelegt, wie die Mediendaten beschrieben werden. Das "wie" bezieht sich hierbei aber nicht auf die Kodierung der Metdaten (siehe Abschnitt 2.6 "Kodierung von Metadaten"), sondern auf die Bezeichner und die Wertebereiche der Elemente. Im Gegensatz zur strukturellen Ebene und der Wahl des Kodierungs-

format, ist die begriffliche Ebene völlig unabhängig vom Medienformat der Ressource. Das oben aufgeführte Beispiel zeigt zum einen die Wichtigkeit der Strukturinformationen einer Metadatenbeschreibung. Erst durch die eindeutige Zuordnung der einzelnen Werte der Beschreibung zu eindeutigen Metadatenelementen, die auch die Attribute des Schemas genannt werden, können die Informationen der Beschreibung sinnvoll verwendet werden. Dieses Beispiel zeigt aber auch, dass die richtige Wahl der Metadatenelemente entscheidend für die spätere Verwendung ist. Das in Tabelle 1 beschriebene Buch existiert beispielsweise in mehreren Auflagen. Steht für die Speicherung der Auflage des beschriebenen Buchs kein Metadatenelement zur Verfügung, können verschiedene Auflagen des Buchs nicht unterschieden werden.

Die Kombination der Auswahl der strukturellen Ebene und der begrifflichen Ebene bilden zusammen ein Metadatenschema. Dabei kann die Auswahl eine spezielle Kodierung der Metadatenbeschreibungen bedingen; es ist aber auch möglich, daß das Metadatenschema keine Auswirkungen auf das Kodierungsformat hat. Erst nach der Auswahl eines Schemas für die Beschreibung der Ressourcen kann daher, unter Berücksichtigung der vorliegenden Medienformate, über die Kodierung der Metadatenbeschreibung und die Zuordnung der Beschreibung zu den Ressourcen entschieden werden. Die Kombination der Metadaten und der Mediendaten wird in der Abbildung als Medienbaustein bezeichnet; die Definition des Begriffs wird in Abschnitt 2.2.3 gegeben. Aus der Abbildung wird auch ersichtlich, daß es für das gleiche Metadatenschema unterschiedliche Formen der Kodierung geben kann. Dabei wird sich aber in Abschnitt 2.4 "Beispiele für Metadaten" zeigen, daß die Auswahl eines Kodierungsformats nicht unabhängig von den Entscheidungen, die auf der strukturellen Ebene getroffen werden, gewählt werden kann. Während Abbildung 2 den Zusammenhang zwischen den eben eingeführten Begriffen zeigt, folgen in den nächsten Abschnitten detaillierte Informationen zu den Begriffen.

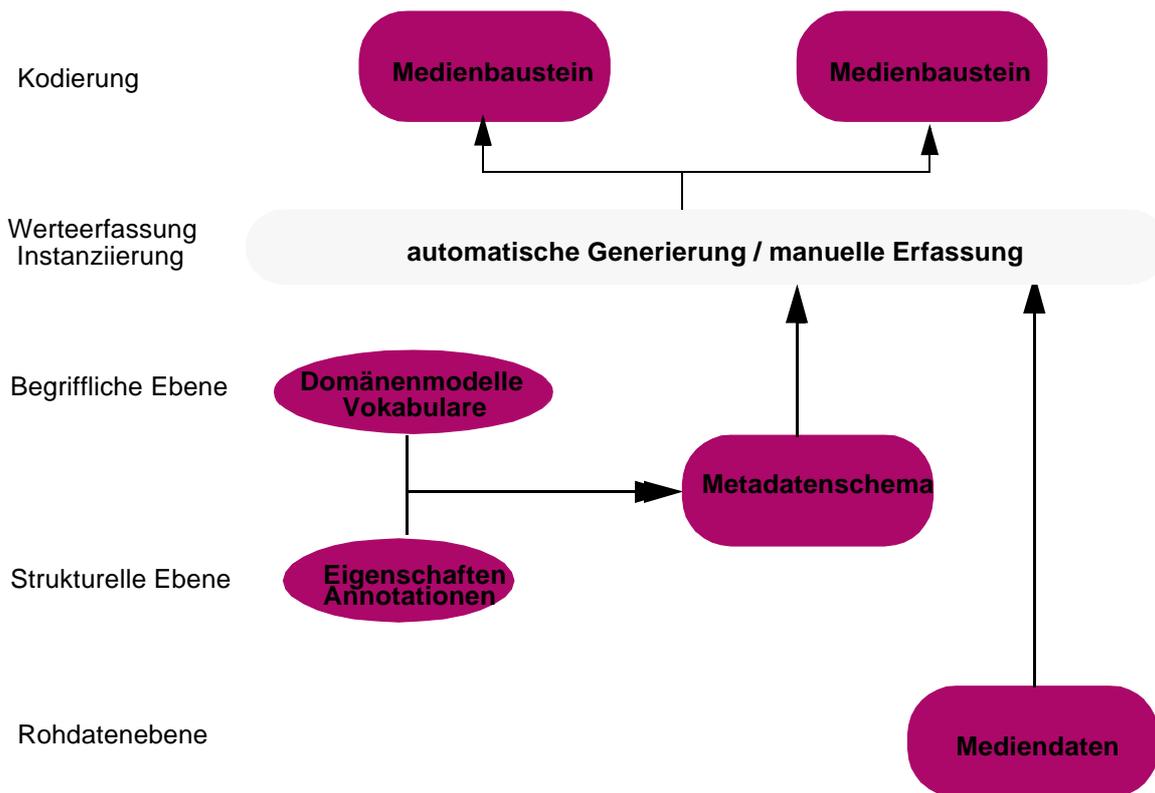


Abbildung 2: Zusammensetzung eines Medienbausteins (vgl.[130])

2.2 Struktur von Metadaten

2.2.1 Strukturelle Ebene

Die Auswahl oder Erstellung eines Metdatenschemas verlangt eine restriktive und präzise Spezifikationsmethodik, um dem Problem der orthogonalen Dimensionen der Beschreibungselemente zu begegnen. Einzelne Metadatenelemente sollten völlig unabhängig von den anderen Elementen des Schemas sein. Ist dies nicht der Fall, können Änderungen bei einzelnen Elementen zu Inkonsistenzen bei der kompletten Beschreibung führen. In [144] wird ein einfaches Beispiel eines Metdatenschemas zur Beschreibung von allgemeinen Dreiecken aufgeführt. Während ein Metadatenelement Farbe, in dem die Farbe des Dreieckes beschrieben ist, vollständig unabhängig von anderen Metadatenelementen wie den Angaben über die Kantenlängen des Dreiecks ist, ist dies bei einem Metadatenelement Fläche nicht der Fall, da sich die Fläche eines Dreieckes aus den Kantenlängen berechnen lässt. Ein anderes Beispiel ist ein Metdatenschema zur Beschreibung von Lernressourcen, das sowohl ein Metadatenelement für den Schwierigkeitsgrad, als auch ein Metadatenelement zur Angabe des Alters der Zielgruppe beinhaltet (vgl. Abschnitt 3.2.10 "Zusammenfassung und Bewertung"). Es ist offensichtlich, daß neben anderen Faktoren, das Alter des Lerner einen erheblichen Einfluß auf den Schwierigkeitsgrad einer Lernressource hat, sodas diese beiden Metadatenelemente direkt

voneinander abhängen und nicht orthogonal sind.

Zusätzlich wird es bei der Suche auf den Metdatenbeschreibungen für Benutzer schwerer zu unterscheiden, auf welchen Feldern nach bestimmten Begriffen gesucht werden soll, wenn nicht klar ersichtlich ist, für welche Informationen die einzelnen Elemente verantwortlich sind. Auf der strukturellen Ebene wird also festgelegt, welche Eigenschaften einer Ressource beschrieben werden sollen. Hieraus ergibt sich dann auch, für welchen Zweck die Metadaten später verwendet werden sollen. Wird beispielsweise ein Schema zur Beschreibung von Bild-daten spezifiziert, das nur Eigenschaften wie Farbverläufe oder Histogramme speichern kann, so wird auf den Metadaten keine inhaltliche Suche anhand von Textbeschreibungen möglich sein.

2.2.2 Begriffliche Ebene

Die begriffliche Ebene eines Metdatenschemas legt sowohl die Bezeichnungen der Eigenschaften eines Medienbausteins und damit auch die Bezeichnungen der einzelnen Metadaten-elemente bzw. Attribute des Schemas fest, als auch die Wertebereiche der einzelnen Eigenschaften. Für eine maschinelle Suche auf Metadaten sind die Bezeichnungen für die Metadatenelemente nur insoweit von Bedeutung sind, als das sie eine eindeutige Zuordnung von Suchkriterien zu Metadatenelementen ermöglichen. Für einen Benutzer, der die Suchkriterien spezifiziert, sind die Bezeichner der einzige Hinweis, welche Eigenschaft der Ressource, durch das entsprechende Metadatenelement beschrieben ist. Die Einschränkung der möglichen Werte für ein Element eines Metdatenschemas stellt immer einen Kompromiß dar. Sie kann zum einen mehrdeutige Beschreibungen vermeiden und Autoren einer Metadatenbeschreibung einen Hinweis für die Verwendung eines Metadatenelements geben. Auf der anderen Seite wird die Vielfalt der Beschreibungsmöglichkeiten eingeschränkt. Wason schreibt in [144] als Argument für die Verwendung von vorgeschriebenen Vokabularen: “Ask ten searchers to find the resource, again, using an unrestricted vocabulary. (..) It is highly unlikely, that all searchers will find the resource, because the catalogers will differ in their descriptions of the resource and the searchers will differ in their search strategies and terminology.”

2.2.3 Definition Medienbaustein

Wie Abbildung 2 zeigt, entsteht durch die Kombination der Mediendaten mit Metadaten, ein Medienbaustein, der gesucht und verwendet werden kann. Der Begriff Medienbaustein betont dabei ausdrücklich, die mögliche Kombinierbarkeit der Ressource mit anderen multimedialen Ressourcen. Obwohl der Begriff Medienbaustein regelmäßig in der Literatur verwendet wird, existiert keine Definition des Begriffs. Caumanns definiert in [24] den Begriff Medienobjekt als: “Jede textuelle, grafische oder sonstige Kodierung von für das behandelte Wissensgebiet relevantem faktischen Wissen”. Digitale Ressourcen benötigen nach dieser Definition immer den Kontext eines bestimmten Wissensgebietes um als Medienobjekt bezeichnet zu werden. Da bei dieser Definition ein bestimmtes Wissensgebiet für die Bezeichnung Medienobjekt nötig ist und keine Aussagen über die unabhängige Wiederverwendung oder Kombinierbarkeit von Objekten gemacht wird, wird der Begriff Medienbaustein, der im weiteren Verlauf der Arbeit verwendet wird, wie folgt definiert:

“Unter einem Medienbaustein versteht man jede digitale Ressource, der eine Metadatenbeschreibung eindeutig zugeordnet werden kann, die unabhängig von der Verfügbarkeit anderer Ressourcen für die Vermittlung von Wissen verwendet und mit anderen Medienbausteinen zu größeren Einheiten kombiniert werden kann.”

2.3 Arten von Metadaten

Metadaten können entsprechend der im letzten Abschnitt beschriebenen vielfältigen Verwendungsmöglichkeiten nach sehr verschiedenen Kriterien unterschieden werden. Denkbar ist es, Metadaten nach ihrem Zweck zu unterscheiden: also zu trennen, welche Metadaten beispielsweise speziell für die Suche nach Informationen vorhanden sind und welche die Informationen über die Zugriffsrechte der Ressource speichern. In [37] werden Metadaten danach unterschieden, welche Elemente, zur Suche nach einer Ressource verwendet werden und welche zum Verstehen der Ressource benötigt werden. In der Literatur findet sich häufig eine Kategorisierung, die als Kriterium die Beziehung zwischen den Metadatenelementen und der beschriebenen Ressource zur Unterscheidung heranzieht. [12] [108]. Demzufolge werden Metadaten nach inhaltsabhängigen und inhaltsunabhängigen Metadaten unterschieden. Diese beiden Kategorisierungen werden in den folgenden Abschnitten 2.3.1 und 2.3.2 beschrieben.

2.3.1 Inhaltsabhängige Metadaten

Hierbei handelt es sich um die Metadaten, die man direkt aus dem Inhalt der eigentlichen Daten z.B. durch Volltextsuche bei textbasierten Bausteinen gewinnen kann. Das Auftreten bestimmter Schlüsselwörtern kann eine wichtige Rolle bei der Extraktion der Metadaten spielen. Auf diesem System basieren Indizierungstechniken für Texte. Ein anderes Beispiel ist die Größe einer Ressource in Bytes.

Inhaltsabhängige Metadaten können weiter in folgenden Kategorien unterteilt werden:

- 1 Direct content-based metadata (Direkt inhaltsbasierte Metadaten)
z.B. Referenzen auf den Text eines Dokuments.
- 1 Content Descriptive (Inhaltsbeschreibende Metadaten)
Inhaltsbeschreibende Metadaten sind Informationen, die auf dem Inhalt der ursprünglichen Daten beruhen, die aber nicht direkt aus ihnen hervorgehen. Am besten lässt sich die Idee, die hinter den inhaltsbeschreibenden Metadaten steckt, anhand eines Beispiels veranschaulichen: Man stelle sich vor, jemand schreibt einen Aufsatz und stellt diesen im Web zur Verfügung. Später hält er einen Vortrag auf einer Konferenz. Der Vortrag basiere auf dem Aufsatz. Obwohl der Aufsatz jetzt auf gewisse Art mit der Konferenz in Verbindung steht, wird im gesamten Aufsatz die Konferenz nicht erwähnt. Das hat zur Folge, daß bei einer reinen Schlüsselwortsuche nach dem Begriff "Konferenz" der Aufsatz übersehen wird. Er kann vom Suchsystem nicht mit der Konferenz in Verbindung gebracht werden. Inhaltsbeschreibende Metadaten könnten jedoch in einem solchen Fall eine Verbindung vom Aufsatz zur Konferenz herstellen, so daß bei Analyse der zum Aufsatz gehörigen Metadaten erkannt wird, daß der Aufsatz etwas mit der Konferenz zu tun hat. Eine Information über ein Textdokument, aus der man entnehmen kann, für wel-

ches Publikum das Textdokument gedacht ist, ist ein ähnliches Beispiel für eine inhaltsbeschreibende Metainformation. Die inhaltsbeschreibende Metadaten werden wiederum in folgenden Kategorien unterteilt:

- u Domain independent metadata (Bereichsunabhängige Metadaten)
Diese Art der Metadaten erfassen Informationen, welche im Dokument präsentiert sind aber unabhängig von der Anwendung oder der Themenbereiche sind. In dieser Kategorie gehören HTML-Typdefinitionen bzw. HTML-Metadaten.
- u Domain specific metadata (Bereichs- oder Themenspezifische Metadaten)
Metadaten dieses Typs beschreiben spezifische und spezielle Informationen für einen bestimmten Anwendungsbereich.

2.3.2 Inhaltsunabhängige Metadaten

Mittels inhaltsunabhängiger Metadaten wird auf Informationen Bezug genommen, die nicht in direkter Abhängigkeit zum Inhalt der beschriebenen Ressourcen stehen. Nach [90] ist eine Gliederung der inhaltsunabhängigen Metadaten in identifizierende und administrative Metadaten möglich. Die eindeutige Identifizierung von Ressourcen wird durch identifizierende Metadaten vorgenommen. Der so entstehende Metadatenatz enthält unter anderem Informationen über den Namen des Autors, das Erstellungsdatum und die Versionsnummer des Dokuments, sowie das Erstellungsdatum der Metadatenbeschreibung. Identifizierende Merkmale gewährleisten also die Unterscheidbarkeit von ansonsten identischem Datenmaterial, wie bsp. den Bildern einer Überwachungskamera. Als administrative Metadaten werden alle Informationen bezeichnet, die zur Verwaltung der zugehörigen Ressourcen dienen.

In [97] wird eine weitere Unterteilung der administrativen Metadaten in Registrierungsdaten und präsentationspezifischen Daten vorgenommen. Unter den Registrierungsdaten versteht man die Informationen, die der korrekten Interpretation der Medien bei der internen und externen Verarbeitung dienen. Bei der Nutzung von Registrierungsdaten ist zu beachten, dass diese zwar inhaltsunabhängig, aber nicht typunabhängig sind. Hieraus folgt, dass zur Beschreibung eines Textdokuments andere Informationen notwendig sind als zur Charakterisierung eines Rasterbildes oder einer Vektorgrafik. Die präsentationsspezifischen Daten eines Mediums sind notwendig, um die Darstellung einer Ressource steuern zu können. Beispiele hierfür sind die Synchronisationsbeziehungen zwischen verschiedenen Medienströmen oder deren Dienstgüte. Die präsentationsspezifischen Daten sind auch ein Beispiel dafür, daß nicht bei allen möglichen Elementen eine eindeutige Zuordnung zu einer Kategorie von Metadaten möglich ist. So leiten sich die Dienstgüteanforderungen für die Präsentation einer Ressource direkt vom Typ und der Größe der Ressource ab. Bei diesen Werten handelt es sich unzweifelhaft um inhaltsabhängige Metadaten In [70] werden inhaltsunabhängige Metadaten, als Metadaten bezeichnet, die vollständig vom Anwendungsgebiet abhängen. Bestimmte Typen von Metadaten sind demzufolge entweder nur für bestimmte Anwendungsgebiete anwendbar oder benutzen anwendungsgebietspezifische Merkmale. Eine zusammenfassende Klassifikation von Metadaten zeigt Abbildung 3

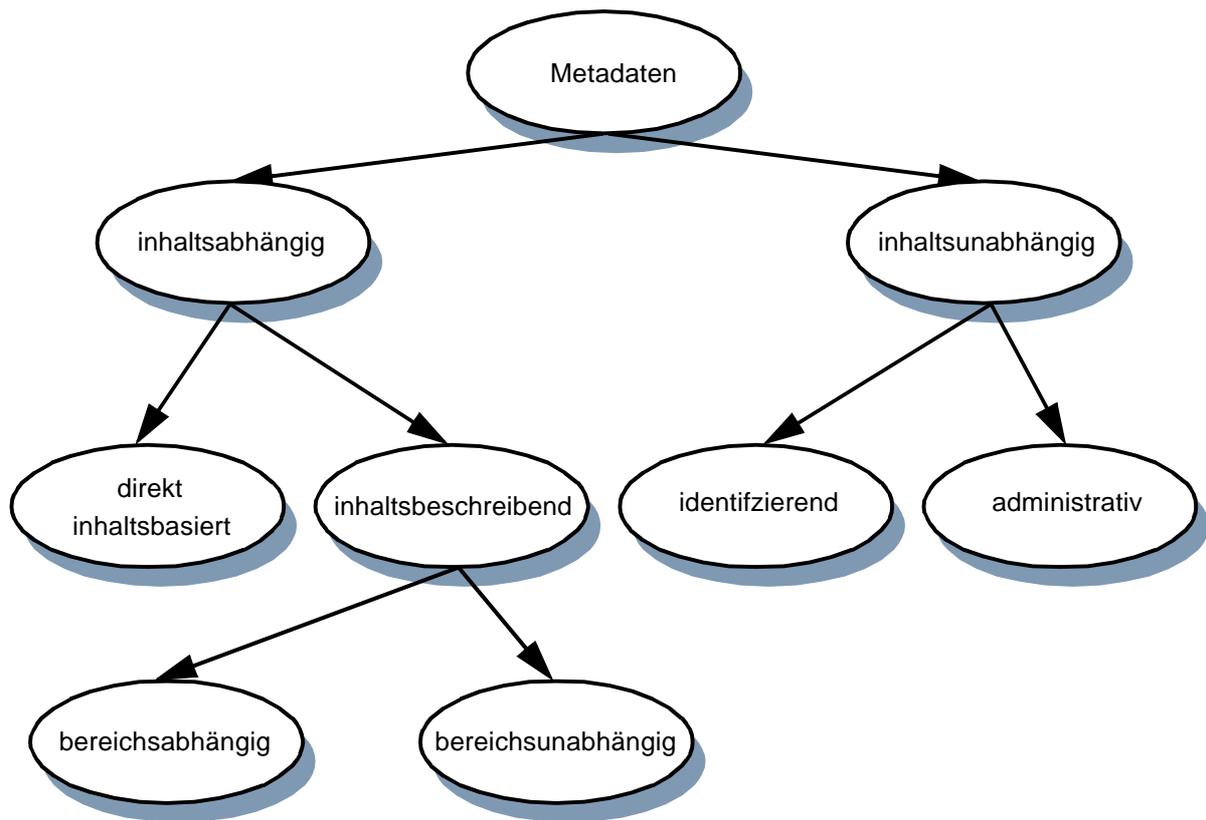


Abbildung 3: Klassifikation von Metadaten

Obwohl jedes Metadatenschema zur Suche und Verwaltung von Ressourcen entworfen wird, ist die Auswahl eines Schemas stark von der Zielgruppe der Ressource abhängig [127]. Die Anzahl und Auswahl von Metadatenelementen für einen Bibliothekar, der ein Video katalogisieren möchte, unterscheidet sich von den Anforderungen einer Universität, die das gleiche Video als wiederverwendbare, multimediale Lernressource anbieten möchte, obwohl beide dieselbe Ressource beschreiben. Im folgenden werden daher Beispiele für weitverbreitete Metadatenschema aus verschiedenen Anwendungsgebieten vorgestellt.

2.4 Beispiele für Metadaten

Wie die Abbildung 2 in Abschnitt 2.1.2 "Zweck von Metadaten" zeigt, setzt sich ein Metadatenschema aus der strukturellen Ebene und der begrifflichen Ebene zusammen. Während ein einfaches Metadatenschema nur generische Spezifikationen auf der strukturellen Ebene enthält, enthalten andere Schema sehr detaillierte Vorschriften, die spezielle Kodierungsformate erfordern. Im folgenden sollen daher Beispiele für Metadatenschema beschrieben werden, die die Bandbreite der verschiedenen Möglichkeiten abdecken. Abschnitt 2.4.1 beschreibt die Platform for Internet Content Selection (PICS), die vom W3C zur Zugriffskontrolle für Ressourcen entwickelt wurde. Der Standard beinhaltet lediglich Konventionen über das Beschreibungsformat und die möglichen Verteilungsmethoden und daher nur Spezifikationen auf der

strukturellen Ebene. Aussagen über zu verwendende Bezeichnungen und deren Vokabular werden nicht getroffen. Es handelt sich daher um die einfachste Form einer Metadatenbeschreibung für Web-Ressourcen. Ein Metadatenschema für allgemeine web-basierte Ressourcen, das detaillierte Spezifikationen auf der strukturellen und der begrifflichen Ebene enthält, aber unabhängig vom Kodierungsformat ist, ist in Abschnitt 2.4.2 "Dublin Core" beschrieben. Das in Abschnitt 2.4.3 "MPEG-7" beschriebene Metadatenschema MPEG-7, wird ebenfalls zur Beschreibung allgemeiner multimedialer Ressourcen entwickelt und weist viele Gemeinsamkeiten mit Dublin Core auf. Im Gegensatz zu Dublin Core enthält dieses Schema allerdings Anforderungen, die die Wahl eines möglichen Kodierungsformats stark einschränken. Die Auswirkungen dieser beiden unterschiedlichen Ansätze werden in Abschnitt 2.6 "Kodierung von Metadaten" aufgezeigt. Abschnitt 2.4.4 "Warwick Framework" schließlich beschreibt kein Metadatenschema im eigentlichen Sinne, sondern ein Rahmenwerk um die Beschreibung einer Ressource durch verschiedene Metadatensätze zu ermöglichen. Mit Hilfe des Warwick Frameworks soll es also möglich sein, einheitliche Zugriffsschnittstellen auf Metadatenbeschreibungen zu ermöglichen, auch wenn die einzelnen Beschreibungen mit Hilfe von unterschiedlichen Schemata erstellt und/oder unterschiedlich kodiert wurden.

2.4.1 PICS

Der unter der Schirmherrschaft des World Wide Web Consortium (W3C) entwickelte PICS-Standard (Platform for Internet Content Selection) legt den Grundstein für eine digitale Beschreibung von elektronischen Ressourcen im Internet. Ausgangspunkt der Bemühungen war die Suche nach geeigneten Kontrollmechanismen, um Kinder und Jugendliche vor dem Zugriff auf ungeeignete und sittenwidrige Web-Inhalte zu schützen. Durch die Verwendung entsprechender PICS-Labels lässt sich aber nicht nur ungeeignetes Datenmaterial sperren, sondern auch gezielt nach indexierten Ressourcen suchen [107].

Zum Verständnis des PICS-Standards ist ein grundlegendes Wissen über die in [84] definierten Begriffe wie *Rating Service*, *Rating System* und *Content Label* notwendig. Unter einem *Rating Service* versteht man eine Person, Gruppe oder Organisation, die entsprechende Metadaten (*Content Labels*) für Internet-Ressourcen zur Verfügung stellt. Jeder *Rating Service* ist durch seine URL eindeutig spezifiziert. Diese URL muss bei der Klassifizierung von Web-Inhalten in allen *Content Labels* angegeben werden und erlaubt somit eine eindeutige Identifizierung der Quelle. Das *Rating System* spezifiziert die einzelnen Merkmale, ihre zulässigen Wertebereiche und beschreibt die Kriterien für die Zuweisung der möglichen Werte. Wie auch der *Rating Service* wird das *Rating System* eindeutig durch eine gültige URL beschrieben. Diese klare Zuweisung ermöglicht es, dass mehrere *Rating Services* auf der Basis des gleichen *Rating Systems* arbeiten.

Die eigentlichen Informationen über eine Ressource sind im *Content Label* enthalten. Das *Content Label* oder auch *Rating* besteht aus drei Teilen:

- 1 der URL des *Rating Services*, der die Klassifizierung vorgenommen hat,
- 1 einer Menge von PICS-konformen Attributwerten, die generelle Informationen über die Klassifizierung liefern, und
- 1 den gewichteten Merkmalen des entsprechenden *Rating Systems*.

Die folgende Abbildung 4 zeigt eine PICS-kompatible Beschreibung eines *Rating Services* auf Basis des sogenannten MPAA *Movie Rating Schemes*.

```
((PICS-version 1.1)
(rating-system "http://MPAAAscale.org/Ratings/Description/")
(rating-service "http://MPAAAscale.org/v1.0")
(icon "icons/MPAAAscale.gif")
(name "The MPAA's Movie-rating Service")
(description "A rating service based on the MPAA's movie-rating scale")

(category
(transmit-as "r")
(name "Rating")
(label (name "G") (value 0) (icon "icons/G.gif"))
(label (name "PG") (value 1) (icon "icons/PG.gif"))
(label (name "PG-13") (value 2) (icon "icons/PG-13.gif"))
(label (name "R") (value 3) (icon "icons/R.gif" ))
(label (name "NC-17") (value 4) (icon "icons/NC-17.gif")))))
```

Abbildung 4: Beschreibung eines Rating Services für PICS

In Abbildung 5 ist ein auf der vorangegangenen Beschreibung basierendes PICS-Label beschrieben. Durch die URL in der ersten Zeile wird der entsprechende Label Service identifiziert. Die folgenden Zeilen enthalten Informationen über das Erstellungsdatum, den Gültigkeitszeitraum, die klassifizierte Ressource sowie den Autor des Metadatensatzes. Die eigentliche Bewertung des Dokumentes findet in der letzten Zeile statt. Den einzelnen Merkmalen *Language* (l), *Sex* (s) und *Violence* (v) sind die Werte 3, 2 und 0 zugewiesen.

```
(PICS-1.1 "http://old.rsac.org/v1.0/" labels
on "1994.11.05T08:15-0500"
until "1995.12.31T23:59-0000"
for "http://www.gcf.org/stuff.html"
by "John Doe"
ratings (l 3 s 2 v 0))
```

Abbildung 5: Beispiel für ein PICS-Label

Zur Übermittlung der PICS-Metadaten stehen mehrere Möglichkeiten zur Verfügung. [85]. Bei der Klassifizierung von HTML-Dokumenten kann die Ressourcen-Beschreibung durch das Metatag direkt in den *Header* des Dokumentes eingebunden werden und ist somit jederzeit verfügbar. Eine weitere Methode ist das direkte Anfordern des gewünschten *Labels* beim entsprechenden *HTTP-Server* über einen *GET-Request*. Die gespeicherten Metadaten werden dann zusammen mit dem eigentlichen Dokument an den *Client* übertragen. Eine dritte Variante ist die Verteilung durch ein *Label Bureau*, das die Metadaten eines oder mehrerer *Rating Services* zur Verfügung stellt. Ein *Label Bureau* stellt einen *HTTP-Server* dar, der spezielle *URL-Query-Strings* verarbeitet. Die Trennung von Metadaten und Inhalt erlaubt die Beschrei-

bung von Ressourcen durch Dritte, auch wenn die eigentlichen Autoren eine Klassifizierung ihres eigenen Datenmaterials ablehnen. Dies ermöglicht unter anderem die Kennzeichnung von jugendgefährdendem oder propagandistischem Datenmaterial durch unabhängige Instanzen.

2.4.2 Dublin Core

Das Dublin Core Metadatenchema ist ein einfaches und weit verbreitetes Schema zur Beschreibung web-basierter Ressourcen. Es wurde im Jahre 1997 von einer Gruppe von Informationswissenschaftlern, Bibliothekaren und Dokumentaren in Dublin, Ohio (USA), erarbeitet. [29]. Dublin Core ist vollständig textorientiert und daher sowohl von Menschen, als auch von automatisierten Systemen interpretierbar. Dublin Core ist ein besonders im Bibliothekswesen verbreitetes Schema. Folgende Eigenschaften charakterisieren Dublin Core und zeichnen ihn als zukünftigen Standard für die Beschreibung elektronischer Ressourcen aus:

- 1 Einfachheit
Dublin Core ist sowohl für Spezialisten als auch Anfänger geeignet, da die meisten Elemente eine allgemein verständliche und überschaubare Syntax haben.
- 1 Semantische Interoperabilität
Im Internet existieren verschiedene Beschreibungsmodelle mit der Fähigkeit, über interdisziplinäre Grenzen hinaus zu suchen. Die Förderung eines allgemein verständlichen Satzes von Deskriptoren vereinfacht die Vereinigung mit anderen Datenstandards und erhöht die Möglichkeit semantischer Interoperabilität zwischen den Disziplinen.
- 1 Erweiterbarkeit
Dublin Core stellt eine wirtschaftliche Alternative im Vergleich zu detaillierteren Beschreibungsmodellen wie den MARC-Katalog der Bibliotheken dar. Beim MARC-Katalog handelt es sich um ein Standardformat zur Darstellung bibliographischer Daten wie Text, Computer-Dateien, Karten, Audio und Video in maschinen-lesbarer Form. Bibliographische Daten enthalten dabei Metadaten wie Titel, Namen, Themen und Inhalte sowie Veröffentlichungsdaten. Für weitergehende Informationen siehe [43].

Die 15 Elemente des Dublin Core

Dublin Core sieht 15 Elemente vor, die als allgemein verständliche Bezeichner von Meta-Tags verwendet werden können [44], [146]:

1. *DC.title* (Titel) gibt den Namen der WWW-Seite an, wobei DC.title nicht mit der Angabe im Title-Tag der HTML-Seite übereinstimmen muss.
2. *DC.creator* (Urheber) gibt den Autor der Seite an, also die Person, die für den Inhalt verantwortlich ist.
3. *DC.subject* (Schlüsselwörter) nennt alle Themen der Publikation. Hierbei sollten Schlagwörter verwendet werden, die den Inhalt am besten wiedergeben.
4. *DC.description* (Beschreibung) dient zur Beschreibung bzw. als Zusammenfassung des Inhalts.
5. *DC.publisher* (Verlag) bezeichnet die Person oder Organisation, die für die Publikation in der vorliegenden Form verantwortlich ist (Web-Administrator, Abteilung einer Firma

oder ein Verlag).

6. *DC.contributor* (Mitwirkende) zählt die Personen oder Organisationen auf, die einen nennenswerten Beitrag zu der Veröffentlichung geleistet haben, jedoch unter *DC.creator* bzw. *DC.publisher* nicht erwähnt wurden.
7. *DC.date* (Datum) gibt das Datum der Publikation in der vorliegenden Form an. Dabei empfiehlt sich die normierte Darstellung JJJJ-MM-TT, also zunächst die vierstellige Jahreszahl, gefolgt von den zweistelligen Zahlen für Monat und Tag.
8. *DC.type* (Art der Quelle) beinhaltet Terme, die die generellen Kategorien, Funktionen und Genres beschreiben.
9. *DC.format* (Format) enthält normalerweise den Medientyp. Format kann benutzt werden, um die Software, Hardware oder andere Ausrüstung zu beschreiben, die notwendig ist, um die Ressource darzustellen oder zu verarbeiten.
10. *DC.identifier* (Bezeichnung) kennzeichnet das Dokument eindeutig beispielsweise mittels einer ISBN, wenn es sich um die elektronische Variante eines Buches handelt oder die ISSN einer digitalen Zeitschrift.
11. *DC.source* (Quelle) dient zur Angabe der Quelle des Werkes, also z.B. Titel, Verlag und ISBN der entsprechenden gedruckten Version.
12. *DC.language* (Sprache) bezeichnet die verwendete natürliche Sprache.
13. *DC.relation* (Verhältnis) kennzeichnet das Verhältnis des Dokumentes zu anderen Ressourcen (z.B. Bilder in einem Dokument, Buch etc.).
14. *DC.coverage* (erfaßte Menge) wird typischerweise den Aufenthaltsort, die Zeitperiode oder die Jurisdiktion beinhalten.
15. *DC.rights* (Rechte) soll Informationen über Rechte in und über die Ressource bereitstellen oder auf einen Dienst verweisen, der solche Informationen zur Verfügung stellt. Informationen über Rechte beinhalten oft Urheberrechte und das Copyright. Fehlt das *Right* Element, können keine Aussagen über den Status von Rechten in bezug auf das Dokument gemacht werden.

Bezogen auf Abbildung 3 lassen sich die 15 Elemente folgendermaßen gruppieren:

- 1 Inhaltsabhängige direkt inhaltsbasierende Elemente: Title, Subject, Description, Source, Language
- 1 Inhaltsabhängige inhaltsbeschreibende Elemente: Relation, Coverage;
- 1 Inhaltsunabhängige administrative Elemente: Creator, Publisher, Contributor, Rights
- 1 Inhaltsabhängige identifizierende Elemente: Date, Type, Format, Identifier.

Da die 15 Elemente bei vielen Anwendungen nicht zur Beschreibung ausreichen, bietet Dublin Core die Möglichkeit das Schema zu erweitern, um die Struktur und ausführlichere Semantik, zu kodieren. Das Basis-Schema wurde weiterhin um eine Variante erweitert, die "fully qualified Dublin Core" genannt wird. In diesem Schema werden alle 15 Elemente weiter verfeinert und zusätzlich wird für bestimmte Elemente ein festes Vokabular als möglicher Wert vorgegeben. Beispiel:

Einer der Hauptkritikpunkte an Dublin Core ist, daß dem Schema der bibliographische Hintergrund zu deutlich anzusehen ist und es kaum möglich ist, alle multimedialen Eigenschaften einer Ressource zu beschreiben [128]. In [53] wird hingegen eine auf Dublin Core basierende Struktur vorgestellt, die mit Hilfe der Vokabulare eine detaillierte Beschreibung eines Videos ermöglicht. Diese Struktur ist in Abbildung 6 gezeigt.

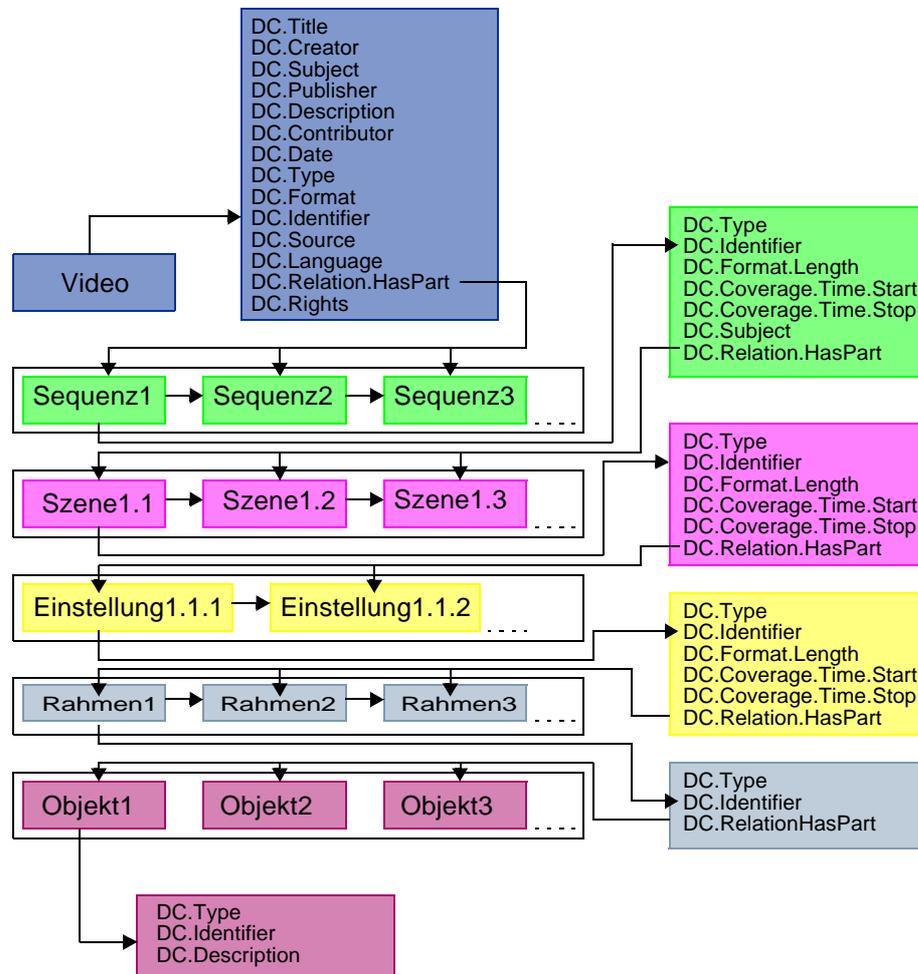


Abbildung 6: Beispiel für eine Dublin Core Beschreibung (vgl.[53])

Die Abbildung zeigt, daß das Video in Sequenzen, Szenen, Einstellungen und Rahmen unterteilt ist, wobei bsp. jede Sequenz einem Video eindeutig zugeordnet ist und aus einer eindeutig bestimmten Menge von Einstellungen besteht. Jeder Komponente des Videos ist eine Dublin Core Beschreibung zugeordnet, die aus unterschiedlichen Elementen besteht. Aus der Abbildung ist auch ersichtlich, daß die hierarchische Struktur der Komponenten des Videos nicht direkt in der Dublin Core Struktur abgebildet werden kann. Der Grund hierfür ist die Unabhängigkeit Dublins von der Kodierung der Metadatenbeschreibungen. Da nicht davon ausgegangen werden kann, daß die Hierarchie der Komponenten des Videos durch eine hierarchische Kodierung der Dublin Core Beschreibungen ausgedrückt werden kann, muß ein Vokabular eines Dublin Core Metadatenelements, in diesem Fall der Wert *hasPart* des *Relation*-Elements

verwendet werden um die hierarchische Struktur in den Dublin Core Beschreibungen abzubilden. Eine Anwendung, die die Metadatenbeschreibungen analysiert und für eine Suche verwendet muß daher die semantische Bedeutung des Vokabulars kennen. Ist dies nicht der Fall, kann bei einer Suche der Sachverhalt, daß mehrere Einstellungen zu der gleichen Szene gehören, oder daß ein Objekt in mehreren Einstellungen auftaucht, nicht verwendet werden. Die Definition eines Metadatenschemas, daß keinerlei Aussagen über das Kodierungsformat macht, stellt daher höhere Anforderungen an das verwendete Vokabular, als es bei einem Schema der Fall ist, daß einen Teil der Beziehungen der zu beschreibenden Ressourcen implizit durch das Kodierungsformat ausdrückt. Ein Beispiel für ein solches Schema ist das im nächsten Abschnitt besprochene MPEG-7

2.4.3 MPEG-7

Auf dem Gebiet der Metadaten für allgemeine multimediale Ressourcen existieren bisher nur Teillösungen [95][23]. So ist z.B. in heutigen speziellen Multimedia-Datenbanken die Suche nach charakteristischen Farben, Texturen oder Oberflächenbeschaffenheiten von Objekten möglich. Die Frage nach dem Auffinden von multimedialen Inhalten ist nicht nur auf Datenbank-Retrieval-Anwendungen beschränkt, sondern ist z.B. auch interessant, wenn es um eine Auswahl an potentiell interessanten digitalen Fernseh- oder Radiosendern geht, die heute in einem immer größer werdenden Angebot konkurrieren. Dies war der Hauptgrund für die Motion Picture Expert Group (MPEG), mit der Arbeit an einem umfassenden Beschreibungssystem zu beginnen, das unter dem Namen MPEG-7 bekannt ist [80]. MPEG-7 ist eine standardisierte Beschreibung von verschiedenen Typen multimedialer Information. Diese Beschreibung wird mit dem Inhalt selbst assoziiert sein, um schnelles und effizientes Suchen für den Nutzer zu gewährleisten. MPEG-7 wurde aus dem früheren Multimedia Content Description Interface heraus entwickelt. Demzufolge enthält MPEG-7 einen Standardsatz an Deskriptoren, die benutzt werden, um die verschiedenen Typen multimedialer Information zu beschreiben [11]. MPEG-7 bietet ebenso die Möglichkeit, andere Deskriptoren zu spezifizieren, sowohl auch ganze Beschreibungs-Schemata (DS) für die Deskriptoren und ihre Beziehungen ermöglichen. Abbildung 7 zeigt die allgemeine Struktur eines MPEG-7 Systems.



Abbildung 7: Komponenten des MPEG7-Systems (vgl. [71])

Entsprechend der Abbildung besteht ein MPEG7-System also aus drei Komponenten:

- 1 Die Merkmalsextraktion
Die Analyse der Ressource und die automatische Generierung der darin enthaltenen

Informationen ist nicht normativ und kann für ein kompatibles System frei gewählt werden.

1 Inhaltsbeschreibung

Die Beschreibung des extrahierten Inhalts ist die eigentliche Aufgabe der MPEG-7-Standardisierung. MPEG-7-Beschreibungen werden durch die unten erläuterten MPEG7-Deskriptoren erstellt.

1 Anwendung

Als Anwendung ist in Abbildung 7 eine Suchmaschine angegeben. Sie stellt lediglich einen der in 2.1.2 beschriebenen Anwendungsfälle für die Verwendung von Metadaten dar und ist ebenfalls nicht Teil des Standards.

MPEG7-Deskriptoren

1 Descriptors (D)

Ein Deskriptor (D) ist eine Repräsentation eines Merkmals. Ein Deskriptor definiert die Syntax und die Semantik der Merkmalsrepräsentation. Er entspricht damit einem Attribut eines Metadatenschema

1 Descriptor Values (DV)

Ein Deskriptorwert ist eine Instanziierung eines Deskriptors mit einer gegebenen Datenmenge und repräsentiert damit die möglichen Werten eines Attributs

1 Descriptor Scheme (DS)

Ein Deskriptor-Schema (DS) spezifiziert die Struktur und Semantik zwischen den Komponenten und stellt damit ein vollständiges Metadatenschema dar. Als Komponenten eines Schema kommen dabei sowohl Deskriptoren als auch weitere Schemata in Frage.

1 Description Definition Language (DDL)

Die "Description Definition Language" ist eine Sprache zur Spezifikation von Deskriptor Schemata. Sie erlaubt zusätzlich die Erweiterung und Modifikation von existierenden "Description Scheme". Mit der Spezifikation der DDL gibt MPEG-7, anders als andere Entwürfe, die Unabhängigkeit des Schemas von der Kodierung auf, da es die DDL beispielsweise ermöglicht, den Attributen Kardinalitäten zuzuweisen, Vererbungsbeziehungen zwischen Deskriptoren festzulegen, oder hierarchische Strukturen unabhängig vom verwendeten Vokabular zu definieren. Die verwendete Kodierung einer MPEG-7 Beschreibung muß daher eine Kontrolle ermöglichen, wie oft ein bestimmtes Attribut innerhalb einer Beschreibung vorkommen darf, oder Attribute einer Superklasse automatisch auch der Basisklasse zuweisen.

1 Description (Beschreibung)

Eine Beschreibung setzt sich zusammen aus einem DS und einer Menge von Deskriptoren oder weiteren Schemata. Abbildung 8 zeigt die Metadatenstruktur aus Abbildung 6 im Kontext von MPEG-7. Wie die Abbildung zeigt, kann die Dublin Core-Struktur direkt in eine MPEG-7 Struktur integriert werden, da MPEG-7 die Verwendung mehrerer Deskriptor Schemata zur Beschreibung einer Ressource erlaubt und ein Deskriptor Schema für Dublin Core definiert werden kann. Wie aus der Abbildung ersichtlich ist, wird bei dieser Struktur jede Dublin Core Beschreibung um spezielle MPEG-7 Deskriptoren erweitert, die in der Abbildung durch den weißen Rahmen gekennzeichnet sind. Zusätzlich zur Beschreibung der einzelnen Sequenzen des Videos, kann eine MPEG-7 Beschreibung weitere Deskriptor Schema enthalten, mit denen beispielsweise der Soundtrack des Videos beschrieben ist.

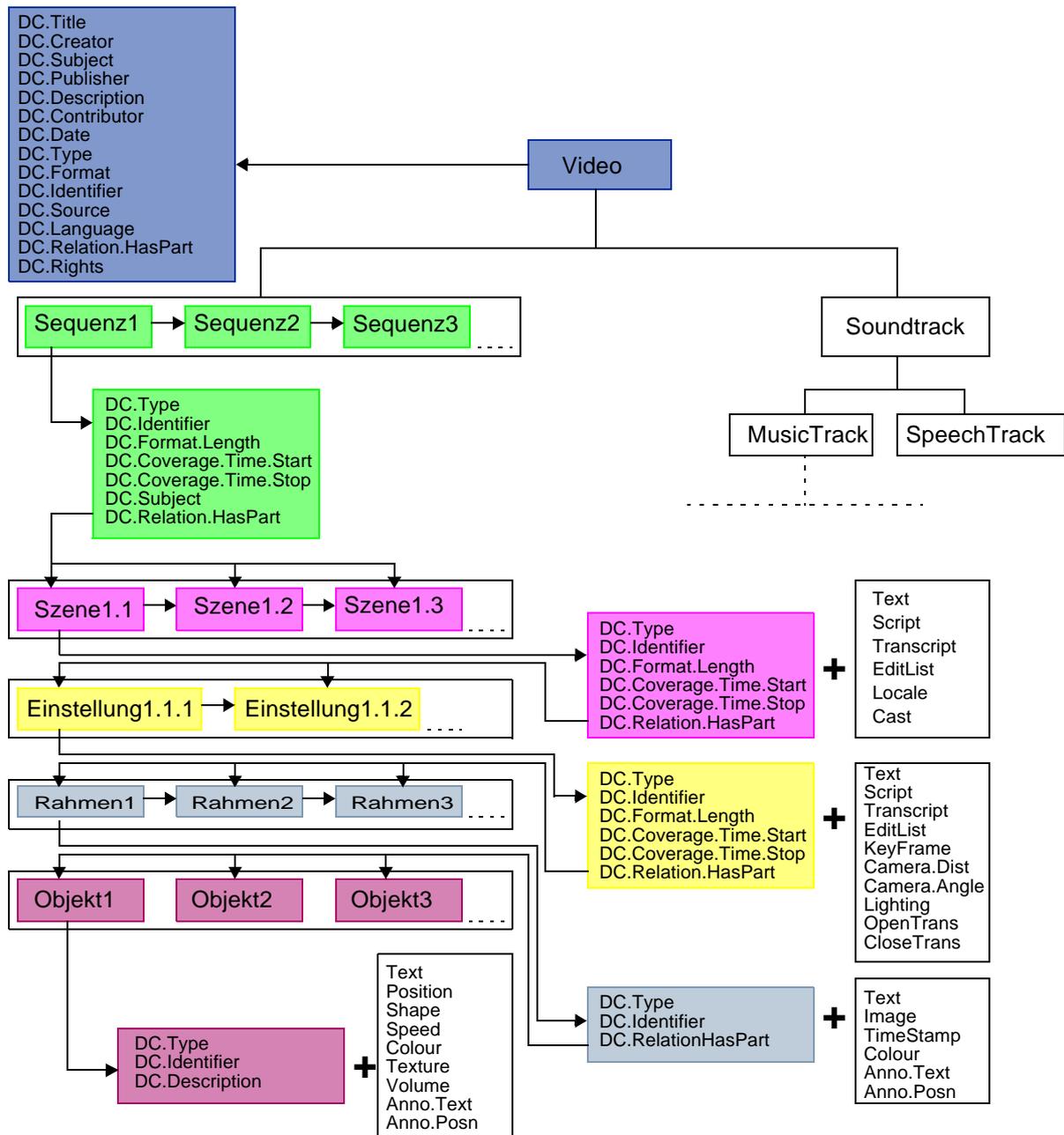


Abbildung 8: Beispiel einer MPEG7-Beschreibung (vgl.[53])

MPEG-7 ist, wie auch andere Mitglieder der MPEG-Familie, eine standardisierte Repräsentation von audiovisuellen Informationen die spezielle Anforderungen erfüllen müssen. Der MPEG-7 Standard baut auf anderen Repräsentationen wie z.B. analog PCM, MPEG-1, MPEG-2, und MPEG-4 auf. Eine Funktion des Standards ist es, Referenzen zu passenden Anteilen der anderen Standards bereitzustellen. So kann z.B. ein Oberflächen-Deskriptor aus MPEG-4 auch in einem MPEG-7 Kontext hilfreich sein. Dasselbe gilt z.B. auch für Bewegungsvektoren, welche in MPEG-1 und MPEG-2 vorkommen. MPEG-7-Deskriptoren sind unabhängig von der Art und Weise der Speicherung bzw. Kodierung des Inhalts. So ist es z.B. möglich, einem analogen Film, sowie einem Bild, das später ausgedruckt werden soll, mit einer MPEG-7

Beschreibung zu verbinden. Ebenso ist MPEG-7 von der kodierten Repräsentation des Materials unabhängig. Diese Fähigkeit baut auf dem MPEG-4 Standard auf, der über Methoden verfügt, audiovisuelle Daten als Objekte, die bestimmte zeitliche (Synchronisation) und räumliche (auf dem Bildschirm für Video, bzw. im Raum für Audio) Beziehungen haben, zu enkodieren. Beim Gebrauch der MPEG-4 Kodierung ist es möglich, Beschreibungen Elementen (Audio- oder visuellen Objekten) innerhalb einer Szene zuzuordnen. MPEG-7 hingegen ermöglicht verschiedene Abstufungen in seiner Beschreibung, was eine Möglichkeit verschiedener Unterscheidungsstufen bietet. Da die deskriptiven Merkmale innerhalb des Kontexts aussagekräftig sein müssen, unterscheiden sie sich für unterschiedliche Nutzerbereiche wie auch für unterschiedliche Anwendungen. Das bedeutet, dass dasselbe Material unter Zuhilfenahme unterschiedlicher Merkmalstypen beschrieben werden kann, je nach Einsatzgebiet der Anwendung. Einfache Abstraktionsmaße für visuelle Materialien sind z.B. Oberflächenbeschaffenheit, Größe, Textur, Bewegung und die Position eines Objektes. Für Audio-Materialien sind dies dann z.B. Tonhöhe, Tempo, Tempoveränderung und die Position. Komplexe Abstraktionsmaße geben dann semantische Informationen wie z.B. "dies ist eine Szene mit einem braunen, bellenden Hund im Vordergrund links und einem blauen, herabfallenden Ball auf der rechten Seite, mit dem Hintergrund vorbeifahrender Autos".

2.4.4 Warwick Framework

Wie die Beispiele PICS, MPEG7 und Dublin Core und die Erläuterungen aus Abschnitt 2.1.2 "Zweck von Metadaten" zeigen, kann die gleiche Ressource durch unterschiedliche Metadaten beschrieben werden. Zudem kann jedes Metadatenschema noch auf verschiedene Weise kodiert werden (siehe Abschnitt 2.6 "Kodierung von Metadaten"). Diese Vielfalt kann dazu führen, daß kein einheitlicher Zugang zu den Metadatenbeschreibungen möglich ist. Damit besteht aber die Gefahr, daß der grundlegende Ansatz, durch Metadaten eine von der Kodierung der Ressourcen unabhängige Suche nach Medienbausteinen zu ermöglichen, verloren geht. Ein Ansatz, diesem Problem zu begegnen, stellt das Warwick Framework dar. Das Warwick Framework stellt eine Architektur zur Verfügung, um die Beschreibung einer Ressource durch verschiedene Metadatensätze zu ermöglichen. Es ermöglicht gleichzeitig die Definition von individuell angepaßten Metadatensätzen, ohne hierbei auf eine universelle Verwendbarkeit Rücksicht nehmen zu müssen. [51]

Das während des „*Metadata Workshop II*“ in Warwick entstandene Konzept basiert auf einer Container-Architektur, die Kombinationen unterschiedlicher Metadaten-Konzepte ermöglicht. Das *Warwick Framework* besteht aus zwei grundlegenden Komponenten, einem Container und den darin enthaltenen Paketen. Der Container an sich besitzt bis auf seine Aggregationsaufgabe keine weitere Funktionalität. Es besteht keine Möglichkeit die Pakete als mehr oder weniger signifikant zu kennzeichnen. Die innerhalb eines Containers enthaltenen Pakete müssen einem der drei folgenden Typen zugeordnet sein.

- 1 *Metadata Set* (beinhaltet die eigentlichen Metadaten)
- 1 *Indirect* (Referenz auf ein anderes Objekt)
- 1 *Container* (Paket ist seinerseits ein Container)

Der Mechanismus, der einen *Warwick Framework* Container mit einer bestimmten Ressource assoziiert ist implementierungsabhängig. Prinzipiell unterscheidet man aber zwischen intern

und extern referenzierten Metadaten-Containern. Ein intern referenzierter Metadaten-Container beinhaltet die Informationen, die vom Ersteller einer Ressource als relevant angesehen werden. Die Metadaten sind entweder direkt in die Struktur der entsprechenden Originaldaten eingebunden oder werden über einen **Uniform Resource Identifier (URI)** mit diesen verknüpft. Extern referenzierte Metadaten-Container beinhalten Metadatensätze, die nicht direkt zum Erstellungszeitpunkt der Ressource generiert, sondern erst später durch Außenstehende hinzugefügt werden. *Das Warwick Framework* beinhaltet keine Einschränkungen bezüglich der Kodierung der Metadaten und ist damit unabhängig von der physikalischen Speicherung der Metadaten und der Verbindung zwischen Metadaten und Ressource. Zum jetzigen Zeitpunkt existieren bereits Implementierungen in MIME und SGML. Verschiedene Ansätze zur Einbettung von Containern in Web-Seiten durch HTML-Metatags werden momentan noch diskutiert.

2.5 Generierung von Metadaten

Eines der Hauptargumente gegen die Verwendung von Metadaten zur detaillierten Beschreibung von Ressourcen ist der Aufwand zur Erstellung der Beschreibungen. Zusätzlich zum zeitlichen Aufwand, der für die Beschreibung einer Ressource anfällt, stellt sich die Frage, wer die Beschreibungen erstellen soll. Am naheliegendsten scheint der Autor der Ressource zu sein. Die Einordnung einer Ressource in bibliographische Kataloge wird in der Regel aber schon nicht mehr vom Autor selbst, sondern von, in der Erfassung von Metadaten ausgebildeten Personen, durchgeführt werden. Die Erfassung von Metadaten stellt einen intellektuellen Prozeß dar, der sich nur teilweise automatisieren lässt. Eine Extraktion von Meta-Informationen ist im Allgemeinen nicht aus dem Datenmaterial alleine, sondern nur unter Einbeziehung des entsprechenden Umfeldes möglich. Die durch eine Ressource zur Verfügung gestellten Informationen dienen also lediglich als Grundlage der Metadaten, stellen aber keinesfalls die alleinige Quelle dar [101].

Bei der Ressourcenbeschreibung unterscheidet man zwischen inhalts-unabhängigen und inhaltsabhängigen Metadaten, bzw. den auf Feature- und Concept-Level extrahierten Informationen.

2.5.1 Methoden zur Generierung von Metadaten

Zur Generierung von Metadaten stehen verschiedene Methoden zur Verfügung, die in der Regel vom Zeitpunkt der Erstellung und der Art des Datenmaterials abhängen. Nach [7] lassen sich folgende Methoden unterscheiden:

- 1 Manuelle Eingabe
- 1 Vergleich mit Referenzwerten
- 1 Messung
- 1 Berechnung
- 1 Ableitung aus dem Inhalt

Die Generierung von Metadaten durch manuelle Eingabe stellt in der Regel die ineffizienteste Methode dar, die zudem relativ inkonsistente Metadatensätze liefert. Für einige Metadatenenele-

mente, insbesondere solche, die auf semantischen Informationen beruhen, ist dies allerdings die einzige Möglichkeit der Generierung.

Das Erfassen von Metadaten durch einen Vergleich mit Referenzwerten setzt voraus, dass das System bereits über Informationen verfügt, die eine Übernahme eines Elementwertes aus korrespondierenden Datenelementen erlauben. Als Grundlage hierfür können beispielsweise für Bilder unter anderem Histogramme oder Farbverteilungen herangezogen werden, denen entsprechend ihrer Ähnlichkeit zu Referenzen Attributwerte zugewiesen werden.

Das direkte Messen von Metadatenwerten stellt eine einfache und gegen Fehler weitgehend resistente Methode dar, deren Anwendung allerdings nicht bei allen Datenelementen möglich ist. Ähnlich exakte Werte ergeben sich auch bei einer Berechnung der Metadaten aus vorher extrahierten Werten oder bereits existierenden Metadaten.

Die Ableitung von Meta-Informationen alleine aus den Mediendaten stellt in vielen Fällen die einzige Möglichkeit der nachträglichen Generierung von Metadaten dar. Der Informationsgehalt, der aus einer Ressource gewonnen werden kann, hängt stark vom bereits vorhandenen Wissen und eventuellen Informationen aus dem Umfeld der Ressource ab. Eine computergestützte Extraktion von inhaltlichen Informationen stellt hohe Anforderungen an das System und ist in der Regel nur mit bereits vorverarbeitetem Datenmaterial möglich. Die Erfassung der einzelnen Metainformationen muss zu keinem gemeinsamen Zeitpunkt stattfinden, was in den meisten Fällen auch gar nicht möglich ist. Ein Teil der benötigten Informationen wird bereits während des Erstellungsprozesses der Ressource erzeugt, hierzu gehören unter anderem das Dateiformat oder Erstellungsdatum. Diese Daten müssen bei der Generierung des Metadatensatzes nicht erneut erfaßt werden, sondern können aus der Ressource extrahiert werden. Zu diesem Zeitpunkt werden allerdings noch nicht alle relevanten Daten existieren. Einige Meta-Informationen entstehen erst während der weiteren Verarbeitung des Datenmaterials und können folglich auch erst dann zum bestehenden Metadatensatz ergänzt werden. Hierzu zählen vor allem die Relationen zu anderen Ressourcen.

Prinzipiell ist es durchaus möglich, bestimmte Metadaten vor der Erstellung der eigentlichen Ressource zu generieren. Das unter dem Begriff "pre-data collection" (vgl. [Bea95]) bekannte Verfahren findet Anwendung, falls durch die Metadaten Informationen zum Erstellungsprozess geliefert werden können. Generell sollte die Spanne zwischen dem Zeitpunkt, zu dem alle relevanten Informationen vorliegen, und der Generierung der Metadaten möglichst klein gehalten werden.

Die Vorteile, die sich durch die Nutzung von Metadaten bei der Suche und Wiederverwendung von Lernressourcen ergeben, hängen stark von der Qualität und dem Umfang der Metainformationen ab. Ein vollständig bestimmter Metadatensatz liefert im Allgemeinen eine bessere Grundlage für die konsistente und eindeutige Beschreibung einer Lernressource. Geht man vom Einsatz von beispielsweise Dublin Core aus, so ist eine manuelle Beschreibung der Ressource wenig effizient. Die Vollständigkeit eines Metadatensatzes alleine ist aber kein ausreichendes Merkmal für dessen Qualität, d.h. die bloße Existenz eines Elementes liefert noch lange keine zusätzliche Funktionalität. Die manuelle Beschreibung einer Ressource birgt unweigerlich die Gefahr einer mehrdeutigen und inkonsistenten Objektbeschreibung. Die automatische Generierung der Feldeinträge ist ein wichtiger Schritt in Richtung einer objektiven Charakterisierung der Lernressource. Allerdings ist zu beachten, dass das Ausschalten jeg-

licher Subjektivität bei der Beschreibung einer Ressource, auch aufgrund des vorgegebenen Datenschemas, nicht möglich ist. Die homogene Beschreibung von Ressourcen kann jedoch durch die Definition von Vorgabe- und Referenzwerten in den meisten Fällen gewährleistet werden.

2.5.2 Extraktionsebenen

Die Extraktion von Eigenschaften einer Ressource ist auf zwei unterschiedlichen Ebenen möglich: auf dem Feature- und dem Concept-Level. Auf der ersten Ebene können alle Dateimerkmale zusammengefaßt werden, zu deren Generierung kein zusätzliches Wissen über den Dateiinhalt notwendig ist. Die Informationen lassen sich direkt aus dem uninterpretierten Datenmaterial ableiten. Ein Teil, der zur Generierung der Metadaten notwendigen Eigenschaftswerte ist unmittelbar gegeben (Dateigröße, Erstellungsdatum etc.), andere müssen erst aus dem Rohdatenmaterial extrahiert werden (Histogramm, Farbtiefe etc.). Aus den so ermittelten Dateimerkmalen sind auch ohne inhaltliches Wissen weitere Charakteristika abzuleiten. Durch den Datentyp der einzelnen Medien sind mit entsprechendem Hintergrundwissen Angaben über die zur Wiedergabe benötigte Hard- und Software möglich. Mit den auf diese Weise extrahierten Dateimerkmalen ist allerdings keine vollständige Beschreibung einer Ressource möglich.

Eine weitergehende Klassifizierung der Ressource ist nur auf Concept-Level mit zusätzlichen, inhaltlichen Informationen möglich. Das Ableiten von Dateimerkmalen auf Concept-Level stellt wesentlich größere Probleme dar als die Merkmalsextrahierung auf Feature-Level. Die Schwierigkeit liegt vor allem in der Informationspräsentation. Während die Informationen auf dem Feature-Level meist explizit gegeben sind, liegen sie auf Concept-Level in der Regel nur in impliziter Form vor. Als Ausgangspunkt für eine computergestützte Generierung von Metadaten müssen die zu verarbeitenden Objektmerkmale aber in einer eindeutigen und vom Programm verständlichen Form vorliegen. Diese Voraussetzung kann aber bei Multimediadaten nur in den wenigsten Fällen als existent angenommen werden, weshalb eine direkte Informationsverarbeitung durch das System selten realisierbar ist. In vielen Fällen sind die aus einer Datei abzuleitenden Informationen auch von dem bereits vorhandenen Wissen abhängig. Die Erkenntnisse, die ein Laie zum Beispiel aus einem Satellitenfoto gewinnen kann, fallen in der Regel recht dürftig aus. Bei entsprechenden Vorkenntnissen aus dem Bereich der Meteorologie lassen sich aus dem gleichen Bild eine Vielzahl von Informationen ableiten. Durch dieses Beispiel sollte klar geworden sein, dass die Erfassung aller, in einer multimedialen Datei enthaltenen Informationen, nur in den seltensten Fällen möglich ist. Ziel der Informationsgewinnung muss es deshalb sein, die für eine bestimmte Anwendung wesentlichen Informationen zu erfassen und irrelevante oder redundante Daten zu verwerfen. Nur so ist eine effiziente Beschreibung einer Ressource mit einem überschaubaren Metadatensatz möglich.

2.5.3 Generierungsmechanismen

Die computergestützte Generierung von Metadaten setzt nicht zwangsläufig eine automatische Extraktion der Merkmale aus dem zu beschreibenden Datenmaterial voraus. In bestimmten Fällen ist dies auch gar nicht möglich. Die Tatsache, dass eine Beschreibung bestimmter Metadaten-Felder alleine mit dem automatisch aus Datei- oder Programminformationen gewonnenen Wissen nicht möglich ist, sagt noch nicht aus, dass eine programmgestützte Realisierung

prinzipiell nicht möglich ist. In fast jedem Metadaten-Standard befinden sich Datenelemente, die sich nicht explizit auf eine Beschreibung des Inhaltes oder auf die technische Charakterisierung der Datei beziehen. Hierzu zählen vor allem Angaben über den Autor und die rechtlichen Grundlagen für die Nutzung der Ressource durch Dritte. In der Regel kann man davon ausgehen, dass sich diese Informationen beim Erstellen weiterer Metadatensätze desselben Autors nicht mehr ändern werden und somit automatisch für jede weitere Ressource verwendet werden können.

Nutzung existierender Metadaten

Die einfachste Art, einen neuen Metadatensatz zu erzeugen, besteht in der Nutzung von bereits vorhandenen Metadaten, die auf entsprechenden Feldern des neuen Datensatzes abgebildet werden. Die Metadaten einer bereits klassifizierten Ressource können entsprechend eines existierenden Metadatenstandards gespeichert sein, oder die Informationen liegen in einem anwendungsspezifischen, internen Format vor. Aus einem Textdokument können Metainformationen durch Parsen der Datei herausgefiltert werden. Bei der Beschreibung einer Datei kann die Konformität bezüglich eines gängigen Metadaten-Standards aber nicht vorausgesetzt werden. Viele Anwendungsprogramme speichern die zusätzlichen Metainformationen in einer programmspezifischen Strukturierung und einem eigenen Format. Auch wenn durch den Anwender keine zusätzlichen Einträge in die zur Dokumentenbeschreibung vorgesehenen Felder vorgenommen wurde, ist eine Datei durch die anwendungsinternen Informationen meist ausreichend beschrieben. Durch fehlende Export-Funktionen und Austauschformate sind die vorhandenen Informationen aber wesentlich schwieriger zu verarbeiten als beispielsweise die Metadaten eines HTML-Dokumentes. Ob der Aufwand für Auswertung und Verarbeitung dieser Informationen gerechtfertigt ist, hängt stark von der Anwendung und der Menge an Informationen, die herausgefiltert werden können, ab.

Ableitung aus Dateimerkmalen

Eine weitere Möglichkeit, Metainformationen aus bereits bestehendem Datenmaterial zu gewinnen, ist die Ableitung der Informationen aus Dateimerkmalen. Fast alle zu extrahierenden Informationen lassen sich auf der Basis von Dateinamenerweiterungen generieren. Dies setzt allerdings voraus, dass für alle Metadatenelemente entsprechende Lookup-Tabellen existieren, mittels derer eine Abbildung der Dateinamenerweiterung auf den dazugehörigen Feldeintrag möglich ist.

Durch die Kombination von abgeleiteten Merkmalen mit bereits existierenden Metainformationen ist im Allgemeinen eine effiziente Abbildung auf vorhandene Elementeinträge möglich. Eine Erfassung semantischer Informationen ist allerdings nicht möglich. Der Informationsgehalt und die Verwendbarkeit der generierten Metadaten hängt stark von der Qualität und dem Umfang der entsprechenden Abbildungsvorschriften ab. Automatische Generierungsmechanismen, die auf der Grundlage von Lookup-Tabellen arbeiten, liefern in der Regel zwar konsistente Ergebnisse, die aber durch die endliche Zahl an Tabelleneinträgen meist nur allgemeingültigen Charakter besitzen.

Benutzerspezifische Schablonen

Die Datenelemente, die nicht automatisch aus einer Ressource extrahiert werden können, las-

sen sich entsprechend vorgegebener Merkmale gruppieren und können bei der späteren Charakterisierung zum Metadatensatz hinzugefügt werden. Jedem Autor wird hierbei die Möglichkeit gegeben, sich eigenständig Vorlagen zu definieren, die eine bestimmte Menge von Datenelementen beschreiben. In welchem Umfang eine individuelle Erstellung von Schablonen möglich sein sollte, hängt einerseits von der Anwendung, andererseits von den programmtechnischen Kenntnissen des Autors ab. Eine Möglichkeit wäre die Vorgabe von fest definierten Schablonen für bestimmte Bereiche, bei denen der Autor die Möglichkeit hat, den einzelnen Datenelementen Werte zuzuweisen und die entsprechenden Eingaben zur weiteren Verwendung zu speichern. Alle innerhalb eines Bereiches erstellten Ressourcen lassen sich anschließend über diese Schablone mit relativ geringem Zeitaufwand eindeutig beschreiben. Die Schwierigkeit dieses Ansatzes liegt in der effizienten Komposition von Merkmalen. Werden zu wenig Elemente zu einer Schablone zusammengefaßt, ist der anschließende Aufwand bei der eigentlichen Beschreibung verhältnismäßig hoch, da einzelne Kombinationen von Schablonen eventuell wiederholt geladen werden müssen. Die Verwendung größerer Schablonen ermöglicht zwar eine effizientere Beschreibung von Ressourcen, allerdings steigt mit zunehmender Anzahl von gruppierten Elementen auch die Wahrscheinlichkeit für Änderungen innerhalb der Schablone. Bei einer großen Anzahl von zusammengefaßten Elementen muss eventuell eine beträchtliche Menge ähnlicher Schablonen zur Verfügung stehen, um die entsprechenden Felder zu beschreiben. Die Summe annähernd gleicher Schablonen lässt sich durch Aufspaltung in mehrere kleine reduzieren, die dann entsprechend der Anforderungen verknüpft werden können. Erweitert man den Handlungsspielraum der Autoren so, dass die einzelnen Datenelemente in Abhängigkeit der persönlichen Bedürfnisse zu Schablonen zusammengefaßt werden können, ist davon auszugehen, dass die Anzahl ähnlicher Schablonen deutlich reduziert werden kann. Diese Reduzierung ist innerhalb eines spezifischen Projektes mit Sicherheit als großer Vorteil zu werten. Allerdings muss gleichzeitig davon ausgegangen werden, dass durch die gezielte Anpassung der Schablonen an einen konkreten Anwendungsfall deren Wiederverwendbarkeit stark eingeschränkt wird. Schablonen, die beispielsweise zur Klassifizierung von Textdokumenten erstellt wurden und dementsprechend bezüglich der Auswahl von Merkmalen an das Medium Text angepaßt sind, lassen sich nur nach intensiver Überarbeitung auch zur Beschreibung von Videostreamen einsetzen. Die möglichst universelle Verwendbarkeit von Schablonen kann bei der Vorgabe fest definierter Schablonen im Vorfeld berücksichtigt werden, bei der Definition durch den Anwender ist sie allerdings in den wenigsten Fällen gewährleistet.

Globale Schablonen

Benutzerspezifische Schablonen ermöglichen eine individuelle Anpassung an persönliche Bedürfnisse und ein schnelles und unkompliziertes Erstellen neuer Schablonen. Allerdings existieren auch einige Schwachpunkte, die nicht vernachlässigt werden dürfen. Sind innerhalb eines Projektes mehrere Personen mit der Generierung von Metadaten betraut, so hat jeder die Möglichkeit, ohne Einschränkung eigene Schablonen zu definieren. Der Nachteil besteht nicht nur in der offensichtlichen Ineffizienz, da die Schablonen auf jedem System neu definiert werden, sondern vor allem in der fehlenden Gewähr der Analogie. Bei der Beschreibung ein und derselben Ressource durch mehrere Anwender kann nicht garantiert werden, dass die entstehenden Metadatensätze identisch sind. Wie bereits beschrieben, wird die Eliminierung jeglicher Subjektivität bei der Beschreibung von Lernobjekten wahrscheinlich nie möglich sein. Trotz allem ist durch die Nutzung von global definierten Schablonen ein Schritt in Richtung größtmöglicher Identität bei der Charakterisierung von Ressourcen möglich. Globale Schablonen werden von einer Person erstellt und können dann von allen Anwendern bei der Generie-

zung von Metadaten genutzt werden. Damit ist gewährleistet, dass Datenfelder, die mittels einer Schablone beschrieben werden, auf allen Systemen identische Werte besitzen. Die Ressourcenbeschreibungen werden damit aber nicht nur homogener, sondern in der Regel auch detaillierter. Es kann nicht immer davon ausgegangen werden, dass alle relevanten Informationen in vollem Umfang für den Autor zugänglich sind. Ein Beispiel hierfür ist ein Metadaten-element, in dem die Nutzungsbedingungen der Ressource spezifiziert werden. Ein denkbarer Ansatz wäre die Zusammenfassung einzelner Ressourcen aufgrund ihrer rechtlichen Bestimmungen zur Wiederverwendung. Für jede dieser Gruppen sollten dann juristisch einwandfreie Beschreibungen in Form von globalen Schablonen zur Verfügung stehen, die von jedem Anwender zur Klassifizierung genutzt werden können. Eine Beschreibung durch globale Schablonen ist mit Sicherheit nicht für alle Datenelemente praktikabel oder sinnvoll. In Bereichen, in denen die Verwendung von globalen Schablonen realisierbar ist, sollte in Hinblick auf eine homogene und detaillierte Klassifizierung auch von dieser Möglichkeit Gebrauch gemacht werden.

2.6 Kodierung von Metadaten

Die Einigung auf bestimmte Elemente und Werte für ein Metdatenschema ist eine der Voraussetzungen dafür, dass eine Suche nach Ressourcen möglich ist. Sobald Metadatenbeschreibungen aber nicht zentral erfaßt werden, sondern verteilt erstellt und ausgetauscht werden müssen, entscheidet die Kodierung der erfaßten Beschreibungen letztendlich über den Grad der Wiederverwendung der Beschreibungen. Die Kodierung der Metadatenbeschreibung ist direkt abhängig von der Beziehung der Metadaten zu der dazugehörigen Ressource. Dabei lassen sich wie in Abbildung 9 gezeigt drei prinzipielle Varianten der Beziehung unterscheiden:

- 1 Kodierung innerhalb der Ressource (Abbildung 9a)
- 1 Kodierung zusammen mit der Ressource (Abbildung 9b)
- 1 Kodierung unabhängig von der Ressource (Abbildung 9c)

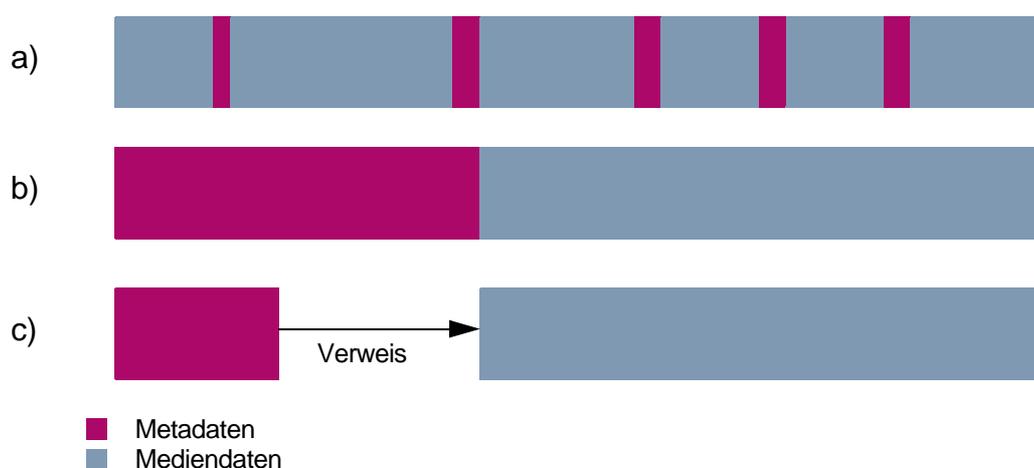


Abbildung 9: Zusammenhang zwischen Mediendaten und Metadaten

2.6.1 Kodierung innerhalb der Ressource

Im Fall der Kodierung der Metadatenbeschreibung zusammen mit den Mediendaten wird die Kodierung der Metadaten von der Kodierung der Ressource bestimmt. Abbildung 10 zeigt einen Auszug aus einer HTML-Seite, indem das in Tabelle 1 aufgeführte Buch beschrieben ist. Die Metadaten werden in diesem Fall durch die Verwendung des seit HTML 3.2 eingeführten META-Tags identifiziert. In diesem Fall sind alle Metadatenelemente durch das Hinzufügen der Zeichenkette "DC." zum Namen des Meta-Tags zusätzlich noch als Dublin Core kompatible Metadaten erkennbar.

```
<!doctype html public "-//w3c//dtd html 4.0 transitional//en">
<html> <head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<meta name="GENERATOR" content="Mozilla/4.7 [en] (X11; I; Linux 2.2.12-20 i686)
[Netscape]">
<META NAME="DC.Title" CONTENT="Multimedia-Technologie; Grundlagen Komponenten
und Systeme">
<META NAME="DC.Description" CONTENT="Visualization of Carrier Sense Multiple
Access Protocol with Collision Detection (CSMA/CD) IEEE 802.3 (Ethernet)">
<META NAME="DC.Type" CONTENT="Learning Material">
META NAME="DC.Format" CONTENT="html">
<META NAME="DC.Relation.References" CONTENT="http://www.multibook.de/">
</head>
<body text="#000000" bgcolor="#FFFFFF" link="#009AA8" vlink="#FF0000"
alink="#000088"><h1>
Multimedia-Technologie - Grundlagen, Komponenten und Systeme</h1>
...
</body>
</html>
```

Abbildung 10: Metadatenbeschreibung in HTML

Der Nachteil dieser Form der Kodierung ist es, dass es ohne physikalischen Zugang zu der Ressource nicht möglich ist, die Metadaten zu erstellen oder nachträglich zu ändern. Zudem wird das in der Metadatenbeschreibung verwendete DC.Title-Element möglicherweise im regulären TITLE-Tag der HTML-Seite erneut auftauchen, was zu Konsistenzproblem führen kann. Semantische Beschreibungssprachen für HTML-Seiten vermeiden diese Inkonsistenzen, haben aber andere Nachteile. [127]. Der große Vorteil einer Kodierung der Metadatenbeschreibung innerhalb der Ressource liegt hingegen in der Vermeidung des im Web als "tote Links" bekannten Problems. Wird eine Ressource gelöscht oder steht aus anderen Gründen für eine Suche nicht mehr zur Verfügung, so gilt dies automatisch auch für die entsprechende Metadatenbeschreibung. Ressourcen, die nicht mehr existieren, können daher nicht als Ergebnis einer Suchanfrage auftauchen

2.6.2 Kodierung zusammen mit der Ressource

Bei Metadatenbeschreibungen, die zusammen mit der Ressource kodiert werden, handelt es

sich in der Regel nicht um identifizierende, sondern um administrative Metadaten. Sie werden nach Auswahl der Ressource, zusammen mit den eigentlichen Mediendaten übertragen. Ein Beispiel für eine Kodierung der Metadaten zusammen mit der Ressource findet sich im HTTP-Header der Antwort eines Web-Servers auf eine Client-Anfrage. Der Server überträgt im Feld "Content-Type" des Response Headers beispielsweise den MIME-Typ der angehängten Information, sodass auf Seiten des Clients u.U. direkt ein Browser PlugIn geladen werden kann, um die angeforderte Ressource anzeigen zu können. [135]. Bei dieser Form der Speicherung der Metadaten ist das Format der Kodierung also ebenfalls vorgegeben; im Fall von HTML also durch das Format von HTTP.

2.6.3 Kodierung unabhängig von der Ressource

Bei der Kodierung der Metadaten unabhängig von der Ressource kann auch das Format der Metadatenkodierung unabhängig von der Ressource gewählt werden. Obwohl es fast so viele Ansätze zur Kodierung von Metadaten gibt, wie Metadatenschemata existieren, können große Teile jedes Schema durch eine entsprechende Extensible Markup Language (XML)-Beschreibung ausgedrückt werden. So beschreiben alle in Abschnitt 2.4 beschriebenen Schemata abstrakte und sprachenunabhängige Objekte. XML bietet sich an, diese Objekte zu realisieren, da diese Sprache ein Objektmodell definiert. Das Metadatenmodell wird in einer DTD deklariert und alle Dokumente, die diesem Schema entsprechen müssen, können dann in einer der DTD entsprechenden XML-Beschreibung gespeichert werden (siehe die DTD für das Learning Object Metadata Schema in Anhang A). XML ist eine textbasierte Meta-Auszeichnungssprache, die es ermöglicht, Daten bzw. Dokumente derart zu beschreiben und zu strukturieren, dass sie zwischen einer Vielzahl von Anwendungen ausgetauscht und weiterverarbeitet werden können. XML-Dokumente unterscheiden sich auf den ersten Blick nicht wesentlich von HTML-Dokumenten. Auch XML-Dokumente enthalten Tags als Auszeichner und Inhalt zwischen den Tags. Während die Anzahl und Benennung der Tags für HTML aber vorgegeben sind, können für XML-Dokumente beliebig viele und frei ("semantisch") benannte Tags verwendet werden. Somit besteht der wesentliche Unterschied darin, dass die Auszeichner Informationen über den Inhalt enthalten können und die Verschachtelung der Tags ineinander die Struktur der Daten abbilden kann.

Um Metadatenbeschreibungen mit Hilfe von XML zu beschreiben, muss für das entsprechende Schema eine XML-DTD definiert werden. In dieser können aber nur die Namen der Metadatenelemente festgelegt werden. Es ist nicht möglich, durch eine DTD Wertebereiche für die Werte der Metadatenelemente festzulegen. Zusätzlich ist es nicht möglich bei der Verwendung einer DTD zur Beschreibung des Metadatenschemas zu spezifizieren, was mit der Bezeichnung eines einzelnen Elements gemeint ist, oder anders ausgedrückt, welche semantische Bedeutung der Name des Elements besitzt. Bei der Verwendung einer XML-Beschreibung muss die Bedeutung des Metadatenelements klar sein. Enthält ein Metadatenschema beispielsweise ein <DATE>-Tag, so lässt sich nicht erkennen, ob damit das ursprüngliche Erstellungsdatum der Ressource gemeint ist oder das Datum der letzten Änderung. Ein anderes Beispiel ist das in Abbildung 6 gezeigte Problem der hierarchischen Strukturierung der Metadatenbeschreibungen.

In [82] ist beschrieben, dass es für die Repräsentation einer einfachen Online-Bestellung eine kaum mehr überschaubare Menge an DTDs existiert, die jeweils eine Datenstruktur für die Kodierung einer Bestellung festlegen. Obwohl alle DTDs den gleichen Sachverhalt ausdrük-

ken und teilweise auch identische Namen für die Tags verwenden, können Anwendungen, die aufgrund einer bestimmten DTD entwickelt wurden, XML-Beschreibungen, die konform zu einer anderen DTD kodiert wurden, nicht automatisch weiterverarbeiten. Bevor diese Beschreibungen genutzt werden können, muss eine Untersuchung der anderen DTD stattfinden, um herauszufinden, was mit den Bezeichnungen der Tags dieser DTD gemeint ist. Bevor also eine XML-Beschreibung bei der Suche eingesetzt werden kann, muss also auch hier für einen Benutzer die Bedeutung des Elements klar sein.

Da es mit XML zudem nur sehr eingeschränkt möglich ist Datentypen für die Werte der einzelnen Elemente zu definieren oder Kardinalitäten für die Häufigkeit des Auftretens bestimmter Elemente anzugeben, wurde vom W3C die Sprache XML Schema standardisiert, die diese Limitierungen aufheben [142]. Eine Spezifikation der semantischen Bedeutung der Elemente oder Werte ist aber auch mit XML Schema nicht möglich.

Aus diesem Grund wurde vom W3C das Resource Description Framework entwickelt [74]. RDF ist ein Rahmenwerk, das ermöglicht, strukturierte Metadaten zu kodieren und auszutauschen. Es basiert auf der XML-Struktur und erweitert sie um die Fähigkeit semantische Aussagen über eine Ressource auszudrücken. RDF basiert auf einem Graphmodell, der die Semantik der Metdaten ausdrückt. Das Basiskonzept von RDF besteht darin, dass eine Ressource durch eine Menge von “properties” (Eigenschaften), die RDF “description” (Beschreibung) genannt wird, beschrieben wird. Jedes dieser “properties” hat dabei einen eindeutigen Typ und einen Wert. Im Gegensatz zu Metadatenbeschreibungen, die nur auf XML basieren, können durch dieses Modell Aussagen über andere Ressourcen getroffen werden. Der Grundgedanke von RDF ist also ein Model zur Darstellung von Eigenschaften und Eigenschaftswerten. Das RDF Model stützt sich auf Prinzipien von verschiedenen Daten zur Darstellung von Ähnlichkeiten. RDF Eigenschaften können als Attribute von Quellen betrachtet werden und entsprechen den traditionellen Attributenpaaren [15]. RDF repräsentiert Verbindungen und Beziehungen zwischen Quellen, weshalb es auch einem Verbindungsdiagramm ähnelt. Im objektorientierten Design würden Quellen Objekten und die Quelleneigenschaften Instanzen entsprechen. Das RDF Datenmodell ist ein syntaxneutraler Weg zur Darstellung von RDF-Ausdrücken.

Im Gegensatz zu XML, das nur die Struktur der Metdaten beschreibt, können durch die Kodierung mit RDF Anfragen der Art: “Über was wird eine Aussage getroffen?” und “Wer trifft eine Aussage” gestellt werden. Es ist mit RDF aber ebenfalls nicht möglich, Fragen der Art “Was wird gesagt?” auszudrücken. Zur Beantwortung dieser Frage ist auch bei RDF die Einigung auf eine Vokabular und auf die Bedeutung der einzelnen Elemente notwendig. Aus diesem Grund wird derzeit vom W3C RDF-Schema entwickelt [16]. Durch RDF-Schema wird ein minimales Basisvokabular für die Bedeutung einer Aussage und für die Bedeutung von Beziehungen zwischen Metadatenbeschreibungen eingeführt.

Auswahl eines Kodierungsformats

Wie beschrieben, kann die Auswahl eines Metadatenschemas dazu führen, daß das Kodierungsformat nicht mehr frei gewählt werden kann. Wird ein kodierungsunabhängiges Schema gewählt so können die Eigenschaften der möglichen externen Kodierungsformate wie folgt zusammengefaßt werden:

- 1 Die beschränkten Möglichkeiten von XML erweisen sich als großer Vorteil bei der Ent-

wicklung von Anwendungen die XML Metadatenbeschreibungen analysieren oder erzeugen. Da keine Aussagen über die Eigenschaften oder Typen der Metadatenelemente spezifiziert werden können, lassen sich sehr einfach Parser und damit auch Anwendungen implementieren, die XML kodierte Metadatenbeschreibungen bearbeiten können. Je mehr Anwendungen existieren, desto mehr Beschreibungen existieren aber auch und können für eine Suche verwendet werden.

- 1 Obwohl mit XML Schema, RDF und RDF Schema mittlerweile 3 Erweiterungen von XML vom W3C entwickelt wurden, ist keine der Erweiterungen in der Lage, alle Schemata korrekt abzubilden, die durch eine MPEG-7 DDL ausgedrückt werden können.
- 1 Alle außer XML beschriebenen Sprachen sind noch nicht endgültig vom W3C spezifiziert. Dies bedeutet daß die endgültigen Versionen noch Änderungen enthalten werden. Anwendungen, die Metadatenbeschreibungen nach diesen Sprachen erzeugen oder analysieren müssen daher geändert werden, sobald Änderungen an der Spezifikation der Sprachen vorgenommen werden.
- 1 Spätestens mit der Entwicklung von RDF Schema lösen sich die Metadatenbeschreibungen von Aussagen über eine einzelne Ressource und beginnen Aussagen über Beziehungen zwischen Ressourcen oder allgemeine Aussagen in der Kodierung zu speichern. Für die Abbildung allgemeiner Aussagen existieren aber schon seit langer Zeit Sprachen aus dem Bereich der Wissensmodellierung, oder der semantischen Netze (vgl. Abschnitt 3.7 "Semantische Netze")
- 1 Eine Entscheidung über eine Sprache zur Kodierung von Metadatenbeschreibungen kann nicht unabhängig davon getroffen werden, wer die Beschreibungen erstellt. Während einfache Abhängigkeiten wie bsp. welches Objekt in welcher Szene eines Videos enthalten ist, automatisch extrahiert und damit auch kodiert werden können (vgl. Abschnitt 2.4.3 "MPEG-7" und Abschnitt 2.5.3 "Generierungsmechanismen"), müssen inhaltliche Beziehungen zwischen Ressourcen manuell erfaßt werden. Wie in Abschnitt 3.8.3 "Architektur" gezeigt werden wird, sind die Autoren der Metadatenbeschreibung einer Ressource aber nicht geeignet auch die Beziehungen zwischen Ressourcen zu beschreiben. Die Metadatenbeschreibung einer Ressource sollte sich daher nur auf Eigenschaften einer Ressource beziehen. Die mit RDF Schema möglichen Aussagen über Beziehungen hingegen sollten separat modelliert werden.

Die Aufzählung zeigt, daß die Spezifikation einer einfachen DTD derzeit am besten geeignet ist, wenn eine Entscheidung über das Kodierungsformat eines Metadatenschemas getroffen werden muß. Zudem zeigt eine in [102] beschriebene Analyse über die Verwendung von Metadatenschema im Web, daß nur für Dublin Core eine signifikante Anzahl von Metadatenbeschreibungen existieren. Dies wird dabei ausdrücklich darauf zurückgeführt, daß Dublin Core Metadaten sowohl direkt in den Web-Seiten, als auch mit Hilfe einer einfachen XML-Beschreibung kodiert werden können. Im folgenden Abschnitt, in dem Metadatenbeschreibungen für web-basierte Lernressourcen beschrieben werden, wird daher für den geeignetsten Ansatz nur eine DTD zur Beschreibung spezifiziert. Die Diskussion über die optimale Kodierung der Metadatenbeschreibungen wird erst in Abschnitt 3.8.1 "Datenmodell" wieder aufgenommen, nachdem die Datenmodelle für semantische Netze erläutert wurden.

3 Metadaten für multimediale Lernressourcen

Unter den vielen Vertretern computergestützter Lehr- und Lernsysteme¹, waren die Intelligen-ten Tutoriellen Systeme (ITS), die ersten Systeme, die sich mit der strukturierten Beschreibung der vermittelten Lerninhalte beschäftigten. Der Grund hierfür ist, daß tutorielle Systeme versu-chen, aufgrund von Informationen über den Kenntnisstand des Benutzers, Entscheidungen zu treffen, welcher Medienbaustein einem Benutzer als nächstes angeboten werden soll, ob dem Benutzer bestimmte Bausteine wahlweise angeboten werden sollen, oder wie die Bausteine präsentiert werden sollen. Um diese Entscheidungen treffen zu können, benötigt die tutorielle Komponente des Systems aber Informationen über die Bausteine, die zur Verfügung stehen. Veröffentlichungen über die verwendeten Datenmodelle einzelner Systeme finden sich aller-dings kaum. Sie sind zudem nicht darauf ausgerichtet externes Material mit in die Auswahl aufzunehmen. Lerninformation und Informationen über die Lerninformation sind in diesen Modellen nicht voneinander getrennt, sodas sich der Austausch von Bausteinen oder die Ein-bringung neuer Bausteine als eine komplexe Aufgabe darstellt. In [18] wird das Datenmodell des bekannten ITS Elm-Art als Netz von Begriffen, Kanten und Regeln beschrieben. Die Begriffe stellen praktisch die Schlagwörter des Wissensgebiets dar, das in den Texten vermit-telt wird und denen die Texte zugeordnet sind. Über eine Typisierung der zwischen den Begriffen bestehenden Kanten, können dann Algorithmen das durch die Verbindung der Begriffe entstehende Netz analysieren und passende Informationen auswählen. Das Datenmo-dell gehört damit in den Bereich der Semantischen Netze die in Abschnitt 3.7.4 beschrieben sind. In diesem Fall besitzen die Bausteine selbst aber keinerlei weitere Informationen über die Art und Inhalt des in ihnen enthaltenen Materials. Der Zweck der verwendeten Metadaten ist bei diesen Systemen, die Auswahl passender Bausteine. Eine allgemeine Suche nach Bausteinen ist nicht erforderlich. Das System kennt alle zur Verfügung stehenden Module, da ITS eben geschlossene Systeme sind, die keine externen Informationen in die Auswahl miteinbe-ziehen können. Durch die Verbreitung des WWW und der Übergang von ITS zu web-basierten Adaptiven Hypermedia Systemen (AHS), die als Präsentationsmedium einen gängigen Web-Browser verwenden, und demzufolge auch web-basierte Ressourcen als Bausteine verwenden, hat die Möglichkeit der Suche und Identifikation von web-basierten Bausteine für Lernsysteme eine zunehmende Bedeutung erlangt. Ein Vorschlag für einen Standard, der die Suchfunktio-nalität der Metadatenelemente weit stärker betont, als ITS oder AHS, wird von der Working Group P1484.12 des IEEE Learning Technology Standard Comitee (LTSC) erarbeitet, das Learning Object Metadata Schema (LOM). Obwohl noch weit von einem offiziellen Standard entfernt, wird dieser Entwurf von praktisch allen internationalen Organisationen die sich mit computergestütztem Lernen beschäftigen, unterstützt. Während bei ITS oder AHS, die Aus-tauschbarkeit und damit der Modulgedanke der Bausteine eine untergeordnete Rolle spielt, ist LOM sehr stark an den Inhalten der Ressource selbst ausgerichtet.

1. Eine detaillierte Klassifikation computergestützter Lehr- und Lernsysteme findet sich bsp. unter [11]

3.1 Learning Objects Metadata (LOM)

3.1.1 Einbettung von LOM

LOM ist Teil der vom LTSC definierten Learning Technology Systems Architecture (LTSA). Die LTSA ist der Versuch auf einer abstrakten Ebene, Komponenten und Anforderungen an allgemeine computerbasierte Lernsysteme, zu definieren. Dabei werden keine expliziten Lernstrategien, oder Szenarien vorgeschrieben, oder detaillierte Protokolle und APIs spezifiziert. Vielmehr wird versucht, Komponenten zu identifizieren und zu beschreiben, die in welcher Form auch immer, Bestandteil eines jeden computergestützten Lernsystems sind. LTSA ist daher in ein 5 Schichten-Modell eingebettet, das in Abbildung 11 gezeigt ist.

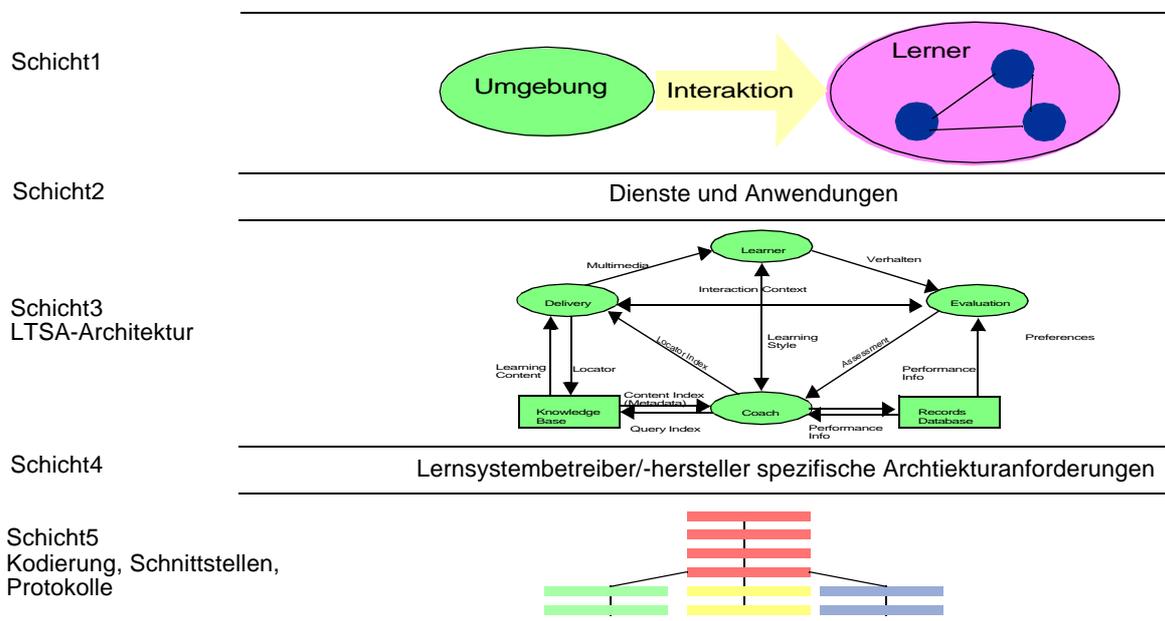
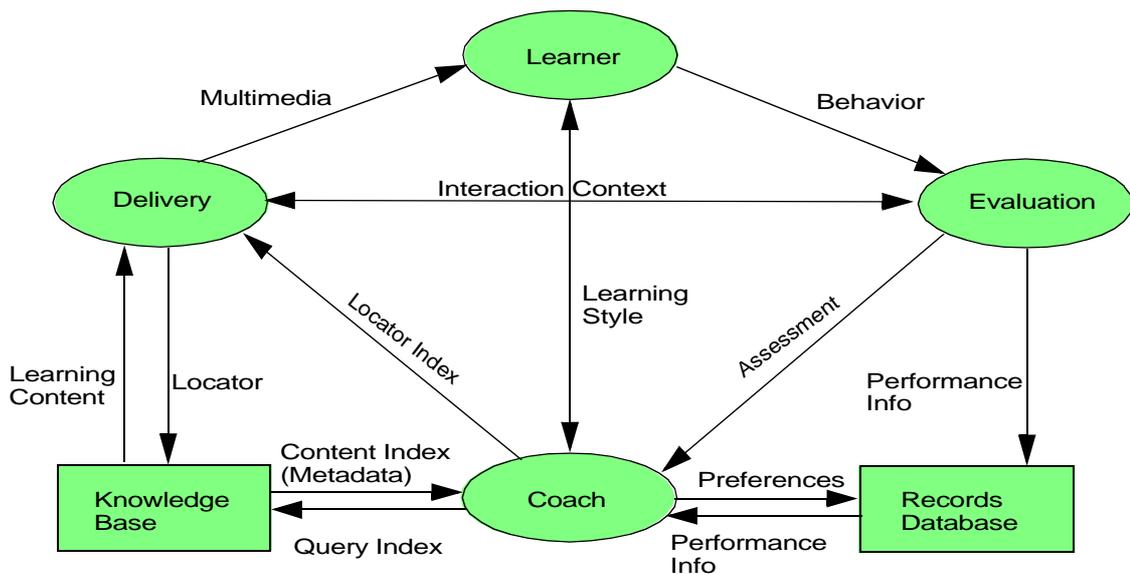


Abbildung 11: 5 Schichten-Modell der LTSA (vgl. [55])

Die in der Abbildung gezeigten Schichten 1, 2 und 4 sind nicht Teil der Standardisierung, sondern dienen dazu die allgemeinen Anforderungen an die Datenmodelle und Verhaltensweisen der Komponenten in Schicht 3 zu spezifizieren und zu gruppieren. Während in Schicht 3 die allgemeinen Komponenten eines Lernsystems und der Informationsfluß zwischen den Komponenten beschrieben ist, enthält Schicht 5, konkrete Schnittstellen und Protokolle zur Realisierung der Komponenten in verschiedenen Programmiersprachen. Die Beschreibung der Medienbausteine durch LOM ist dementsprechend Schicht 3 zuzuordnen. Die in verkleinert dargestellte Anordnung der Komponenten eines Lernsystems sind in vergrößerter Form in

Abbildung 12 wiedergegeben

**Abbildung 12:** Komponenten eines web-basierten Lernsystems ([55])

Die Abbildung zeigt die Rollen Lehrer (Coach) und Lerner (Learner), wobei der Coach keine menschliche Person sein muß, sondern auch aus Regelsätzen und Algorithmen bestehen kann, die passendes Material für den Lerner auswählen. Als Entscheidungsgrundlage stehen für den Lehrer Informationen über die Präferenzen oder das Vorwissen des Lerners zur Verfügung. Der Lehrer erhält diese Informationen entweder über eine direkte Befragung des Lerners (Learning Style), oder über das beobachtete Verhalten (Behaviour), das in einer Evaluation ausgewertet wird und in Form eines Benutzerprofils gespeichert ist (Records Database). Anhand dieser Informationen können durch den Coach Suchkriterien für Medienbausteine abgeleitet werden. Durch eine Suche auf den Metadatenbeschreibungen (Content Index, Query Index), können Verweise auf multimediale Ressourcen über eine Delivery-Komponente ausgeliefert werden, die für die Präsentation der Materialien verantwortlich ist. Obwohl die Wissensbasis (Knowledge Base) in der Abbildung keine übergeordnete Rolle einnimmt, ist sie von zentraler Bedeutung. Der komplette Informationsfluß zwischen Lerner und Lehrer dient nur dazu, Kriterien zu ermitteln, die sich für die Auswahl von passendem Material nutzen lassen. Entsprechender Bedeutung kommt damit der Fähigkeit einer Wissensbasis, Beschreibungen des von verwaltetem Material in Form von Metadaten anzubieten, die eine gezielte Auswahl von Medienbausteinen ermöglichen.

Als Schema zur Beschreibung der Metadaten, wird derzeit LOM erarbeitet. LOM basiert vollständig auf Dublin Core und erweitert es um lernspezifische Elemente. Wie Dublin Core, ist LOM entsprechend textorientiert. Die Entwicklung von LOM basiert maßgeblich auf den Datenmodellen für multimediale Lernressourcen des europäischen ARIADNE-Konsortiums [3] und den Empfehlungen des nordamerikanischen IMS Global Learning-Konsortiums [58]. Während es sich bei den Mitgliedern von ARIADNE derzeit ausschließlich um europäische Universitäten und andere akademische Einrichtungen handelt, bestehen die Mitglieder bei IMS sowohl aus nordamerikanischen Universitäten, als auch kommerziellen Unternehmen. IMS erarbeitet Standards und Spezifikationen, zur Schaffung von interoperablen Anwendungen für den Bereich der verteilten Lernsysteme und stellt Beispielimplementierungen zur Verfügung.

Interoperabilität zwischen Anwendungen kann aber nur dann gelingen, wenn zur Beschreibung der Lernmaterialien ein einheitliches Format verwendet wird. Aus diesem Grund hat IMS seit der Gründung des Konsortiums aktiv an der Entwicklung von LOM beteiligt.

Das ARIADNE Konsortium ist aus einem von der Europäischen Union im Rahmen des "4th Framework Telematics for Education and Training" geförderten, Projekt hervorgegangen. Ziel von ARIADNE ist es durch das Entwickeln von für Mitglieder kostenloser Software, europaweit Repositories für multimediale Lernressourcen zu etablieren.

3.1.2 Ziele von LOM

Ausgehend von den Zielen von ARIADNE und IMS sind die Ziele von LOM in [56] wie folgt definiert: ([14])

- 1 Lernenden und Lehrenden das Suchen, Auswerten, den Erwerb und den Gebrauch von Lernobjekten zu ermöglichen.
- 1 Den Austausch und gemeinsames Verwenden von Lernobjekten in jedem technisch unterstützten Lernsystem zu ermöglichen.
- 1 Das Entwickeln von Lernobjekten mithilfe von kombinierbaren und speicherbaren Einheiten zu ermöglichen.
- 1 Es Agenten zu erlauben, automatisch und dynamisch individuelle Kurse für einen Lerner zusammenzustellen.
- 1 Die Arbeit an anderen Standards zu unterstützen, die sich mit dem Problem beschäftigen, wie Lernobjekte in einer offenen, verzweigten Lernumgebung zusammenarbeiten können.
- 1 Die Dokumentation oder Vervollständigung von bisher existierenden Lernobjekten zu unterstützen.
- 1 Ein starkes, wachsendes System von Lernobjekten zu unterstützen, das alle Arten von Beiträgen unterstützt, nicht-kommerzielle, oder kommerzielle.
- 1 Es Lehr- und Lernorganisationen, egal ob öffentlicher, privater oder staatlicher Natur, zu ermöglichen, Inhalt und Gestaltung von Lernobjekten in einer Weise auszudrücken, die unabhängig vom Inhalt selber ist.
- 1 Entwickler mit einem Hilfsmittel zu versehen, um Daten zu sammeln und auszutauschen, die die Anwendbarkeit und Effektivität von Lernobjekten betreffen.
- 1 Einen Standard zu definieren, der einfach ist, und in vielen Bereichen leicht angewandt und angepaßt werden kann.
- 1 Die nötige Sicherheit und Kennzeichnung für die Verbreitung von Lernobjekten zu unterstützen.

Unter Lernobjekten werden dabei ausdrücklich digitale und nicht-digitale Ressourcen verstanden. Es ist daher möglich mit LOM eine Vorlesung zu beschreiben, die nicht elektronisch zur Verfügung steht. Zudem macht der Vorschlag keinerlei Aussagen über die Größe einer Resource. Es soll also möglich sein, sowohl komplette Kurse mit einer LOM-Struktur zu beschreiben, als auch einen atomaren Baustein, wie bsp. ein einzelnes Bild oder eine einzelne

Animation. Entscheidend ist also das Wiederverwendungspotential einer Ressource und nicht ihre Größe. Dieser sehr umfassende Ansatz der LOM führt zu einer großen und allgemein gehaltenen Anzahl von Metadatenelementen die im nächsten Abschnitt beschrieben ist.

3.1.3 Elemente von LOM

LOM gruppiert zur besseren Übersicht alle Metadatenelemente in insgesamt 9 Kategorien, die sich wie folgt zusammensetzen:

- 1 Allgemeine Klasse (General)
Stellt alle kontextunabhängigen Eigenschaften der Ressource zusammen, sowie dessen semantische Beschreibungen.
- 1 Lebenszyklus Klasse (LifeCycle)
Enthält die Eigenschaften, welche im gesamten Lebenszyklus einer Ressource auftreten können.
- 1 Meta-Metadaten (Meta-Metadata)
Diese Klasse enthält Informationen und Beschreibungen über die Metadaten selbst.
- 1 Technische Anforderungen (Technical)
Diese Klasse beschreibt die Soft- und Hardwareanforderungen, welche für die Benutzung der Ressource notwendig sind.
- 1 Lehre (Educational)
Beschreibt die pädagogischen und für die Lehre wichtigen Eigenschaften einer Ressource.
- 1 Rechte (Rights)
Beschreibt unter welchen Voraussetzungen eine Ressource zu benutzen ist.
- 1 Beziehungen (Relation)
Stellt die Beziehungen der Ressource im Vergleich zu anderen dar.
- 1 Kommentare (Annotation)
Erlaubt Notizen über die pädagogische Nutzung dieser Ressource.
- 1 Klassifikation (Classification)
Gruppiert, kategorisiert und ordnet die entsprechenden Ressourcen in einer systematischen Art und Weise.

Diese Kategorien beschreiben das sogenannte LOM Basisschema. Alle Kategorien sind beliebig erweiterbar, und können so an individuelle Gegebenheiten angepaßt werden.

3.1.4 Struktur von LOM

Die in 3.1.3 beschriebenen 9 Kategorien sind baumartig in weitere Unterkategorien aufgeteilt. Die Blätter des Baumes sind die in LOM wie folgt definierten Datenelemente:

- 1 Name:
Datenelemente können nicht über einen eindeutigen Namen identifiziert werden, da sie abhängig von der Kategorie der sie zugeordnet sind, unterschiedliche Bedeutungen

haben können. Zur eindeutigen Identifizierung muß daher die Kategorie(n), in denen das Datenelement enthalten ist, jeweils mit angegeben werden. Zusätzlich kann die Bedeutung der Werte einzelner Datenelemente von den Werten anderer Datenelemente abhängen, sodass eine unabhängige Bewertung nicht möglich ist.

Beispiel: Sowohl in der MetaMetaData-Kategorie, als auch in der LifeCycle-Kategorie existiert ein Metadatenelement "Contribute". Während bei der MetaMetaData-Kategorie in diesem Element Informationen über Personen gespeichert werden, die für die Entwicklung des verwendeten Metdatenschemas verantwortlich sind, gespeichert werden, beziehen sich die Informationen der LifeCycle-Kategorie auf die Bedeutung die eine Person, während des Erstellungsprozeß der Ressource hat.

1 Vokabulare

Alle Datenelemente von LOM werden als Zeichenketten kodiert. Neben der Kardinalität der Elemente wird zwischen den Basis-Datentypen VOCABULARY und LANGSTRING unterschieden. Für die Werte eines VOCABULARY-Element gilt, daß er nicht frei vom Autor einer Beschreibung gewählt werden kann, sondern aus einem fest vorgegebenen Vokabular ausgewählt wird. Dabei wird im LOM-Vorschlag zwischen "restricted" und "best practice"-Einträgen unterschieden.

Beispiel: Das Datenelement "Aggregation Level" der "General"-Kategorie beschreibt die Granularität der Ressource. Als Werte stehen dabei nur Zahlen von 0-4 zur Verfügung, wobei 0 angibt, daß es sich bei der Ressource um ein Fragment oder Teil einer umfassenderen Ressource handelt. Bei einer Ressource mit dem Wert 4 handelt es sich dann um einen vollständigen Kurs.

1 Kardinalität

Mit wenigen Ausnahmen, können sowohl Unterkategorien, als auch Datenelemente mehrmals in einer LOM-Beschreibung auftauchen. Während die Reihenfolge der Anordnung der Kategorien oder Unterkategorien keine Rolle bei der Bewertung spielt, gibt es Datenelemente, bei denen die Reihenfolge der Werte eine Bedeutung hat.

Beispiel: Das Metadatenelement "Learning Resource Type" der Educational-Kategorie wird zur Speicherung des Ressourcentyps verwendet. Obwohl als mögliche Werte ein festes Vokabular vorgegeben ist, kann eine Ressource in der Regel nicht mit der Auswahl nur eines Wertes beschrieben werden. Aus diesem Grund kann das Feld mehrfach verwendet werden, wobei die Reihenfolge aussagt, welchem Typ die Ressource am ehesten zuzuordnen ist.

1 Neben den Vokabularen, existiert wie beschrieben noch das Datenelement LANGSTRING zur Speicherung der Werte der Datenelemente. Das LANGSTRING-Element unterscheidet sich von einer herkömmlichen Zeichenkette dadurch, daß jeweils zu, eigentlichen Wert noch zusätzlich ein Sprachkürzel gespeichert wird, durch das die verwendete Sprache bei Wert des Elements spezifiziert wird.

Beispiel: Wird im Metadatenelement "Title" der General-Kategorie ein deutscher Titel beschrieben so enthält das Metadatenelement neben dem Titel : "Multimedia: Grundlagen Komponenten und Systeme" zusätzlich noch die Zeichenkette "de" gespeichert. Da jedes LANGSTRING-Element aus einer beliebigen Anzahl von Sprach/Wert-Tupels bestehen kann, können in einer LOM-Beschreibung vollständige Beschreibungen in unterschiedlichen Sprachen gespeichert werden.

3.2 Programmgestützte Generierung von LOM-Metadaten

In den folgenden Kapiteln soll auf die verschiedenen Möglichkeiten der computer-gestützten Generierung von Feldeinträgen, des LOM Basis Schemas, eingegangen werden. Durch die Einbeziehung bestimmter Vorgaben oder Einschränkungen beim Einsatz des LOM-Editors in bestimmten Projekten, können die Generierungsmechanismen einzelner Datenfelder von den allgemeingültigen Erfassungsmöglichkeiten abweichen. Die verschiedenen Möglichkeiten einer computergestützten Generierung werden im Folgenden hinsichtlich der zur Automatisierung notwendigen Informationen in verschiedene Gruppen unterteilt. Die erste Differenzierung erfolgt anhand der Ebene, auf der einzelne Merkmale extrahierbar sind, die zweite bezieht die notwendigen Zusatzinformationen mit ein. Die nachfolgende Auflistung der möglichen Generierungsmechanismen soll die Gliederung in einzelne Gruppen verdeutlichen.

- 1 Eine automatische Generierung des Elementeintrages ist auf Feature-Level eindeutig möglich. Die aus der Ressource gewonnene Information lässt in der Regel nur einen bestimmten Eintrag zu.
Beispiel: Die Größe einer digitalen Ressource ist eindeutig durch die Datei vorgegeben und ohne Benutzerinformationen weiter zu verarbeiten.
- 1 Eine automatische Generierung des Elementeintrages ist auf Feature-Level bedingt möglich. Um einen Eintrag vornehmen zu können sind allerdings weitere Informationen notwendig, die aber unabhängig von einer spezifischen Ressource sind.
Beispiel: Audiodateien sind durch die Dateinamenerweiterung als solche zu erkennen. Durch zusätzliche Informationen ist es möglich, den Eintrag "Soundkarte" im Datenelement "OtherPlatformRequirements" zu ergänzen.
- 1 Eine automatische Generierung des Elementeintrages ist auf Concept-Level eindeutig möglich. Die durch inhaltliches Wissen gewonnene Information lässt in der Regel nur einen bestimmten Eintrag zu.
Beispiel: Die in einer Ressource primär verwendete Sprache gibt den Eintrag in das Datenelement "Language" vor.
- 1 Eine automatische Generierung des Elementeintrages ist auf Concept-Level bedingt möglich. Je nach Definitionsgrundlage sind verschiedenartige Eintragungen in Bezug auf Elementwert und Datenumfang möglich.
Beispiel: Die in "Coverage" erfaßten Informationen des Umfangs und der Spanne bezüglich Zeit und Ort erlauben einen sehr großen Interpretationsspielraum. Ein eindeutiger Eintrag ist in diesem Fall fast nicht möglich.
- 1 Kein Eintrag möglich, da das Feld zum jetzigen Zeitpunkt noch reserviert ist oder kein Datenelement repräsentiert.
- 1 Auch mit allen notwendigen Informationen über eine Ressource ist kein automatischer Eintrag möglich.

3.2.1 General

Soll eine Ressource nach seiner Erstellung und der Generierung der LOM-Daten einem fremden Benutzerkreis zur Verfügung gestellt werden, so ist eine automatische Generierung des Eintrags im Feld "Title" in der Regel nicht sinnvoll. Steht allerdings die Verarbeitung der Res-

source in internen Anwendungen im Vordergrund, ist eine automatische Erstellung des Titels ratsam. Da das Datenelement "Identifier" aufgrund noch fehlender Methoden zur Erstellung einer weltweit eindeutigen Identifikation reserviert ist, stellt "Title" momentan die einzige Möglichkeit zur Identifikation des Lernobjektes dar. Aus diesem Grund sollte darauf geachtet werden, dass der Titel der Ressource eindeutig ist, d.h. nicht mehrere Dateien mit identischem Namen existieren. Werden die Einträge im Feld "Title" allerdings manuell vorgenommen, besteht die Möglichkeit, an verschiedene Ressourcen den gleichen Titel zu vergeben. Die Wahrscheinlichkeit hierfür wird umso höher, je mehr Personen parallel mit der Generierung der Metadaten beschäftigt sind und je ähnlicher die zu verarbeitenden Ressourcen sind. Dieses Problem lässt sich umgehen, indem man den Titel einer Ressource nach einem bestimmten Schema erstellt.

Eine automatische Erstellung von Schlüsselwörtern ist für die meisten Medien nur mit bereits vorhandenem Anwenderwissen möglich und selbst dann in nur sehr eingeschränktem Rahmen. Einzige Ausnahme hierbei sind Textdateien. Hier besteht die Möglichkeit, einzelne Wörter oder Textpassagen, ohne Wissen über den eigentlichen Inhalt, aus dem Dokument zu extrahieren. Denkbar wäre ein Verfahren, das die Substantive und Eigennamen innerhalb eines Dokumentes nach ihrer Auftrittswahrscheinlichkeit ordnet und die am häufigsten auftretenden Begriffe als Schlüsselwörter in den Metadatensatz übernimmt. Dieser Ansatz wird so lange brauchbare Ergebnisse liefern, solange ein Text durch seinen Inhalt in ausreichender Weise charakterisiert wird. Probleme werden dann auftreten, wenn ein Dokument in der hierarchischen Struktur einer Lernsequenz relativ weit unten eingeordnet ist.

Das Datenfeld "Coverage" dient eigentlich der Beschreibung von Epoche und geografischer Region, auf die sich der Inhalt einer Ressource bezieht. Aus einem für alle Medien einheitlichen Eintrag in der Form "Deutschland 21. Jahrhundert" wird in der Regel kein nennenswerter Gewinn an zusätzlichen Informationen für die weitere Verarbeitung der Ressourcen resultieren. Die logische Konsequenz hieraus wäre das Ignorieren des Eintrags. Da die offizielle Beschreibung des Datenfeldes durch das Learning Technology Standardization Committee (LTSC) aber einen wesentlich größeren Spielraum in Bezug auf die Verwendung dieses Feldes zulässt, kann es zur weiteren, verfeinerten Charakterisierung der Ressourcen verwendet werden. Die zugrundeliegende, organisatorische Struktur einer Ressource lässt sich in der Regel schon auf Feature-Level aus dem Dateiformat bestimmen. Geht man bei den zu verarbeitenden Ressourcen von Rohdaten (Text, Audio, Grafik, etc.) aus, so ist die Struktur durch die Dateinamenerweiterung in ausreichendem Maße bestimmt. Ausnahmen, wie zum Beispiel ein Hypertext-Dokument, bei dem die Informationen ohne entsprechende Links innerhalb des Dokuments und somit in linearer Form vorliegen, sollen hier nicht weiter berücksichtigt werden. Bei aggregierten Ressourcen ist eine eindeutige Bestimmung der organisatorischen Struktur alleine durch Dateiinformationen nicht mehr möglich. Ein aus mehreren Medien zusammengestellter Kurs kann in beliebiger Form organisiert sein, ohne dass dies Auswirkung auf das Dateiformat haben muss. Zur Klassifizierung der Organisationsstruktur muss deshalb Anwendungswissen zur Verfügung stehen. Das Datenelement "AggregationLevel" beschreibt die funktionale Granularität einer gegebenen Ressource mittels eines eingeschränkten Vokabulars der Werte 0 bis 4. Der Wert "0" wird als kleinste Dateneinheit bzw. Datenfragment definiert, steht also stellvertretend für Rohdaten. Die nächste Stufe ist durch die Kombination von mehreren Rohdaten gekennzeichnet, beschreibt damit beispielsweise ein Hypertextdokument mit eingebetteten Bildern. Eine Ressource mit einem "AggregationLevel" der Stufe 2 umfasst mehrere Ressourcen der Stufe 1. Als Beispiel hierfür ist eine Menge einzelner HTML-Dokumente genannt, die über eine gemeinsame Index-Seite verfügen, welche die einzelnen Doku-

mente über Links miteinander verknüpft. Die höchste Stufe der Granularität wird durch ein "AggregationLevel" der Stufe 3 beschrieben, die im Allgemeinen nur durch fertig zusammengestellte Kurse erreicht wird. Eine Klassifizierung der Ressource anhand ihrer Dateinamenerweiterung ist für die Granularitätsstufe 0, 1 und 3 ohne Kenntnis des Dateiinhaltes möglich. Probleme treten allerdings bei Ressourcen auf, deren "AggregationLevel" im Bereich der Stufe 2 liegt. Eine Menge zusammengehörender HTML-Dokumente lässt sich nämlich nicht durch Dateinamen alleine als solche erkennen, vielmehr sind dazu Kenntnisse über die inneren Strukturen der Dokumente notwendig. Die Generierung eines exakten Feldeintrages ist für dieses Datenelement also lediglich auf dem Concept-Level möglich.

3.2.2 Life Cycle

Bei den ersten beiden Datenelementen "Version" und "Status" dieser Kategorie ist eine automatische Generierung der Einträge mittels eines Vorgabewertes im Großteil aller Fälle als sinnvoll zu erachten. Im Normalfall wird eine einmal erstellte Ressource im weiteren Verlauf der Anwendung nicht mehr verändert. Trotz allem muss die Möglichkeit bestehen, überarbeitetes Datenmaterial kenntlich zu machen. Gerade bei verteilter oder redundanter Speicherung von Ressourcen ist ein einfaches Ersetzen aller Dateien nicht immer möglich. Eine Belegung der Datenfelder "Version" und "Status" mit Standardwerten, die bei Bedarf geändert werden können, scheint eine zufriedenstellende Lösung des Problems darzustellen. Bei Eintragungen in die Felder "Contribute.Role" und "Contribute.Entity" ist zu beachten, dass der Ersteller des Metadatensatzes nicht notwendigerweise mit dem Autor der Ressource übereinstimmen muss. Informationen über Autor oder Herausgeber einer Ressource sind meistens aus den Dateiinformatoren zu übernehmen und somit automatisch zu generieren. Schwierigkeiten treten auf, falls mehrere Personen oder Organisationen an der Erstellung einer Ressource in verschiedenen Entwicklungsstufen beteiligt waren, Name und Aufgabe aber nicht in den Dateiinformatoren enthalten sind. In diesem Fall ist der Mehraufwand, der im Vorfeld einer computergestützten Generierung von Metainformationen notwendig ist, der Zeitersparnis durch die Nutzung vordefinierter Benutzergruppen gegenüberzustellen und im Einzelfall zu entscheiden. Als Datum des Beitrages im Datenelement "Contribute.Date" kann im Normalfall mit ausreichender Genauigkeit das Erstellungsdatum bzw. das Datum der letzten Änderung übernommen werden.

3.2.3 MetaMetadata

Bei der programmgestützten Erfassung der Meta-Metadaten ist zu berücksichtigen, dass sich Informationen auf Feature-Level nur aus den eigenen Systemdateien sowie den Informationen eines unter Umständen vorhandenen Editors extrahieren lassen. Eine Nutzung von Dateiinformatoren der entsprechenden Ressource ist nicht möglich. Im Gegensatz zu den Eintragungen bezüglich der Elemente "Contribute.Role" und "Contribute.Entity" in der Kategorie "Life-Cycle" liegen hier alle notwendigen Informationen vor. Die Benutzung der gespeicherten Anwenderinformationen sollte in der Regel zur vollständigen Beschreibung beider Datenfelder ausreichen. Zusatzinformationen lassen sich in diesem Zusammenhang über vorgefertigte Benutzerprofile ergänzen. Das Datum des Beitrags entspricht der aktuellen Systemzeit und lässt sich somit einfach übernehmen. Die Elemente "MetaDataScheme" und "Language" sind mit Hilfe von Programminformationen eines Editors zu vervollständigen. Voraussetzung hierfür ist allerdings die konsequente Nutzung einer Primärsprache für alle zu bearbeitenden Res-

sourcen.

3.2.4 Technical

Die technischen Voraussetzungen, die notwendig sind, um eine entsprechende Ressource zu nutzen, sind in der Regel aus den Medientypen abzuleiten. Da der Medientyp schon auf Feature-Level aus der Dateinamenerweiterung ersichtlich ist, ist zusätzliche, semantische Information in den meisten Fällen irrelevant. Im Gegensatz zu den Feldern "Format", "Size", "Location" und "Duration" muss hier aber zusätzliches Anwenderwissen zur Verfügung gestellt werden. Eine systemgestützte Generierung der Feldeinträge setzt voraus, dass im Vorfeld für alle Medientypen eine Liste der entsprechenden Hard- und Softwareanforderungen definiert wurde, auf die das System zurückgreifen kann. Die Datenelemente "Requirements.MinimumVersion" und "Requirements.MaximumVersion" bedürfen in diesem Kontext allerdings einer gesonderten Behandlung, da nicht alle Dateien Rückschlüsse auf die Versionsnummer der Erstellungssoftware zulassen. Eventuell ist bei diesen beiden Feldern von einer automatischen Generierung der Metadaten abzusehen. Metadaten bezüglich der Installation von Lernobjekten sind nur bei zusammengesetzten Ressourcen, wie etwa abgeschlossenen Kursen, notwendig. Ob aufgrund der Anzahl der zu erfassenden Lernobjekte die Notwendigkeit einer automatischen Generierung von Installationsanforderungen besteht, ist im Einzelfall zu entscheiden.

3.2.5 Educational

Das Datenelement "InteractivityType" gibt ein eingeschränktes Vokabular vor, wodurch eine automatische Generierung des Eintrags ermöglicht wird. Die von der IEEE definierte Einteilung einzelner Medienobjekte in die vorgegebenen Interaktivitätsklassen lässt auch hier wieder eine Klassifizierung der Ressource durch deren Dateinamenerweiterung zu.

Um die Art der Lernressource zu definieren, steht dem Benutzer im Feld "LearningSourceType" ein offenes Vokabular zur Verfügung, wobei gleichzeitig "Best-Practice"-Werte vorge schlagen werden. Eine Beschränkung auf diese Eingabe-möglichkeiten ist aber als Grundlage für eine automatische Erstellung nicht ausreichend. Eine Datei, die durch ein Tabellenkalkulationsprogramm erstellt wurde, ist zwar als solche zu erkennen, eine Aussage, ob es sich um eine Tabelle, einen Graphen oder ein Diagramm handelt, ist indes nicht möglich. Genauso wenig ist es möglich, eine Textdatei ohne semantische Informationen in die Bereiche "Aufgabe", "Lösung" oder "Erklärung" einzuteilen.

Auch die weiteren Datenfelder dieser Kategorie besitzen ein eingeschränktes Vokabular oder zumindest "Best-Practice"-Vorgaben, falls keine Einschränkung vorgesehen ist. Die gegebenen Voraussetzungen sind aber nicht ausreichend, um eine Generierung der Metadaten durch das System zu realisieren. Zur Erstellung eines Feldeintrages reichen die ausschließlich aus den Daten extrahierten Informationen in der Regel nicht aus. Vielmehr muss fachspezifisches Wissen als Grundlage der Klassifizierung einer Ressource vorhanden sein. Es wird also keine reine Beschreibung der Ressource vorgenommen, sondern eine Bewertung des Inhaltes. Als Grundlage dieser Bewertung können subjektive Einschätzungen durch den Anwender, aber auch vorher definierte Referenzwerte dienen.

3.2.6 Rights

Prinzipiell ist es nicht möglich, Eigentumsrechte aus den Dateinformationen oder dem Dateinhalt abzuleiten. Die Copyright- und Nutzungsbedingungen einer Ressource stehen nicht in direkter Abhängigkeit zum entsprechenden Datenmaterial, sondern werden vom Ersteller der Ressource definiert. Geht man davon aus, dass die Nutzungsbedingungen aller erstellten Ressourcen identisch sind, lässt sich die gesamte Kategorie "Rights" mit Vorgabewerten belegen und muss somit vom Benutzer nicht ausgefüllt werden.

3.2.7 Relation

Wie auch in der vorangegangenen Kategorie sind die Relationen der einzelnen Lernobjekte untereinander nicht aus dem Datenmaterial zu extrahieren. Die Beziehungen der Objekte untereinander werden vom Benutzer unter Berücksichtigung des Ressourceninhaltes und des beabsichtigten Lernziels definiert. Da zur Erreichung eines Lernzieles im Allgemeinen mehrere Wege möglich sind und somit verschiedene Möglichkeiten bestehen, die Lernobjekte zueinander in Beziehung zu setzen, ist eine automatische Generierung nicht ohne weiteres zu realisieren. Allenfalls der Katalogeintrag, der einer Zielressource in einem anderen Identifikationssystem zugewiesen ist, lässt sich unter bestimmten Voraussetzungen automatisieren.

3.2.8 Annotation

Informationen über den Verfasser des Kommentars lassen sich auf den oben schon beschriebenen Wegen automatisch erzeugen. Auch auf die Erfassung der Systemzeit als Eintrag in das Datenelement "Date" wurde bereits eingegangen. Das Verfassen der eigentlichen Anmerkung ist allerdings nur durch den Autor persönlich möglich.

3.2.9 Classification

Die Einordnung eines Lernobjektes in ein gegebenes Klassifizierungssystem ist nur vom Anwender selbst durchzuführen. Das Wissen, das zur Klassifizierung einer Ressource notwendig ist, übersteigt den Informationsgehalt des Datenmaterials um ein Vielfaches. Die Eingliederung in eine hierarchische Struktur setzt nicht nur Kenntnisse über das zu charakterisierende Objekt voraus, sondern über alle Ressourcen eines Klassifizierungs-raums. Nur aus diesem Wissen lässt sich eine Relation zwischen Objekten definieren und eine Einordnung in ein bestimmtes System vornehmen.

3.2.10 Zusammenfassung und Bewertung

Die Auswertung der detaillierten Ergebnisse der einzelnen Kategorien zeigt, dass eine computergestützte Generierung der Metadaten auf Concept-Level für über die Hälfte (52%) aller Datenelemente möglich ist. [108]. Selbst ohne semantische Informationen kann noch über ein Drittel (36%) der notwendigen Informationen automatisch aus dem Datenmaterial extrahiert

werden. Auffallend hierbei ist, dass sich die Beschreibung einiger Kategorien fast vollständig automatisieren lässt (z.B. Technical), bei anderen (z.B. Relation, Educational) wiederum kein einziges Element computergestützt generierbar ist. Es sei noch einmal darauf hingewiesen, dass sämtliche Aussagen bezüglich der Automatisierbarkeit der Feldeinträge auf Basis eines heterogenen Datensatzes getroffen wurden. Besondere Anwendungsfälle, in denen einzelne, vom eigentlichen Datenmaterial unabhängige Feldeinträge über einen kompletten Datensatz hinweg konstant bleiben, wurden nicht berücksichtigt.

3.3 Datenmodell für LOM

3.3.1 Struktur

Das Datenmodell der LOM Definition ist hierarchisch gegliedert. Auf der obersten Ebene dieser Hierarchie befindet sich das sogenannte Wurzelement („**Root**“). Dieses Wurzelement enthält Unterelemente. Zu unterscheiden ist hierbei zwischen Unterelementen, die weitere Unterelemente enthalten und daher als Zweig („**Branch**“) bezeichnet werden und solchen, die keine Unterelemente besitzen und als Blätter („**Leaves**“) bezeichnet werden. Die entstehende Hierarchie bildet somit eine Baumstruktur. Jedes Element der Metadatenhierarchie besitzt eine spezifische Definition, Datentyp und zulässige Werte. Die Relationen zwischen Root, Branch und Leaves wird in Abbildung 13 verdeutlicht

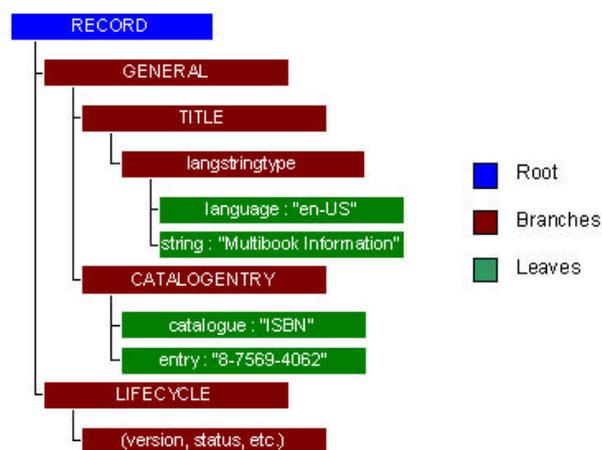


Abbildung 13: Baumstruktur der Learning Object Metadata

Die eigentlichen Inhalte bzw. Metadateninformationen für das jeweilige Lernobjekt werden in den Blattknoten des Baumes eingetragen.

Da der LOM-Entwurf keinerlei Aussagen macht, wie die Metadatenbeschreibung eines Medienbausteins zu kodieren ist, kann man prinzipiell alle in Abschnitt 2.6 beschriebenen Verfahren zur Kodierung der Metadaten verwenden. Es besteht also die Möglichkeit bei visuellen Medienbausteinen analog zu einer MPEG7-Beschreibung einen LOM-Metadatenatz direkt im Medienbaustein selbst zu kodieren. Da sich LOM-Beschreibungsdaten aber nur mittelbar direkt auf einzelne Inhalte beziehen und keine eindeutig identifizierbaren und adressierbaren

Teile des Mediums beschreiben, ist eine direkte Kodierung zusammen mit dem Medium nur von eingeschränktem Nutzen. Vielmehr hat diese Art der Kodierung den gravierenden Nachteil, daß keine Aussage über den Medienbaustein getroffen, werden kann, ohne über die physikalische Ressource selbst zu verfügen. Soll also die Möglichkeit der Wiederverwendung des Videomitschnitts einer Vorlesung beurteilt werden, so kann dies im Zweifel erst geschehen, wenn die unter Umständen sehr große Videodatei komplett zur Verfügung steht. Problematisch gestaltet sich auch die Kodierung, wenn es sich bei einem Medienbaustein nicht um einen monomedialen Typ handelt, sondern er sich aus mehreren unterschiedlichen Medien zusammensetzt. Bei der Kodierung innerhalb des Bausteins, muß dann unter Umständen in jedem Teilbaustein, eine redundante Beschreibung eingefügt werden, sofern das Medium überhaupt eine gemeinsame Kodierung von Metadaten und Mediendaten erlaubt. Aus diesem Grund kommt für die Kodierung von LOM-Beschreibungen nur die separate Speicherung und Verwaltung unabhängig von der Ressource in Frage. Wie in Abschnitt 3.3.1 beschrieben, läßt sich für praktisch jede Metdatenbeschreibung eine XML DTD zur Kodierung der einzelnen Beschreibungen spezifizieren. Da durch das Datenmodell als eine Baumstruktur spezifiziert ist, läßt sich LOM in einer XML DTD beschreiben.

3.3.2 Document Type Defintion für LOM

Entscheidend für die Funktionalität einer Wissensbasis, die multimediale Lernressourcen mit LOM beschreibt ist die Austauschbarkeit des Beschreibungsformats..... Nur über eine einheitliche Kodierung der Beschreibungen kann eine verteilte Wissensbasis entstehen, auf die von zentraler Stelle aus zugegriffen werden kann, und in der alle Beschreibungen ohne Transformation der Kodierung in ein anderes Format direkt verwendet werden können. Als Format bietet sich daher die Entwicklung einer in Abschnitt 2.6.3 beschriebenen XML DTD für LOM an. Erst wenn eine solche DTD zur Verfügung steht, können Anwendungen entwickelt werden die externe multimediale Ressourcen anhand Ihrer LOM-Beschreibungen bewerten und integrieren können. Obwohl die Baumstruktur des LOM-Schemas für die Spezifikation einer DTD sehr geeignet ist, ergeben sich bei der direkten Umsetzung Probleme.

- 1 Die Möglichkeit, daß Kategorien und Datenelemente mehrfach auftreten können, läßt sich in einer DTD problemlos ausdrücken, da auch hier spezifiziert werden kann, daß ein Element mehrfach in einer Beschreibung enthalten sein kann. Hingegen gibt es keine Möglichkeit in einer DTD auszudrücken, daß die Reihenfolge des Auftauchens in einer XML-Struktur eine Bedeutung für die Aussage der Metadatenbeschreibung enthält.
- 1 Mittels einer DTD kann nur ein Datentyp für die Blattelemente einer XML Beschreibung festgelegt werden. Es ist also bei der Umsetzung des VOCBAULARY-Elements des LOM Vorschlags auf eine XML-Struktur weder möglich zu verhindern, daß nicht LOM konforme Werte in den Datenelementen gespeichert werden, noch kann die mögliche Menge an Werten in der DTD kodiert werden.
- 1 Setzt man den LOM-Vorschlag direkt in eine XML DTD um, so wird man für die Relation-Kategorie eine eigene XML-Struktur definieren. Da, wie in Abschnitt 2.6.3 erwähnt mit einer XML-Struktur keine Aussagen über andere Metadatenbeschreibungen gemacht werden können, wird sich die Struktur für die Relation-Kategorie, nicht von den Strukturen der anderen Kategorien unterscheiden. Dies ändert sich aber, wenn man die Ressource mit einer RDF-Struktur beschreiben möchte. RDF kodierte Beschreibungen sind durch den "Resource - Property - Value"-Aufbau direkt darauf ausgelegt, Aussagen über andere Ressourcen zu treffen. Es ist daher nicht ohne weiteres möglich XML-Beschrei-

bungen von Ressourcen direkt in eine RDF-Beschreibung zu überführen.

Das Problem der Reihenfolge der Elemente, oder die Beschränkung der Wertemenge für bestimmte Datenelemente lässt sich durch die Verwendung von XML Schema-Beschreibungen umgehen. XML Schema ist ein vom W3C entwickelter Vorschlag, zur Beschreibung semantischer Strukturen [142]. Es hat damit zwar eine ähnliche Zielsetzung wie RDF Schema, die Anfragemöglichkeiten sind aber im Gegensatz zu RDF Schema beschränkt [32]. Zudem ist der Vorschlag derzeit noch in Bearbeitung und alle verfügbaren Implementierungen haben nur prototypischen Charakter. Bei XML ist es durch die eingeschränkte Aussagemöglichkeit der Strukturen deutlich einfacher Implementierungen zu realisieren, die XML-Strukturen analysieren und erstellen können. Aus diesem Grund existieren für viele Programmiersprachen stabile und frei erhältliche Implementierungen. Die Spezifikation einer DTD für LOM ist Teil der Schicht 5 der in Abbildung 11 gezeigten allgemeinen Architektur der LTSA. Von den dort vorgesehenen Protokollen und Schnittstellendefinitionen, ist zum jetzigen Zeitpunkt keine Spezifikation verfügbar. Der Grund hierfür ist, daß innerhalb der LTSC vereinbart wurde, mit der Spezifikation zu warten, bis LOM offiziell als IEEE-Standard verabschiedet ist. Bis zu dieser offiziellen Standardisierung können aber noch mehrere Jahre vergehen, da der LOM-Vorschlag auf dem Weg von einem Entwurf zu einem Standard eine Vielzahl von Gremien und Begutachtungsprozessen durchlaufen muss [45]. Untersucht man die Entwicklung von LOM ausgehend von der Veröffentlichung des ersten Entwurfs im Januar 1999, hin zur Freigabe des aktuellen Working Drafts 6 (WD6), so kann man feststellen, daß nahezu keine Änderungen an der Struktur des Datenmodells vorgenommen wurden. Eine LOM konforme DTD, die im Rahmen der vorliegenden Arbeit auf Basis des WD6 entwickelt wurde, ist in Anhang A gezeigt. Mit Hilfe dieser DTD ist es möglich austauschbare und wiederverwendbare multimediale Lernressourcen zu beschreiben. Ein Beispiel für eine mit Hilfe dieser DTD beschriebenen LOM-Ressource ist in Abbildung 14 gezeigt.

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE RECORD SYSTEM "http://www.multibook.de/xml/lom.dtd"><RECORD xmlns="http://www.mult-
book.de/metadata/">
<RECORD><GENERAL>
<TITLE><LANGSTRING LANG="de">Ethernet Kurs</LANGSTRING>
<LANGSTRING LANG="en">Ethernet Course</LANGSTRING>
</TITLE>
<CATALOGENTRY>
<CATALOGUE>KOM</CATALOGUE>
<ENTRY>
<LANGSTRING LANG="de">Network/Ethernet</LANGSTRING>
</ENTRY>
</CATALOGENTRY>
<LANGUAGE>de</LANGUAGE>
<DESCRIPTION>
<LANGSTRING LANG="en">The course describes the functionality of the CSMACD protocol used for
the Ethernet.Problems (and solutions) like Collision of Frames and the Short Frame Problem are
visualised with an interactive Java applet. At the end of the course, the learner can play
around with the Applet and build own scenarios</LANGSTRING>
<LANGSTRING LANG="de">Der Kurs beschreibt die Funktionalität des CSMA-CD Zugriffsprotokoll
beim Ethernet. Probleme (und Lösungen) wie die Gefahr der Kollision von Rahmen oder das "Short
Frame Problem" werden mit Hilfe eines Java-Applets visualisiert. Am Ende des Kurses hat der
Lerner die Möglichkeit selbst mit dem Applet zu experimentieren
und eigene Szenarien zu erstellen
</LANGSTRING>
</DESCRIPTION>
<KEYWORDS><LANGSTRING LANG="en">Ethernet</LANGSTRING></KEYWORDS>
<KEYWORDS><LANGSTRING LANG="en">CSMACD</LANGSTRING></KEYWORDS>
<KEYWORDS><LANGSTRING LANG="en">Kollision</LANGSTRING></KEYWORDS>
<KEYWORDS><LANGSTRING LANG="en">Short Frame Problem</LANGSTRING></KEYWORDS>
<KEYWORDS><LANGSTRING LANG="en">padding</LANGSTRING></KEYWORDS>
<STRUCTURE>
<VOCABULARY><VOCENTRY>2</VOCENTRY></VOCABULARY>
</STRUCTURE>
<AGGREGATIONLEVEL>
<VOCABULARY><VOCENTRY>3</VOCENTRY></VOCABULARY>
</AGGREGATIONLEVEL>
</GENERAL>
<METAMETADATA>
<CONTRIBUTE>
<ROLE>
<VOCABULARY><VOCENTRY>3</VOCENTRY></VOCABULARY>
</ROLE>
<CENTITY>
<VCARD>
BEGIN:VCARD
FN:Achim Steinacker
</VCARD>
</CENTITY>
<DATE>
<DATETIME>988290797864</DATETIME>
</DATE>
</CONTRIBUTE>
<METADATAScheme>IEEELOM:1.0</METADATAScheme>
<LANGUAGE>en</LANGUAGE>
</METAMETADATA>
<LIFECYCLE>
<VERSION><LANGSTRING LANG="de">Version 1</LANGSTRING>
<LANGSTRING LANG="en">1.0</LANGSTRING>
</VERSION>
<STATUS>
<VOCABULARY>
<VOCENTRY>1</VOCENTRY>
</VOCABULARY>
</STATUS>
<CONTRIBUTE>
</CONTRIBUTE>
</LIFECYCLE>

```

```

<TECHNICAL>
<FORMAT><LANGSTRING LANG="de">text:html</LANGSTRING></FORMAT>
<FORMAT><LANGSTRING LANG="de">application/x-java-Applet</LANGSTRING></FORMAT>
<SIZE>1032353</SIZE>
<LOCATION>http://www.kom.e-technik.tu-darmstadt.de/~abed/applets/paradelektion</LOCATION>
<REQUIREMENTS>
<TYPE><VOCABULARY><VOCENTRY>4</VOCENTRY></VOCABULARY></TYPE>
<NAME><VOCABULARY><VOCENTRY>101</VOCENTRY></VOCABULARY></NAME>
<MINIMUMVERSION>4.0</MINIMUMVERSION>
<MAXIMUMVERSION></MAXIMUMVERSION>
</REQUIREMENTS>
<REQUIREMENTS>
<TYPE><VOCABULARY><VOCENTRY>4</VOCENTRY></VOCABULARY></TYPE>
<NAME><VOCABULARY><VOCENTRY>102</VOCENTRY></VOCABULARY></NAME>
</REQUIREMENTS>
<INSTALLATIONREMARKS>
<LANGSTRING LANG="de">keine Installation noetig</LANGSTRING>
<LANGSTRING LANG="en">no installation necessary</LANGSTRING>
</INSTALLATIONREMARKS>
<OTHERPLATFORMREQUIREMENTS>
<LANGSTRING LANG="de">Java faehiger WebBrowser wird benötigt</LANGSTRING>
<LANGSTRING LANG="en">Browser must support Java applets</LANGSTRING>
</OTHERPLATFORMREQUIREMENTS>
</TECHNICAL>
<EDUCATIONAL>
<INTERACTIVITYTYPE><VOCABULARY><VOCENTRY>3</VOCENTRY></VOCABULARY></INTERACTIVITYTYPE>
<LEARNINGRESOURCETYPE><VOCABULARY><VOCENTRY>4</VOCENTRY></VOCABULARY></LEARNINGRESOURCETYPE>
<LEARNINGRESOURCETYPE><VOCABULARY><VOCENTRY>14</VOCENTRY></VOCABULARY></LEARNINGRESOURCETYPE>
<INTERACTIVITYLEVEL><VOCABULARY><VOCENTRY>2</VOCENTRY></VOCABULARY></INTERACTIVITYLEVEL>
<SEMANTICDENSITY><VOCABULARY><VOCENTRY>4</VOCENTRY></VOCABULARY></SEMANTICDENSITY>
<INTENDEDENDUSERROLE><VOCABULARY><VOCENTRY>2</VOCENTRY></VOCABULARY></INTENDEDENDUSERROLE>
<CONTEXT><VOCABULARY><VOCENTRY>7</VOCENTRY></VOCABULARY></CONTEXT>
<DIFFICULTY><VOCABULARY><VOCENTRY>2</VOCENTRY></VOCABULARY></DIFFICULTY>
<DESCRIPTION>
<LANGSTRING LANG="de">kann zum Selbststudium verwendet werden</LANGSTRING>
</DESCRIPTION>
<LANGUAGE>de</LANGUAGE>
</EDUCATIONAL>
<RIGHTS>
<COST><VOCABULARY><VOCENTRY>1</VOCENTRY></VOCABULARY></COST>
<COPYRIGHTOROTHERRESTRICTIONS>
<VOCABULARY><VOCENTRY>0</VOCENTRY></VOCABULARY>
</COPYRIGHTOROTHERRESTRICTIONS>
<DESCRIPTION>
<LANGSTRING LANG="de">Copyright 2000 by KOM, TU Darmstadt</LANGSTRING>
<LANGSTRING LANG="en">Copyright KOM, TU Darmstadt</LANGSTRING>
</DESCRIPTION>
<CLASSIFICATION>
<PURPOSE><VOCABULARY><VOCENTRY>3</VOCENTRY></VOCABULARY></PURPOSE>
<TAXONPATH>
<SOURCE>ACM-Classification</SOURCE>
<TAXON>
<ENTRY><LANGSTRING LANG="de">Computer Systems Organization</LANGSTRING></ENTRY>
</TAXON>
<TAXON>
<ENTRY><LANGSTRING LANG="de">COMPUTER-COMMUNICATION NETWORKS</LANGSTRING></ENTRY>
</TAXON>
<TAXON>
<ENTRY><LANGSTRING LANG="de">Local and Wide-Area Networks </LANGSTRING></ENTRY>
</TAXON>
<TAXON>
<ENTRY><LANGSTRING LANG="de">Ethernet (e.g., CSMAD)</LANGSTRING></ENTRY>
</TAXON>
/CLASSIFICATION>

```

Abbildung 14: Beispiel für eine LOM-Beschreibung mit XML

3.4 Dynamische Medienbausteine

Mit Hilfe einer LOM-Beschreibung lässt sich sowohl der Inhalt einer multimedialen Lernressource beschreiben, als auch die Voraussetzungen, die benötigt werden, um die Ressource präsentieren zu können. Die technischen Voraussetzungen sind gerade im Lernbereich von großer Bedeutung, da in diesem Bereich das Wissen sehr oft in Form von interaktiven multimedialen Animationen und Simulationen vermittelt wird [109]. Diese multimedialen Bausteine sind in der Regel aber nicht direkt in einem Web-Browser anzeigbar, sodass spezielle Viewer dafür verwendet werden müssen. LOM bietet hierfür die Technical-Kategorie, in der genau spezifiziert werden kann, welche technischen Voraussetzungen nötig sind, um mit einer Ressource zu arbeiten. Animationen oder Simulationen zeichnen sich häufig allerdings zudem dadurch aus, daß sie direkt auf Benutzereingaben reagieren können, und sich aufgrund dieser Eingaben unterschiedlich verhalten. Ein Beispiel für eine interaktive Ressource zeigt Abbildung 15.

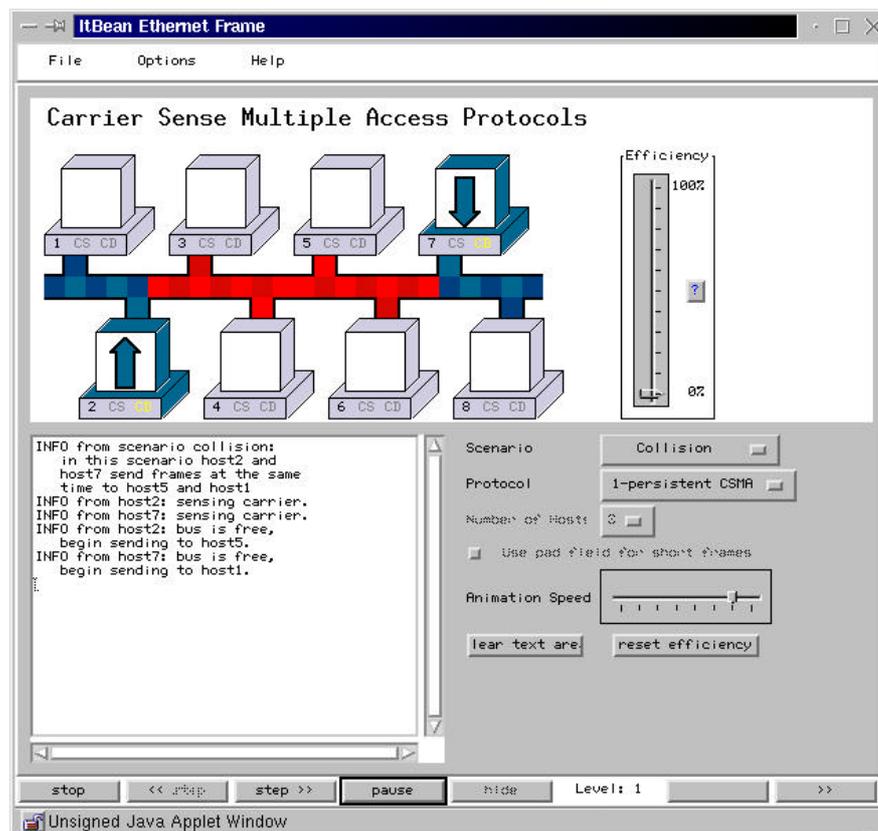


Abbildung 15: Beispiel für einen dynamischen Medienbaustein

Die Abbildung zeigt ein Java-Applet, das das CSMA/CD-Zugriffsverfahren beim Ethernet simuliert. Die entsprechende LOM-Beschreibung dieses Applets wurde in Abbildung 14 gezeigt. Dieses Applet verfügt über eine Reihe von Parametern mit denen das Aussehen und das Verhalten der Animation beeinflusst werden können. So lässt sich bsp. über einen Startparameter die Sprache einstellen, die während der Laufzeit innerhalb des Applets verwendet

wird. Zudem läßt sich über Startparameter bestimmen, welche Szenarien des Zugriffs bei der Simulation gezeigt werden sollen. Das Applet kann daher bei Bedarf detailliertere Probleme (und Lösungen) wie die Gefahr der Rahmenkollision (Collision Detection) beim gleichzeitigen Zugriff mehrerer Rechner auf das Übertragungsmedium oder das Problem der kurzen Rahmen (Short Frame Problem) präsentieren. Soll nur ein erster Überblick über die Funktionsweise des CSMA/CD-Protokolls bei der Simulation gezeigt werden, so können diese weiterführenden Techniken, durch entsprechende Wahl der Startparameter auch unterdrückt werden. Die Simulation kann somit an direkt an die Bedürfnisse einzelner Lerner angepaßt werden und je nach Bedarf unterschiedliche Detaillierungsgrade präsentieren. Zusätzlich kann das Applet als Testmodul verwendet werden. In diesem Fall werden die eben beschriebenen Szenarien wie das Short Frame-Problem ohne die dazugehörige Lösung des Auffüllens der Rahmen präsentiert. Stattdessen werden dem Lerner nach der Präsentation einer Kollision verschiedene Lösungsmöglichkeiten angeboten, aus denen eine ausgewählt muß.

Das Beispiel des Ethernet-Applets zeigt die Interaktionsmöglichkeiten und unterschiedlichen Verhaltensweisen von Medienbausteinen. Dabei beschränkt sich wie erläutert das dynamische Verhalten nicht nur auf den Zeitraum, während mit der Ressource gearbeitet wird, sondern kann in drei Abschnitte unterteilt werden.

1 Startzeitpunkt

Entsprechend der Startparameter, kann die Art der Präsentation an bestimmte Benutzerpräferenzen oder andere externe Restriktionen angepaßt werden. Ein Beispiel für eine Anpassung an den Benutzer ist, die Möglichkeit die Sprache einzustellen, die während der Präsentation verwendet wird. Externe Restriktionen, an die eine Anpassung vorgenommen werden kann ist bsp. der zur Verfügung stehende Platz auf der Visualisierungsoberfläche oder das von anderen Bausteinen verwendete Layout.

1 Zeitraum, während mit der Ressource gearbeitet wird

Bausteine, die eine direkte Interaktion mit dem Benutzer anbieten, können Entscheidungen des Benutzers, während der Arbeit mit der Ressource an eine Steuerkomponente übergeben, die die Entscheidungen auswertet und in beim späteren Verhalten des Systems gegenüber dem Benutzer berücksichtigen kann.

1 Ende der Bearbeitungszeit

Handelt es sich bei einem Medienbaustein um einen Test, so kann der Medienbaustein das Ergebnis des Tests oder die Dauer der Bearbeitungszeit als Information für eine weitere Auswertung zur Verfügung stellen. Steht keine Auswertungsmöglichkeit zur Verfügung, so muß der Baustein die Auswertung des Tests selbst vornehmen. In diesem Fall kann das Ergebnis eines Tests dann nicht dafür verwendet werden, dem Lerner je nach Ergebnis zusätzliche Medienbausteine anzubieten.

Untersucht man die verschiedenen LOM-Kategorien, so kann man feststellen, daß es mit LOM leider bisher nicht möglich ist, das dynamische Verhalten interaktiver Medienbausteine vollständig zu beschreiben. Einzig die Educational-Kategorie von LOM bietet mit den Metadaten-elementen *InteractivityType* und *InteractivityLevel* die Möglichkeit anzugeben ob eine Ressource interaktive Elemente enthält; es stehen jedoch keine Elemente zur Verfügung mit denen man angeben kann, wie sich eine Ressource an den Benutzer anpassen oder mit welchen Parametern eine Ressource gestartet werden kann. Dabei reicht eine einfache Erweiterung von LOM um Elemente mit denen bsp. die Startparameter beschrieben werden können nicht aus. Zusätzlich muß auch noch eine standardisierte Schnittstelle verwendet werden, die spezifiziert, wie die Parameter beim Start übergeben werden, bzw. die Resultate nach Beendigung abge-

fragt werden können.

3.4.1 Parametrisierbare Medienbausteine

Da der LOM-Entwurf eine Erweiterung der einzelnen Kategorie oder auch die Definition neuer Kategorien erlaubt, ist es auf einfache Art und Weise möglich, Metadatenelemente in eine LOM-Beschreibung mitaufzunehmen, die das dynamische Verhalten eines Medienbausteins steuern. Das Problem bei diesem Ansatz besteht allerdings darin, daß es sich hierbei um eine proprietäre Erweiterung handelt, die dann nur von Programmen verwendet werden kann, die diese Erweiterung implementieren. Eine andere Möglichkeit das dynamische Verhalten von Medienbaustein zu beschreiben die durch den LOM-Entwurf ebenfalls abgedeckt ist besteht darin, für jedes mögliche Verhalten des Medienbausteins eine eigene Metadatenbeschreibung zu erstellen. Als Verweis auf die eigentliche Ressource kann dann bei jeder Metadatenbeschreibung eine eigene HTML-Seite verwendet werden, die beim Start der Animation die entsprechenden Parameter an den Baustein übergibt. Dieses Verfahren hat den Vorteil, daß keine Erweiterung von LOM erforderlich ist und die dynamischen Eigenschaften in allen Lernsystemen, die standardkonforme LOM-Beschreibungen benutzen, verwendet werden kann. Der Nachteil dieses Verfahren, besteht darin, daß der Verwaltungsaufwand für die Metadatenbeschreibungen enorm anwächst. Wird ein Medienbaustein aus einer Wissensbasis gelöscht, so muß nicht nur eine Metadatenbeschreibung entfernt werden, sondern jede einzelne Variante.

Ein erster Ansatz für eine Erweiterung von LOM um Elemente mit denen das dynamische Verhalten von Medienbausteinen definiert werden kann ist in [110] beschrieben. Die Autoren erweitern LOM um eine Kategorie *Dynamic Metadata*, die folgende Elemente enthält:

Nummer	Eigenschaft
1	CodeInformation
1.1	CodeName
1.2	CodeLocation
1.3	CodePackage
2	PresentationInformation
2.1	Language
2.2	DifficultyLevel
2.3	InteractivityLevel
2.4	Dimension
2.5	Bidirectional
3	TopicInformation
3.1	Topic

Tabelle 2 Kategorie Dynamic Metadata für LOM (vgl. [110])

Nummer	Eigenschaft
3.2	Scenario
3.2.1.	Mode
3.2.2	Name
3.3	InputData
3.4	OutputData
4	ExplanationInformation
4.1	Type
4.2	Media

Tabelle 2 Kategorie Dynamic Metadata für LOM (vgl. [110])

Untersucht man die vorgeschlagenen Kategorien, so kann man feststellen, daß sie stark auf bestimmte Eigenschaften von Ressourcen ausgerichtet sind. Während alle Elemente der Unterkategorie *CodeInformation* für alle dynamischen Varianten eines Medienbausteins unverändert bleiben und daher die Technical-Kategorie die geeignetere Kategorie zur Speicherung dieser Informationen ist, beschreiben die Elemente der Unterkategorie *PresentationInformation* spezielle Eigenschaften, die nicht für beliebige Animationen oder Simulationen gelten. Durch die Elemente *InputData* und *OutputData* steht eine Möglichkeit zur Verfügung individuelle Startparameter für einzelne Medienbausteine zu spezifizieren. Über diese Elemente können dann auch Werte wie die Sprache die innerhalb des Bausteins verwendet werden soll oder der Interaktionsgrad der Ressource direkt beim Start an den Medienbaustein übergeben werden.

3.4.2 Schnittstelle für Medienbausteine

Die Advanced Distributed Learning Initiative (ADL)¹ des US-Verteidigungsministeriums hat mit dem Shareable Courseware Object Reference Model (SCORM) ein Referenzmodell zur Beschreibung von Kursobjekten entwickelt. Unter Kursobjekten werden dabei im SCORM-Modell kombinierbare multimediale Lerneinheiten verstanden, die als adressierbare Inhaltseinheiten (Assignable Content Unit) bezeichnet werden und zu vollständigen Kursen zusammengesetzt werden können. Sie entsprechen damit dem in der vorliegenden Arbeit verwendeten Begriff der Medienbausteine.

Zur Beschreibung der einzelnen Medienbausteine empfiehlt die ADL die Verwendung von LOM. Zur Beschreibung von kompletten Kurs wurde ein eigenes Modell entwickelt. Zusätzlich zu diesem Modell hat die ADL eine API entwickelt, die zur Kommunikation zwischen Medienbausteinen und einer Steuerungskomponente eingesetzt werden kann (vgl. Abbildung 16)

1. URL: <http://www.adlnet.org>

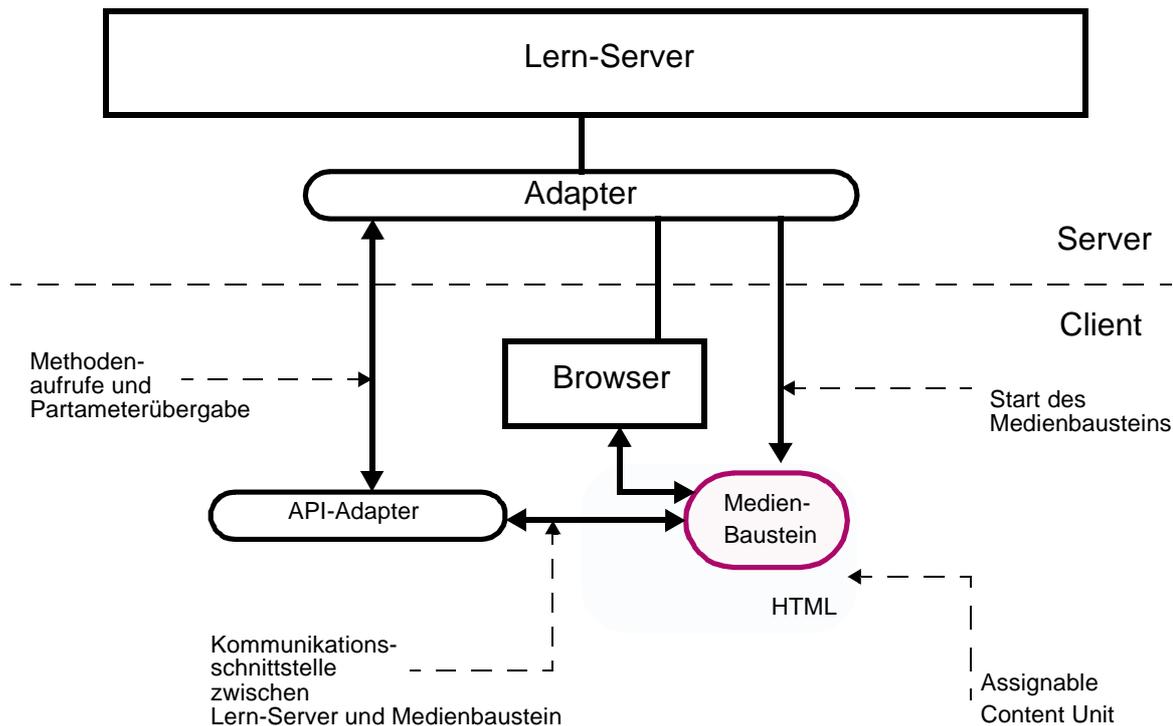


Abbildung 16: SCORM API (vgl. [2])

Im Gegensatz zum LOM-Entwurf, der ausdrücklich die Beschreibung nicht digitaler Ressourcen erlaubt, ist das SCORM-Modell auf Ressourcen beschränkt, die mit einem Web-Browser (erweitert um PlugIns) präsentiert werden können. Diese Einschränkung hat den Vorteil, daß nicht wie bei LOM generische Schnittstellen entwickelt werden müssen, um alle möglichen Anwendungsfälle abdecken zu können, sondern daß die Datenmodelle und Schnittstellen direkt an die Gegebenheiten und Programmiersprachen im Web angepaßt werden können.

Das SCORM-Modell spezifiziert zur Kommunikation zwischen dem Lernsystem (Server) und dem Web-Browser (Client) auf jeder Seite einen Adapter, der auf Client-Seite in Form eines in Java spezifizierten Objekts realisiert ist. Nach Übergabe der Startparameter an den Adapter des Clients wird der Start der Präsentation des Medienbausteins vom Lernsystem initiiert. Die u.U. zum Start benötigten Parameter werden dabei vom Client-Adapter an den Medienbaustein übergeben, sodass zwischen dem Medienbaustein und dem Lernsystem keinerlei Abhängigkeiten bestehen.

Die Schnittstelle zwischen Client-Adapter und Medienbaustein ist in ECMAScript realisiert¹. Da sowohl alle gängigen Browser eine JavaScript Unterstützung bieten, als auch praktisch alle Plug-Ins über eine JavaScript Schnittstelle verfügen, kann diese Schnittstelle ohne weitere

1. Bei ECMAScript handelt es sich um eine von der European Computer Manufacturers Association (ECMA) 1997 standardisierte Form von JavaScript. JavaScript wurde von Netscape Communications erstmals 1995 in der Version 2.0 der Browser Software Netscape Navigator verwendet.

Installation direkt beim Client im Browser verwendet werden. Im Gegensatz zu der expliziten Definition der Parameter in Abschnitt 3.4.1 "Parametrisierbare Medienbausteine", verwendet SORM eine generische Schnittstelle über die eine beliebige Anzahl von Parametern übergeben werden kann. Die Methoden der Schnittstelle lassen sich in drei Kategorien unterteilen (vgl. [2])

1 Status

Zur Abfrage des Status eines Medienbausteins stehen die Funktionen *initialize()* und *finish* zur Verfügung. Diese Methoden müssen von den Medienbausteinen oder von einem sie umgebenden Wrapper direkt nach dem Start bzw. Beendigung aufgerufen werden, um das Lernsystem über den Status des Medienbausteins zu informieren.

1 Fehlerbehandlung

Treten während der Arbeit mit einem Medienbaustein Fehler auf, stehen die Methoden *LMSGetLastError()*, *LMSGetErrorString()*, *LMSDiagnostic()* und *LMSCommit()* zur Verfügung um den Lernsystem Informationen über die Art des aufgetretenen Fehlers zu informieren, bzw. Anweisungen des Lernsystems zur Fehlerbehandlung entgegenzunehmen.

1 Datentransfer

Zur Übertragung von Daten zu und von einem Medienbaustein bietet die Schnittstelle die Methoden *LMSGetValue()* und *LMSSetValue()* an. Als Datenmodell für die Parameter wird dabei ein generisches Modell verwendet, das keinerlei Einschränkungen hinsichtlich der Semantik der übertragenen Parameter macht. Entsprechend können über diese Schnittstelle beliebige applikationsspezifische Daten übertragen werden. Da die Parameter in der Metadatenbeschreibung des Medienbausteins gespeichert werden und dort auch die technischen Spezifikationen des Bausteins verfügbar sind, kann sichergestellt werden, daß nur solche Parameter zum Baustein übertragen werden, die von dem Baustein interpretiert und verarbeitet werden können.

3.5 Automatische Komposition von Medienbausteinen

Die bisherigen Untersuchungen von LOM bezogen sich auf die Möglichkeiten anhand der Metadatenbeschreibung eine Ressource zu finden und unabhängig vom Kontext indem die Ressource verwendet werden soll, zu beurteilen. Eines der Ziele die mit der Beschreibung von LOM erreicht werden soll ist aber die Möglichkeit einzelne Module quasi automatisch zu größeren Einheiten zu kombinieren (vgl. Abschnitt 3.1.2). In diesem Fall sind also nicht mehr nur die Eigenschaften interessant, die ein einzelner Medienbaustein aufweist, sondern vor allem das Verhältnis, das ein einzelner Medienbaustein zu anderen Bausteinen besitzt.

Das Hauptproblem bei der Generierung von Kursen aus abgeschlossenen und unabhängigen Modulen ist nach [117] die Kohärenz der generierten Einheit und eine konsistente Präsentation des Kurses in Bezug auf den Inhalt und die Form der Präsentation. Automatisch generierte Kurse finden demnach nur dann Akzeptanz beim Lerner, wenn die Struktur des Kurses für den Lerner nachvollziehbar ist. Zudem muß der Kurs aus Medienbausteinen bestehen, die eine ähnliche Zielgruppe haben, sodas die Inhalte konsistent vermittelt werden.

3.5.1 Eigenschaften einzelner Medienbausteine

Die Zusammenstellung von Medienbausteinen zu individuellen Kursen schränkt die Auswahl von Medienbausteinen auf digitale Ressourcen ein. Dies bedeutet aber nicht, daß der Kurs selbst auch nur digital präsentiert werden kann. So ist es ohne weiteres denkbar, daß Medienbausteine, bei denen die Informationen im PDF-, oder in einem Grafik-Format kodiert sind, zu einem Kurs zusammengestellt werden, der Kurs selbst dann aber ausgedruckt zur Verfügung gestellt wird.

Wird der Kurs digital präsentiert, so hängt die Auswahl an möglichen Bausteinen unabhängig von den vermittelten Inhalten davon ab, welche Formate von der Präsentationssoftware akzeptiert werden. Da in der Technical-Kategorie einer LOM-Beschreibung detaillierte Informationen über das physikalische Format der Ressource und die Voraussetzungen für eine Präsentation des Bausteins gespeichert werden können, ist ein Auswahl nach technischen Kriterien problemlos möglich.

Deutlich schwieriger gestaltet sich die Suche nach "passenden" Bausteinen aufgrund von inhaltlichen Kriterien. Ein Problem besteht darin, daß verschiedene Autoren einer Metadatenbeschreibung unterschiedliche Vorstellungen über die Werte bestimmter Metadatenelemente haben, selbst wenn für das Element ein festes Vokabular zur Auswahl steht. Bei der automatischen Generierung von Kursen können aber nur genau diese Felder zur Beurteilung, ob zwei Medienbausteine kombiniert werden können, heranziehen. So wird ein Algorithmus beispielsweise dazu tendieren Medienbausteine auszuwählen, die bei dem Metadatenelement "Semantische Dichte" den gleichen Eintrag haben. Wenn die Autoren bei der Erstellung der jeweiligen Metadatenbeschreibung diesem Wert eine unterschiedliche Bedeutung zugemessen haben, lassen sich aber durch eine Gleichheit der Werte keinerlei Rückschlüsse ziehen, ob diese beiden Medienbausteine zu einer größeren Einheit kombiniert werden können.

Die Frage, ob dieses Problem durch andere Metadatenelemente oder andere Vokabulare gelöst werden kann wird von Seeberg in [119] verneint. Belegt wird die These unter anderem anhand eines einfachen Beispiels:

Bei einer Evaluation wurden einer Testgruppe zwei unterschiedliche Formulierungen der Fermatschen Vermutung präsentiert. Eine Version bestand aus einer formalen Beschreibung der Vermutung; die andere verwendete eine natürlichsprachliche Beschreibung der These und verwendete keine anderen mathematischen Symbole außer "+" und "=". Anzugeben waren von den Testpersonen jeweils welche der beiden Erläuterungen ihnen leichter verständlich erschien und ob sie über mathematisches Hintergrundwissen verfügen. Aus der Evaluation lässt sich der Schluss ziehen, daß keine allgemein gültige Aussage getroffen werden kann, welche der beiden Versionen leichter oder schwerer verständlich ist. Objektive Aussage über die pädagogische Eigenschaften eines Medienbausteins können daher nicht getroffen werden, wenn keine Informationen vorliegen, in welchem Kontext ein Medienbaustein verwendet werden soll, oder welche Kriterien bezüglich des mathematischen Hintergrundwissens der Autor der Metadatenbeschreibung bei der Annotierung der Ressource angewendet hat. Das Beispiel zeigt auch, daß auch durch weitere Metadatenelemente keine bessere Bewertung über die Eignung eines Medienbausteins für einen Kurs möglich ist, solange sich die Elemente nur auf den Medienbaustein selbst beziehen.

3.5.2 Beziehungen zwischen Medienbausteinen

Neben den Metadatenelmenten von LOM, die ausschließlich Eigenschaften eines einzelnen Medienbausteins beschreiben, können durch die Relational-Kategorie Beziehungen zwischen Medienbausteinen modelliert werden. Diese Beziehungen können verwendet werden, um während der Suche und Auswahl von Medienbausteinen, Methoden des Instruktionsdesign zu berücksichtigen. Ein Algorithmus könnte beispielsweise zu Beginn einen Medienbaustein auswählen, der ein Problem beschreibt. Über Beziehungen, die dieser Medienbausteinen zu anderen Bausteinen besitzt, können danach Medienbausteine ausgewählt werden, die das Problem illustrieren, oder eine Lösung des Problems beinhalten. Fraglich ist also, ob mit der Relational-Kategorie und vor allem mit dem vorgeschlagenen Vokabular, solche Zusammenhänge zwischen Medienbausteinen modelliert werden können. LOM spezifiziert als mögliche Werte für die Beziehungen kein eigenes Vokabular, sondern übernimmt die Vorschläge von Dublin Core. Die Werte sind im einzelnen:

```
{ isPartOf, HasPart, IsVersionOf, HasVersion, IsFormatOf, HasFormat, References, IsReferencedBy, IsBasedOn, IsBasisFor, Requires, IsRequiredBy }
```

Man erkennt daß zu jeder möglichen Relation im Vokabular eine Umkehrrelation vorgesehen ist, sodas die Anzahl der möglichen Einträge eigentlich nur aus 6 Werten besteht, deren bibliographischer Hintergrund deutlich zu erkennen ist. Zudem vermischen die Werte inhaltliche und semantische Zusammenhänge zwischen zwei Medienbausteinen. Die Tatsache, daß ein Medienbaustein einen anderen referenziert, ist ein Indiz dafür, daß beide Bausteine Informationen über dasselbe Thema beinhalten; man kann daraus aber nicht ablesen, ob die Bausteine zusammen in einer bestimmten Reihenfolge präsentiert werden können. Problematisch sind auch die Relationen *IsPartOf/HasPart* und *isVersionOf/HasVersion*. Diese Relationen werden benötigt um zu verwalten, welche Medienbausteine zu einem Kurs gehören oder in verschiedenen Versionen vorliegen. Handelt es sich bei einem Medienbaustein der eine *HasPart*-Relation zu einem anderen Medienbaustein besitzt um eine physikalische Ressource und nicht um eine Strukturbeschreibung, dann verletzt die Relation zudem, die Voraussetzung der unabhängigen Wiederverwendung eines Bausteins. Das gleiche Problem besteht bei Bausteinen, die durch die *Requires/IsRequiredBy*-Relation verbunden sind. Hängt die Verwendung eines Bausteins vollständig von der Existenz eines anderen Bausteins ab, so kann er nicht unabhängig von dem anderen Baustein verwendet werden. Der Ansatz der Wiederverwendung geht damit komplett verloren.

Bei Verwendung der *IsBasedOn/IsBasisFor*-Relation verhält es sich ähnlich. Will man mit dieser Relation einen inhaltsbasierten Zusammenhang zwischen den Bausteinen ausdrücken, besteht kein Unterschied zwischen der *IsBasedOn*- und der *Requries*-Relation. Soll über diese Beziehung ein semantischer Zusammenhang der Art, daß ein Medienbaustein Informationen über einen Begriff beinhaltet, der in einem anderen Medienbaustein erläutert ist, ausgedrückt werden, so sollte diese Aussage nicht durch das Verbinden zweier Medienbausteine in der

Wissensbasis repräsentiert werden. Durch eine semantische Beziehung zwischen zwei Begriffen, wird eine allgemein gültige Aussage über einen bestimmten Teil eines Wissensgebietes getroffen, die auch noch gültig ist, wenn man einen Medienbaustein aus der Wissensbasis wieder entfernt, sich dann aber nicht mehr aus ihr ableiten läßt.

3.6 Bewertung von LOM

Wie in der Einleitung dieses Abschnitts erwähnt wurde, wird LOM derzeit von nahezu allen Organisationen oder Forschungsprojekten, die sich mit der Archivierung von elektronischen Lernmaterialien beschäftigen, als geeignetes Metadatenschema angesehen. Praktische Erfahrungen mit LOM existieren hingegen bisher nicht.

3.6.1 Anzahl der Metadatenelemente

Ein Problem des LOM-Entwurfs ist die in der Einleitung dieser Arbeit genannte Unklarheit, welche Eigenschaften einer Lernressource beim Einsatz in einem multimedialen Lernsystem zu berücksichtigen sind. Aus diesem Grund versucht der LOM-Entwurf möglichst alle Einsatzbereiche zumindest ansatzweise abzudecken. Die Beschreibung einer Ressource kann damit sehr groß werden. Die Menge an Informationen, die bei der Verwendung von Metadaten zur Beschreibung einer Ressource nötig sind, und der Aufwand die Metadatenbeschreibungen zu erstellen, wird oftmals als Argument gegen die Verbreitung von Metadatenschemata verwendet [81] [WG12]. Argumentiert wird dabei, daß der Autor einer Ressource nicht bereit ist, den Aufwand für die Beschreibung zu leisten, da sich aus den Beschreibungen kein zusätzlicher Nutzen für ihn ergeben würde. Belegt wird dieses Argument beispielsweise mit dem Sachverhalt, daß von der seit HTML 3.2 möglichen Verwendung von Meta-Tags in der Praxis kaum jemand Gebrauch macht. Gegen diese Argumentation lassen sich folgende Punkte anführen:

- 1 Anwachsen der Informationsmenge
Wie in Abschnitt 2.1 "Einleitung" schon erwähnt wurde, vollzieht sich derzeit ein Übergang von dem Problem Informationen verfügbar zu machen, zu der Frage, wie die gewünschte Information unter allen auf einen Mausklick zugänglichen Informationen überhaupt noch zu identifizieren ist. Die Notwendigkeit Ressourcen mit Metadaten zu beschreiben ist daher nicht auf Lernmaterial beschränkt sondern gilt zukünftig für alle web-basierten Dokumente.
- 1 Nutzen für Autoren bei Einsatz der Ressource in einem kommerziellen Szenario.
Metadatenbeschreibungen dienen dazu, daß Ressourcen gefunden werden und die identifizierten Ressourcen in vorhandene Materialien eingebunden werden können. Werden die Ressourcen in einem kommerziellen Szenario eingesetzt, bei der der Autor nicht nur für die einmalige Erstellung, sondern für jede Verwendung eine Aufwandsentschädigung erhält, so steigt das Interesse der Autoren ihr Material so zu beschreiben, daß es von potentiellen Abnehmern gefunden werden kann automatisch.
- 1 Automatische Generierbarkeit von Einträgen
Wie in Abschnitt 3.2 gezeigt wurde, lassen sich circa 50% aller möglichen Einträge durch direkte Analyse der Ressource oder durch Verwendung von Schablonen für wiederkehrende Werte automatisch ermitteln und müssen daher nicht mehr manuell erstellt

werden.

3.6.2 Austauschbarkeit und Werkzeuge zur Erstellung und Suche

Die meisten der bisherigen Lernsysteme verwalten die Metadatenbeschreibungen in einem proprietären Format (vgl. die Beschreibung von AHS zu Anfang des Kapitels). Dies bedeutet, daß ein Einfügen von externen Metadatenbeschreibungen praktisch nicht möglich ist. Zudem ist die Menge an beschriebenen Ressourcen, die von bestehenden Systemen verwaltet werden zu klein, um für die Erstellung komfortable Werkzeuge zu entwickeln. Betrachtet man den heutigen Stand, so kann gesagt werden, daß nur für Dublin Core erste Werkzeuge zur Beschreibung existieren, für andere Metadatenschema bisher keine. Mit der Verbreitung eines einheitlichen Beschreibungsformat wie LOM und der in Abschnitt 3.3.2 "Document Type Definition für LOM" vorgestellten DTD zur Kodierung von Metadatenbeschreibungen, können jetzt Werkzeuge entwickelt werden, die eine komfortable Auszeichnung von Lernressourcen ermöglichen. In Abschnitt 4 "Autoren- und Suchwerkzeuge für LOM" ist die Entwicklung eines solchen Werkzeugs, der LOM-Editor, beschrieben.

Stehen LOM-Beschreibungen in XML kodiert zur Verfügung wird nicht nur die Austauschbarkeit der Beschreibungen gewährleistet, sondern gleichzeitig festgelegt, wie eine Suche auf LOM-Beschreibungen ablaufen kann. Grund hierfür ist die vom W3C derzeit entwickelte XQL-Anfragesprache. (vgl. Abschnitt 4.2.1). Mit dieser auf XML aufbauenden allgemeinen Anfragesprache kann direkt in XML-Strukturen gesucht werden. Eine auf XQL aufbauende web-basierte Suchschnittstelle wird in Abschnitt 4.2 beschrieben.

3.6.3 Vollständigkeit von LOM

Obwohl LOM 9 Kategorien enthält und eine LOM-Beschreibung, wie im letzten Abschnitt erwähnt, sehr viele Einträge enthalten kann stehen für den speziellen Einzelfall dennoch nicht genügend Informationen zur Verfügung um die Eigenschaften einer Ressource zu spezifizieren. So verfügt eine LOM-Beschreibung zwar über eine Rights-Kategorie in der angegeben werden kann, ob die Benutzung kostenlos ist, oder wer über die Urheberrechte der Ressource verfügt. Für die Verwaltung eines Kurse in einem Lernsystem mit sehr vielen Benutzern, die unterschiedliche Zugriffsrechte auf Kurse haben sollen, oder gar das Handel eines solchen Kurses auf einem kommerziellen Marktplatz reichen die verfügbaren Elemente der Rights-Kategorie aber bei weitem nicht aus, um alle Informationen zur Nutzung zu spezifizieren. Andere Kategorien, wie die in Abschnitt 3.4.1 "Parametrisierbare Medienbausteine" beschriebene Möglichkeit dynamische Medienbausteine zu beschreiben, oder die Möglichkeit die Lay-outeigenschaften eines Bausteins anzugeben fehlen völlig.

3.7 Semantische Netze

Die Analyse von LOM in Abschnitt 3.5.2 "Beziehungen zwischen Medienbausteinen" zeigt, daß Metadaten, die sich auf eine konkrete physikalische Ressource beziehen, nicht geeignet sind, die Informationen, die in den Ressourcen beschreiben sind, in einen semantischen Zusammenhang zu bringen. Zwar lassen sich durch Auswahl eines anderen Vokabulars für die

Spezifikation der Beziehung zwischen Metadatenbeschreibungen, semantische Beziehungen zwischen Ressourcen und damit auch zwischen dem Wissen, das durch die Ressourcen vermittelt wird, ausdrücken. Wie in Abschnitt 3.5.2 beschreiben, ist die Modellierung dieses Wissens aber direkt von der Existenz einer Metadatenbeschreibung abhängig, obwohl es auch unabhängig von einer konkreten physikalischen Ressource existiert und für Entscheidungen verwendet werden kann. Zudem ergibt sich wie ebenfalls in 3.5.2 beschrieben, das Problem der redundanten Modellierung, wenn man nur Metadatenbeschreibungen über physikalische Ressourcen zur Beschreibung einer Wissensbasis verwendet. Aus diesem Grund werden in diesem Abschnitt Verfahren und Modelle zur allgemeinen Wissensrepräsentation beschrieben.

3.7.1 Wissensrepräsentation

Die Forschung auf dem Gebiet der Wissensrepräsentation und Wissensmodellierung zeichnet sich dadurch aus, daß eine Vielzahl von Modellen und unterschiedlichen Begriffsbezeichnungen für die gleichen Sachverhalte existieren. Zudem werden umgekehrt die gleichen Begriffe für unterschiedliche Merkmale verwendet. Im folgenden werden deshalb beginnend mit einer Kategorisierung nach Baumert [98] die einzelnen Modelle kurz vorgestellt.

Yale Schule

Das Modell der Yale-Schule stellt eine logische Sprache zur Verfügung, mit der zunächst konzeptuelle Abhängigkeiten beschrieben werden können. Daraus wurden Programme entwickelt, die aus natürlich-sprachlichen Sätzen automatisch auf Tatsachen schließen können. Weiterhin ist es möglich typische Handlungsabläufe, sogenannte Skripten, darzustellen. Hier wird davon ausgegangen, dass wir bei vielen unserer Handlungen auf frühere Erfahrungen zurückgreifen, um eine Situation zu meistern. Das Skript stellt ein Art Drehbuch dar, das den Handlungsablauf für standardisierte Ereignisse vorgibt. Probleme entstehen dann, wenn unbekannte Situationen entstehen. Für Menschen ist es oft kein Problem, sich Ziele und Pläne zu setzen, auch wenn die Situation neu für ihn ist. Im Modell der Yale-Schule werden für diesen Fall Dämonen eingesetzt, die auf bestimmte Änderungen des Zustandes reagieren und einen Plan für die neue Situation suchen.

Frame Semantics

Im Modell der Frame-Semantics werden die strukturierten Objekte, die Frames, betrachtet. Diese sind mit den obigen Skripten vergleichbar, gehen aber von Objekten und nicht von konzeptuellen Abhängigkeiten aus. Ein Frame repräsentiert eine Situation oder ein Objekt, dessen Eigenschaften in Slots gespeichert sind. Wenn wir etwas sehen, dann wird in unserem Gehirn ein Frame aufgerufen, um das Gesehene einordnen zu können. In einem uns unbekanntem Zimmer erkennen wir nicht auf den ersten Blick alle Details, aber wir wissen dass es vier Wände, einen Fussboden und eine Decke aufweisen sollte. Das sind zumindest die Standardwerte für den Slot eines Zimmers. Sollte sich herausstellen, daß das Zimmer fünf Wände hat, dann wird der entsprechende Slot dieses speziellen Zimmers mit neuen Werten gefüllt. Treffen wir eine Person, die Schuhe anhat, dann wissen wir, daß die Person auch Füße hat, obwohl wir die Füße gar nicht sehen. Die Tatsache, daß die Person Füße hat, ist in diesem Fall ebenfalls als Standardwert in den Slots des Fuß-Frames gespeichert. Standardwerte sind also typische Werte, die für die meisten Frames dieser Art zutreffen, im Einzelfall aber auch abweichen können [81].

Semantische Netze

Das dritte vorgestellte Modell ist das der Semantischen Netze. Dabei handelt es sich wie bei den Frames um einen objektorientierten Ansatz, in dem Objekte durch binäre Relationen miteinander verbunden werden [132]. Nach [115] ist ein semantisches Netz eine Ansammlung von Punkten und Knoten, die als Begriffe interpretiert werden wobei jeder Knoten einen eindeutigen Namen hat. Die Knoten werden durch Pfeile, die die Relationen darstellen, verbunden. Besteht eine Relation in einer Richtung, dann gilt sie nicht automatisch auch in der umgekehrten Richtung. Die kleinste Informationseinheit in einem semantischen Netz sind zwei durch eine Relation verbundenen Knoten. Ein Begriff enthält also nur dann Informationen, wenn er mit einem anderen Begriff durch eine Relation verbunden ist. In einem semantischen Netz werden zusätzlich noch Token und Typen unterschieden.

Ein Token steht dabei für einen spezifischen Begriff oder eine Instanz, wie bsp. eine bestimmte Person. Ein Typ hingegen ist nicht spezifisch, sondern bezeichnet eine ganze Kategorie möglicher Begriffe. Der Begriff *Mädchen* kann bsp. über eine *istEin*-Relation mit *Mädchen* verbunden sein und ist somit ein Token vom Typ *Mädchen*. Der Unterschied zwischen Token und Typ wird in einem semantischen Netz mittels verschieden benannter Relationen dargestellt. Um Token und Typ miteinander zu verbinden, kann die *istEin*-Relation verwendet werden und für die Relationen zwischen Typen die Subtyp-Relation. Dann besagt die Verkettung "*Bello istEin Hund Subtyp Tier*", dass Bello ein Token vom Typ Hund und Hund ein Untertyp des Typs Tier ist. Durch Ableitungsprozeduren kann also aufbereitetes Wissen, dass aufgrund von logischen Schlussfolgerungen gewonnen wurde, verwirklicht werden. Dieses Wissen wird auch Inferenzen genannt. Beispielsweise könnte zwischen den Begriffen Vögel und Flügel die Relation besitzt bestehen. Dann gilt auch für einen möglichen Begriff Kanarienvogel, der ein Untertyp von Vogel ist, die Relation besitzt zu Flügel. Da die Beziehung schon über den Begriff Vogel existiert, muss sie nicht erneut modelliert werden. Die Eigenschaft, die sich über die Relation ausdrückt kann aber bei Suchen auf dem Netz verwendet werden.

Ontologien

In den letzten Jahren wird in der Forschung zu Wissensrepräsentation, die sich mit der Darstellung von Begriffen und Relationen großer Wissensgebiete beschäftigen, zunehmend der Begriff Ontologie als Oberbegriff für die Darstellung des Wissens verwendet. Das Wort Ontologie stammt aus dem Griechischen und bedeutet "die Lehre vom Sein, von den Ordnungs-, Begriffs- und Wesensbestimmungen des Seienden" [35]. Nach [150] sind Ontologien aus philosophischer Sicht der Versuch zu bestimmen, welche Dinge existieren. Aufgrund der Uneindeutigkeit der natürlichen Sprache ist es schwierig zu entscheiden, was die ontologische Bestimmung eines Begriffs ist, wie der Begriff also in einer Ontologie einzuordnen ist. Witmer geht von der Sprache aus, und versucht sie zu analysieren. Hat eine Sache einen Namen oder beinhaltet oder impliziert ein Satz eine existentielle Generalisierung, für die die Sache vorhanden sein muß, dann existiert die Sache und hat eine Bestimmung in der Ontologie. Für den ersten Fall ist der Satz "Peter streichelt ein Pferd" ein Beispiel. Der Satz deutet eindeutig darauf hin, dass die Person Peter und ein Pferd existieren. Für den zweiten Fall kann der Satz "Peter pflückt Äpfel im Garten" als Beispiel verwendet werden. Die Tatsache, dass Peter im Garten Äpfel pflückt, legt nahe, daß im Garten Apfelbäume stehen und somit existieren, obwohl sie im Satz selbst nicht explizit erwähnt werden. Vor allem die Frage in welchem semantischen Zusammenhang die Begriffe einer Ontologie in Zusammenhang stehen, hat zu

vielen verschiedenen Interpretationen des Begriffs Ontologie geführt, von denen die wichtigsten im folgenden kurz vorgestellt werden.

Ontologie Definition nach Sowa

Sowa erklärt, dass das Ergebnis einer Studie von Kategorien von Dingen eine Ontologie sei [122]. Eine Ontologie ist demnach ein Katalog der Typen von Dingen oder Begriffen, die in einem bestimmten Bereich beziehungsweise Wissensgebiet existieren und beschreibt eine mögliche Welt. Die Typen repräsentieren Prädikate, Wortsinn oder Begriffe und Beziehungstypen der Sprache, in der über die Dinge einer bestimmten Wissensdomäne gesprochen wird. Die Kategorien können nach dieser Definition in einem semantischen Netz angeordnet werden.

Ontologie Definition nach Guarino

In "Understanding, Building and using Ontologies" [49] erklärt Guarino, daß eine Ontologie eine explizite, partielle Darstellung der intendierten Modelle einer logischen Sprache sei. Modelle der realen Welt werden mit Hilfe logischer Sprache beschrieben. In diesen Modellen werden Relationen zwischen den Begriffen aus der Welt beschrieben, daß heißt die semantische Bedeutung der Relationen. Für diese gilt, daß sie dieselbe bleibt, auch wenn die an der Relation beteiligten Instanzen oder Begriffe ausgetauscht werden. Stehen zum Beispiel zwei Kisten A und B aufeinander gestapelt auf einem Tisch, und zwar so, daß das A zuunterst steht, dann steht ein Klotz auf dem Tisch. Werden nun A und B vertauscht, dann besteht immer noch dieselbe Situation, nämlich daß ein Klotz auf dem Tisch steht. Das bedeutet nicht, daß es nicht verschiedenartige Relationen gibt. Neben der Relation *stehtAuf* kann es auch noch die Relation *liegtNeben* geben. Es wird also nur die Struktur eines Ausschnitts der realen Welt beschrieben und Regeln zur Erhaltung dieser Struktur festgelegt. Diese Regeln können außerdem dazu genutzt werden, die Begriffe der Wissensdomäne zu isolieren und organisieren. Eine Ontologie ist also nicht nur eine Klassifikation der betrachteten Dinge in der Welt, sondern beinhaltet auch Einschränkungen die in Axiomen¹ ausgedrückt werden und Relationen zwischen diesen beziehungsweise den Begriffen. Sie beinhaltet somit die Semantik, die einer logischen Sprache fehlt. Auch bei dieser Definition können die Begriffe und Relationen wieder graphisch in einem Netz angeordnet werden; problematisch gestaltet sich hier aber die graphische Darstellung der Axiome.

Ontologie Definition von Gruber

Gruber geht in seiner Definition einer Ontologie von einer abstrakten und vereinfachten Darstellung der Welt aus, einer Konzeptionalisierung [48]. Eine Ontologie ist nach Gruber eine explizite Spezifikation einer Konzeptionalisierung, das heißt in einer Ontologie werden die Dinge der Welt explizit aufgeführt. Das Wissen eines Domänenbereichs wird in einem deklarativen Formalismus dargestellt. Die auf diese Weise repräsentierten Objekte des Wissensbereiches werden in einem Wörterverzeichnis wiedergegeben, auf das ein Programm das Wissen darstellt und verarbeitet, zugreifen kann. Eine Ontologie setzt sich hier aus Definitionen und Axiomen zusammen. Die Definitionen beschreiben Objekte, das heißt die Begriffe, Typen,

1. Unter einem Axiom versteht man hier eine Aussage, die nicht von anderen Aussagen abgeleitet wurde, immer gültig ist und von der andere Aussagen abgeleitet werden können.

Relationen und Funktionen des betrachteten Wissensgebietes und legen somit das Wörterverzeichnis der Ontologie fest. Mit den Axiomen werden wie bei Guarino Bedingungen für die Interpretation und die Benutzung der Objekte festgelegt.

Untersucht man die verschiedenen Modelle vom Standpunkt eines Anwenders aus, so sind die Unterschiede zwischen den Modellen und Definitionen praktisch unerheblich, da sich mit allen vorgestellten Modellen Wissenstrukturen aufbauen lassen. Da sich alle Strukturen graphisch als Netz in Form von Knoten und Kanten repräsentieren lassen, soll im Folgenden als Oberbegriff für die Repräsentation von Wissensstrukturen nur noch von semantischen Netzen gesprochen werden. Die Motivation für die Verwendung des Begriffes semantische Netze liegt dabei einzig und allein darin, daß er am anschaulichsten beschreibt, welche Strukturen sich bei der Modellierung von Wissen ergeben und nicht in den oben beschriebenen Unterscheidungen. Zudem erfordert die Verwendung eines semantischen Netz in einem Lernsystem, eine Repräsentation der Strukturen für Benutzer, die in der Regel über keinerlei Erfahrungen bei der Wissensmodellierung verfügen. Wie in ZZ beschrieben wird, ist die geeignetste Form der Präsentation der Wissenstrukturen für diese Benutzergruppe die Präsentation in Form eines Netz aus Knoten und Kanten.

3.7.2 Einsatz von semantischen Netzen

Jasper fasst die Verwendungsmöglichkeiten von semantischen Netzen basierend auf [64] folgendermaßen zusammen (vgl. [65]):

- 1 Sprachneutrale Erstellung von Dokumenten
Mit Hilfe von Algorithmen ist es möglich, den Inhalt eines textbasierten Dokumentes automatisch zu extrahieren, zu analysieren und in Form eines semantischen Netzes zu speichern. Durch Abbildung dieses Netzes auf ein Netz in einer anderen Sprache kann somit eine automatische Übersetzung des Textes erfolgen
- 1 Spezifikation
Semantische Netze können als Grundlage für die Spezifikation von und Entwicklung von Software und der dazugehörigen Dokumentation verwendet werden
- 1 einheitlicher Zugang zu Informationen
Unter einem einheitlichen Zugang zu Informationen wird die Verwendung eines einheitlichen Vokabulars zur Beschreibung von Dokumenten verstanden.
- 1 Suchfunktionalität
Semantische Netze können zur Suche auf beliebigen Dokumenten, die mit Begriffen des Netz verbunden sind, verwendet werden. Die Suche kann in einem solchen Fall also auch Ressourcen miteinbeziehen, die aufgrund ihrer Kodierung nicht direkt analysiert werden können.

Wille spricht von “begrifflicher Wissensverarbeitung”, wenn er vom Einsatz semantischer Netze spricht. [149]. Als Hauptvorteile der Nutzung werden gesehen:

- 1 Erkunden
Unter Erkunden wird die Möglichkeit verstanden über Strukturen, die das in unüberschaubaren Datenbeständen enthaltene Wissen abbilden, einem Benutzer eine schrittweise Exploration einzelner Aspekte eines Wissensgebietes zu ermöglichen

- 1 Suchen
Die Möglichkeit, Wissenstrukturen zu durchsuchen ist mit dem Erkunden vergleichbar. Der Unterschied liegt nur darin, daß man bei einer Suche davon ausgeht, daß das gesuchte Wissen zu Teilen schon bekannt ist und es daher nur noch explizit gefunden werden muß, während beim Erkunden nur eine vage Vorstellung von dem existiert, was man sucht.
- 1 Erkennen
Erkennen wird in [149] als “über etwas Klarheit gewinnen” beschrieben. Es stellt damit eine Variante des Erkunden dar, da das Erkennen von Strukturen und Zusammenhängen durch das Erkunden von Teilgebieten des Netz erfolgt
- 1 Identifizieren
Mit Identifizieren wird die Einordnung eines Begriffes in ein Klassifikationssystem bezeichnet. Er ist daher insofern problematisch, da diese Definition voraussetzt, daß Teile des semantischen Netz eine hierarchische Klassifikation bilden.
- 1 Untersuchen, Analysieren, Bewußtmachen, Entscheiden
Auch diese Funktionalitäten stellen jeweils nur Varianten oder Kombinationen der bereits beschriebenen Eigenschaften dar.
- 1 Restrukturieren, Behalten, Verbessern
Zusammengefaßt kann unter den Funktionalitäten Restrukturieren, Behalten und Verbessern sowohl der zusätzliche Erwerb und die Neustrukturierung des Wissens aufgrund der Analyse des Netzes verstanden werden, als auch die Erweiterung und Pflege des semantischen Netz durch die Einbringung neuer Begriffe.

Die ersten 4 Möglichkeiten beruhen auf ähnlichen Vorgehensweisen, wie ein menschlicher Benutzer zu dem gewünschten Wissen kommt. Für das Suchen ist ein gezielter Sprung zu einzelnen Begriffen und Teilstrukturen des Netz erforderlich. Ausgehend von den gesuchten Begriffen können dann interaktiv weitere Begriffe, die in Zusammenhang zu den gesuchten stehen untersucht und analysiert werden. Es kann so weiteres Wissen erlangt werden, um die Bedeutung des ursprünglich gesuchten Begriff besser bewerten zu können. Gleichzeitig können neue Begriffe identifiziert werden. Das Erkunden und Erkennen stellt praktisch nur den umgekehrten Weg dar. Durch eine interaktive Exploration des semantischen Netz hin zu gesuchten Begriffen, wird Wissen über Zusammenhänge in denen der Begriff verwendet wird, praktisch im Vorbeigehen mit aufgenommen. Die letzten Punkte zielen auf eine Erweiterbarkeit des Netzes ab. Neue Erkenntnisse, die auf bereits modellierten Wissensgebieten sollen in das Netz eingebracht werden können, damit es immer den aktuellen Stand des Wissens wieder spiegeln kann. Zusätzlich muß die Möglichkeit bestehen, bestehende Netze um neue Wissensgebiete zu erweitern

3.7.3 Architektur

Die klassische Architektur eines wissensbasierten Systems besteht nach [50] aus einer Wissensbasis, die die Dokumente oder Informationen vorhält und einer Inference Engine genannten Komponente, die bei einer Suchanfrage versucht die gewünschten Informationen aus der Wissensbasis zu extrahieren und zu präsentieren. Abbildung 17 zeigt eine Architektur, in der zusätzlich noch eine Wissenserwerbs-, Erklärungs- und eine Dialogkomponente gezeigt sind. Eine Während die Dialogkomponente in jeder Software vorhanden ist, die Benutzereingaben

verarbeitet und daher nicht weiter erläutert werden soll, werden die Wissenserwerbs- und die Erklärungskomponente zur Durchführung einer Suche verwendet, die bei wissensbasierten Systemen in zwei Schritten abläuft. Aufgrund der Informationen, die ein Benutzer spezifiziert hat, werden in einem ersten Schritt diese Informationen in eine Anfrage transformiert und an die Wissensbasis übergeben. In einem zweiten Schritt werden die möglichen Ergebnisse, die die Wissensbasis zurückliefert analysiert und unter Umständen noch zusammengefaßt und gruppiert [37]. Diese Gruppierung kann anhand von Informationen erfolgen, die die Wissensbasis zusammen mit dem Ergebnis liefert, oder durch zusätzliche Informationen. Ein einfaches Beispiel ist die Angabe eines Wertes für die maximale Anzahl der Suchergebnisse. Ein weiteres Beispiel ist, die Gruppierung anhand externer Klassifikationssysteme um die Suchergebnisse strukturiert präsentieren zu können.,

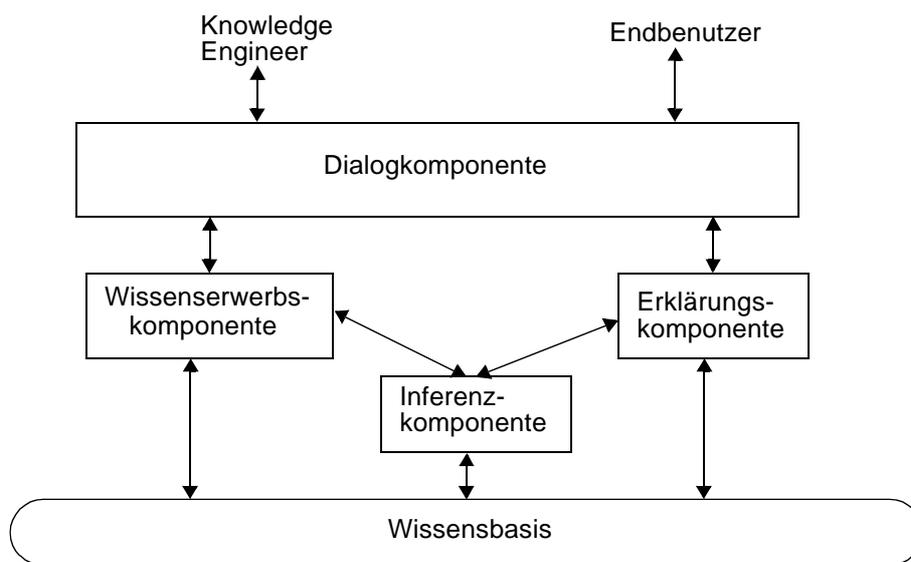


Abbildung 17: Architektur eines wissensbasierten Systems (vgl. [47])

3.7.4 Kodierungsformate für semantische Netze

Wie sich bei der Beschreibung der Austauschformate im Allgemeinen (Abschnitt 2.6.3) und LOM (Abschnitt 3.3.2) gezeigt hat, kann man die Entwicklung von Kodierungsformaten für Datenstrukturen in die Zeit vor und nach der Spezifikation von XML unterscheiden. Dieser Sachverhalt trifft auch auf für semantische Netze zu. Bei semantischen Netzen muß zusätzlich jeweils noch unterschieden werden, ob das Schema oder die Instantziierung eines semantischen Netzes in einem austauschbaren Format kodiert werden soll. Wie in Abschnitt 3.8.2 gezeigt wird, hat die Austauschmöglichkeit eines semantischen Netz eine geringere Bedeutung, als dies bei Metadatenbeschreibungen für Ressourcen der Fall ist. Aus diesem Grund wird in diesem Abschnitt nur kurz auf die drei wichtigsten Vertreter eingegangen. Eine ausführliche Beschreibung findet sich bei [28].

Knowledge Interchange Format (KIF)

KIF wurde an der University of Stanford als Austauschformat für verschiedene Sprachen zur

Beschreibung semantischer Netze entwickelt. KIF verwendet eine Beschreibungslogik und erlaubt die Definition von Begriffstypen, Beziehungen und Funktionen. Basierend auf KIF wurden die Sprachen Ontolingua und Frame Ontology entwickelt. Obwohl semantische Netze, die mit KIF kodiert sind textorientiert sind und damit prinzipiell für menschliche Benutzer lesbar sind, ist die von KIF verwendete Notation äußerst komplex. Eine Trennung zwischen Schema und Instantiierung ist mit KIF nicht möglich; da die Regeln und Beziehungen direkt zusammen mit den Begriffen des Netzes kodiert sind.

Topic Maps

Bei Topic Maps handelt es sich um einen Entwurf zur Kodierung von semantischen Netzen, der 1999 als ISO-Standard verabschiedet worden ist [61]. Topic Maps verwenden zur Beschreibung des Netzes SGML- und HyTime Architekturen und lassen sich daher auch als XML-Beschreibungen kodieren [106]. Ähnlich wie bei KIF lassen sich mit einer Topic Map Begriffe und Beziehungen zwischen den Begriffen, Regeln, Funktionen und Axiomen beschreiben. Dabei können Schema und Instantiierung eines semantischen Netz in einer einzigen Topic Map beschreiben, oder auch separat gespeichert werden. Topic Maps bieten im Gegensatz zu KIF nur sehr eingeschränkte Möglichkeiten Regeln und Axiome zu spezifizieren; entsprechend einfacher lassen sich dafür Werkzeuge zur Suche und Navigation auf einem semantischen Netz realisieren.

Ontology Inference Layer (OIL)

Der Ontology Inference Layer (OIL) ist ein Vorschlag für eine web-basierte Repräsentation und eine Anfragespezifikation für semantische Netze, der frame-basierte Ansätze und Beschreibungslogiken vereint [103]. Ähnlich wie TopicMaps, bietet OIL deutlich eingeschränktere Anfragemöglichkeiten als KIF [40]. Der große Vorteil von OIL besteht darin, daß OIL auf verschiedenen aufeinander aufbauenden Schichten besteht, wobei die höheren Schichten komplexer, dafür aber mächtiger sind [8]. Die unterste Schicht (Core OIL) verwendet zur Kodierung, das in Abschnitt 2.6.3 beschriebene RDF Schema. Dies bedeutet, daß Anwendungen, die in der Lage sind RDF-Schema-Beschreibungen zu verarbeiten, automatisch OIL-Strukturen lesen und die Kodierungen der untersten Schicht analysieren können.

Corcho und Gomez-Perez untersuchen in [28], die Fähigkeiten der verschiedenen Datenstrukturen für semantische Netze und versuchen eine Aussage zu treffen, für welchen Anwendungsfall welche Sprache geeignet ist. Problematisch bei diesem Vergleich ist, daß alle Sprachen formal die gleiche Mächtigkeit besitzen [123]. Entsprechend lautet die Empfehlung, daß die Auswahl wesentlich davon abhängt, welche Werkzeuge zur Erstellung und Pflege eines semantischen Netz zur Verfügung stehen. Eine Untersuchung bestehender Werkzeuge wird in Abschnitt 5.1 beschrieben.

3.8 Einsatz von semantischen Netzen in Lernsystemen

Alle im letzten Abschnitt beschriebenen Nutzen können nur dann auftreten, wenn alle Begriffe des semantischen Netz bekannt sind. Neues Erkennen und Restrukturieren kann nur dadurch erfolgen, daß man bei der Exploration des semantischen Netz bemerkt, daß zwischen zwei

Begriffen, die man schon kennt, ein Zusammenhang besteht, der vorher unbekannt war. Stößt man während der Exploration des Netzes auf einen unbekanntem Begriff ist der Nutzen gering, da keine weiteren Information außer dem Begriff selbst zur Verfügung stehen um die Bedeutung des Begriffs zu erfahren. Während man bei der alleinigen Verwendung von LOM nicht in der Lage ist Zusammenhänge von Begriffen und Strukturen eines Wissensgebietes zu modellieren, fehlt bei semantischen Netzen die Möglichkeit detaillierte Informationen über Begriffe und Schlagwörter zu verwalten, da nicht vorgesehen ist, Begriffe mit Medienbausteinen zu verbinden, sondern nur mit anderen Begriffen.

Ein Ansatz semantische Informationen und Metadatenbeschreibungen von Module zur Informationsspeicherung bei einem web-basiertes Lernsystem zu verwenden ist in [24] beschrieben. In dieser Arbeit werden Algorithmen zur automatisierten Komposition von Kursen aus modularen Einheiten untersucht und entwickelt. Die Medienbausteine sind zum einen mit Metadaten ausgezeichnet, wobei LOM als ein geeignetes Metadatenschema bezeichnet wird. Gleichzeitig wird auch die Notwendigkeit betont, unabhängig von den Bausteinen, semantische Information zur Verfügung zu haben, da nur dann die nötige Information vorliegt, die die Algorithmen zur Auswahl und Anordnung der Module benötigen. Entstehen sollen diese Netze allerdings aus den Modulen selbst heraus. Durch eine direkte Inhaltsanalyse oder durch Ausnutzen der Informationen der Metadatenbeschreibungen, werden die Schlüsselwörter einer Ressource identifiziert. Bei der Anordnung der Schlüsselwörter wird versucht eine hierarchische Struktur nachzubilden. Begründet wird diese Struktur mit der Position eines Moduls in dem zu erstellenden Kurs. Für eine Kursstruktur wird daher wie bei einem linearen text eine hierarchische Struktur angenommen, die einen Kurs in Abschnitte unterteilt, die wiederum beliebig viele Unterabschnitte enthalten können. Anhand dieser Strukturen werden dann die extrahierten Schlagwörter in eine Oberbegriff-Unterbegriff-Beziehung gebracht und somit ein einfaches Klassifikationssystem erstellt. Unabhängig davon, daß in der Praxis die automatische Generierung eines Klassifikationssystems durch Textanalysen nur sehr eingeschränkt funktioniert, wird bei diesem Ansatz einem Modul eine feste Position innerhalb eines Kurses zugewiesen. Ein Modul kann in einem Kurs also nur an einer bestimmten und eindeutig festgelegten Stelle verwendet werden und es kann vor allem auch nur innerhalb eines einzelnen Kurses eingesetzt werden. Zudem muß das Klassifikationsmodell für jeden Kurs neu erstellt werden. Obwohl bei diesem Ansatz also semantische Strukturen bei der Kursgenerierung berücksichtigt werden, bleibt er zu sehr am einzelnen Modul orientiert. Die generierten taxonomische Strukturen beziehen sich immer nur auf den einzelnen Medienbaustein; eine Modellierung der Struktur des kompletten Wissensgebiets findet nicht statt; vielmehr werden Teilbereiche mehrmals modelliert, wenn sich die Inhalte der Medienbausteine überschneiden. Zudem wird nichts darüber ausgesagt, wie die Informationen, die zur Generierung der Kurse benötigt werden strukturiert und verwaltet werden. Bei der Verwendung eines semantischen Netz existiert das Netz und damit das im Netz gespeicherte Wissen auch ohne Medienbausteine. Es kann völlig ohne Zuhilfenahme von konkreten physikalischen Ressourcen zur Vorauswahl und Gliederung eines Wissensgebietes, über das Informationen bereitgestellt werden sollen, verwendet werden.

In [10] wird empfohlen Lernenden bei der Arbeit mit einem web-basierten Lernsystem zusätzlich zu den multimedialen Lerneinheiten ein Inhaltsverzeichnis über das gesamte in einem Kurs behandelte Material zur Verfügung stellen. Eine von den Autoren durchgeführte Evaluation mit 200 Studenten ergab, daß bei Tests Studenten deutlich besser abschnitten, die zusätzlich zu den Lernmaterialien auch ein Inhaltsverzeichnis zur Verfügung hatten, daß den vermittelten Lerninhalt strukturierte und jeweils anzeigte, wo im Kurs sich die Studenten beim

Aufruf einer neuen Seite befanden

Der in [24] beschriebene Ansatz versucht diese Strukturierung für einzelne unabhängige Module durch Analyse der Metadatenbeschreibungen nachzubilden. Dieses Verfahren kann nicht angewendet werden, wenn für einen kompletten Kurs, der aus unabhängigen Medienbausteinen zusammengesetzt werden soll, ein Inhaltsverzeichnis erstellt werden soll. Wie im folgenden gezeigt wird, kann durch die Verwendung eines semantischen Netz, eine Strukturierung für eine Kombination, aus Medienbausteinen und damit für einen kompletten Kurs erstellt werden kann.

Die Struktur eines Inhaltsverzeichnisses für einen Kurs wird grundsätzlich durch semantische Beziehungen zwischen Begriffen gebildet, unabhängig davon ob das Material dem Lerner in Buchform oder elektronisch präsentiert wird [120]. Ein Inhaltsverzeichnis kann damit als ein Pfad durch ein semantisches Netz angesehen werden. Beispiele für Inhaltsverzeichnisse und die semantischen Beziehungen zwischen den Begriffen sind in Abbildung 18 und Abbildung 19 gezeigt

- 1. Media Server
 - 1.1 Disc Controller
 - 1.2 Storage Management
 - 1.3 File System
 - 1.3.1 Allocation Table
 - 1.3.2 Access Control
 - 1.4 Memory Management

Abbildung 18: Beispiel für ein Inhaltsverzeichnis I

In diesem Fall besteht zwischen den Kapitelüberschriften des Inhaltsverzeichnis und den Überschriften der Unterkapitel jeweils eine semantische Relation des Typs “besteht aus”. Ein Inhaltsverzeichnis für einen Kurs über das Bildkomprimierungsverfahren JPEG zeigt die folgende Abbildung 19 In diesem Beispiel besteht zwischen der Kapitelüberschrift und den

- 1. Image Compression
 - 1.1 JPEG
 - 1.2 PNG
 - 1.3 TIFF

Abbildung 19: Beispiel für ein Inhaltsverzeichnis II

Begriffen der Unterkapitel jeweils die semantische Relation “ist ein”.

Damit stellt die spätere Präsentation eines Inhaltsverzeichnis, das aus einem semantischen Netz extrahiert wurde aber nur einen sekundären Nutzen dar. Viel entscheidender ist, daß durch eine Kombination von semantischen Netzen und Metadatenbeschreibungen für Lernressourcen, das in Abschnitt 3.7.2 beschriebene Problem einer für den Lerner und das Wissensgebiet angepaßten Reihenfolge von Medienbausteinen gelöst werden kann. Die in 3.7.2 beschriebenen Nutzen eines semantischen Netz beziehen sich nämlich nur auf einen menschl-

chen Benutzer, der das Netz explorativ nutzt, um neues Wissen zu erlangen. Eine rechnerbasierte Suche, die die semantischen Beziehungen eines Inhaltsverzeichnis nutzt, kann eine Struktur und vor allem eine Reihenfolge der Begriffe eines Wissensgebietes, das in einem Kurs vermittelt werden soll, extrahieren [121]. Die Identifikation und Zusammenstellung von Medienbausteinen zu einem Kurs kann daher in zwei Stufen ablaufen. Im ersten Schritt findet eine Auswahl und Strukturierung der relevanten Begriffe durch eine maschinelle Suche auf einem semantischen Netz dar. Als Ergebnis erhält man durch diesen Schritt die Menge der Begriffe über die der Kurs Informationen bereitstellen soll und gleichzeitig eine Reihenfolge in der die Begriffe präsentiert werden sollen. Anders ausgedrückt, beantwortet der erste Schritt die Frage: Was soll in einem Kurs präsentiert werden?. Diese Frage kann unabhängig von den physikalischen Eigenschaften eines Medienbausteins beantwortet werden. Anhand der Struktur des Kurses können dann in einem zweiten Schritt passende Medienbausteine ausgewählt werden, die die Struktur des Kurses mit Inhalt füllen. In diesem Schritt wird also die Frage: "Wie soll die Information im Kurs präsentiert werden?" beantwortet. Aufgrund von persönlichen Präferenzen oder physikalischen Gegebenheiten bei der Infrastruktur des Lernalters können Medienbausteine zu einem bestimmten Begriff ausgewählt werden, die die Informationen in der gewünschten Form präsentieren können. Diese beiden Schritte können für verschiedene Benutzer unabhängig durchgeführt werden, sodass durch die Kombination von semantischen Netzen und Metadatenbeschreibungen individuell angepasste Kurse aus Medienbausteinen generiert werden können [126].

Ein Beispiel für zwei individuell generierte Kursstrukturen die mit Hilfe eines semantischen Netz erstellt werden können zeigen die folgenden Abbildungen, die beide als Grundlage ein semantisches Netz verwenden, das einen Ausschnitt der multimedialen Datenkompression modelliert. In beiden Abbildungen ist das vollständige Netz jeweils auf der grau hinterlegten Fläche abgebildet. Die ausgewählten Begriffe sind weiss hervorgehoben und die Reihenfolge der Anordnung wird in einem zusätzlichen Fenster dargestellt.

die Erstellung der Struktur der kombinierten Einheit nötig sind, können aus der Integration der bekannten Technik der semantischen Netze mit LOM-Metadaten gewonnen werden.

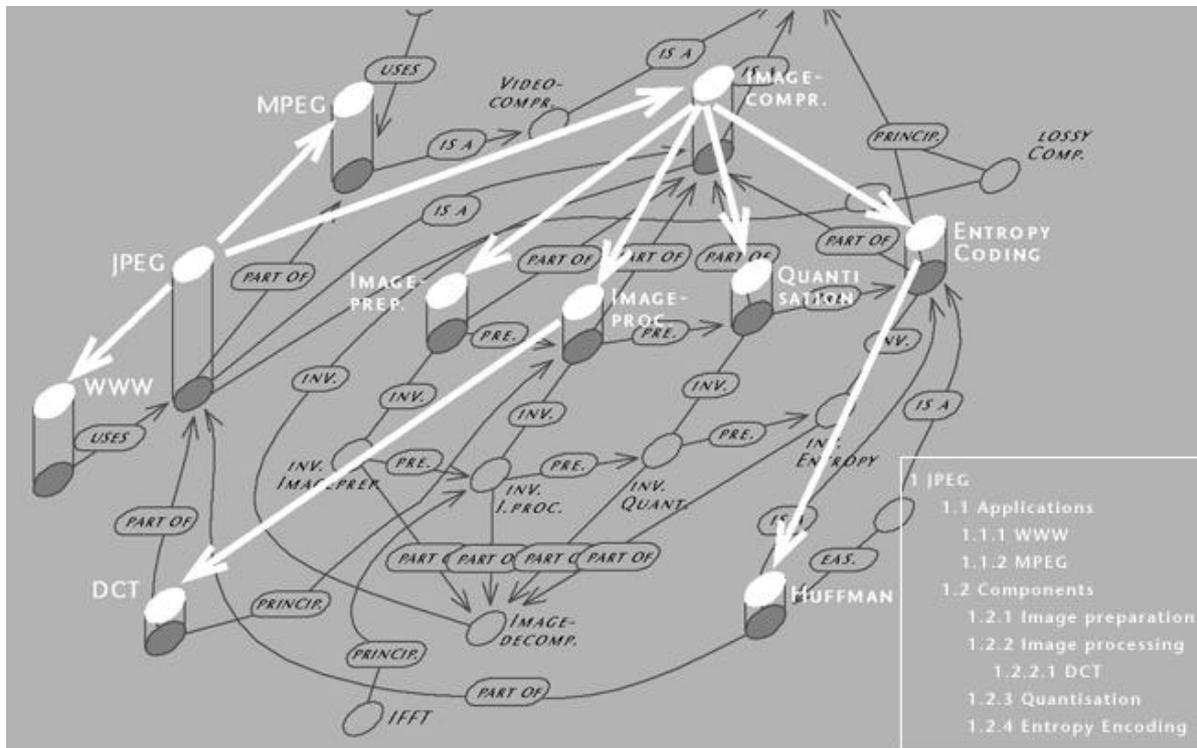


Abbildung 21: extrahiertes Inhaltsverzeichnis über JPEG

3.8.1 Datenmodell

Der Idealfall des Datenmodells für ein web-basiertes Datenmodell stellt damit eine Kombination eines semantischen Netzes und LOM-Beschreibungen dar [118]. Eine schematische Darstellung eines solchen Modells zeigt Abbildung 22. Die in der Abbildung hellgraugrau gezeichnete Ebene, stellt das Web in seiner heutigen Form dar. Die dunkelgrauen Flächen zeigen web-basierte Ressourcen, die durch Links miteinander verbunden sind. Aufbauend auf diesen physikalischen Ressourcen wird durch semantische Netze und die Metadatenbeschreibungen der Ressourcen ein weiteres Netz erstellt, das sich aus logischer Sicht, oberhalb der physikalisch vorhandenen Bausteine befindet. Die in der Abbildung blau gefärbten Knoten stellen dabei einzelne Begriffe eines semantischen Netz dar, die rot gefärbten Knoten repräsentieren LOM-Beschreibungen. Man erkennt in der Abbildung daß sich die LOM-Beschreibungen jeweils auf eine konkrete physikalische Ressource beziehen, während die Begriffe des semantischen Netz unabhängig von einer konkreten Repräsentation existieren. Aus der Abbildung ist weiter ersichtlich, daß Beziehungen zwischen Metadatenbeschreibungen existieren können, die keine Entsprechung bei den beschriebenen Bausteinen haben und das umgekehrt Bausteine Verbindungen zu anderen Bausteinen haben können, die sich nicht in den Metadatenbeschreibungen widerspiegelt. Zusätzlich kann ein Medienbaustein Verbindungen zu

beliebigen vielen Begriffen aus verschiedenen semantischen Netzen haben.

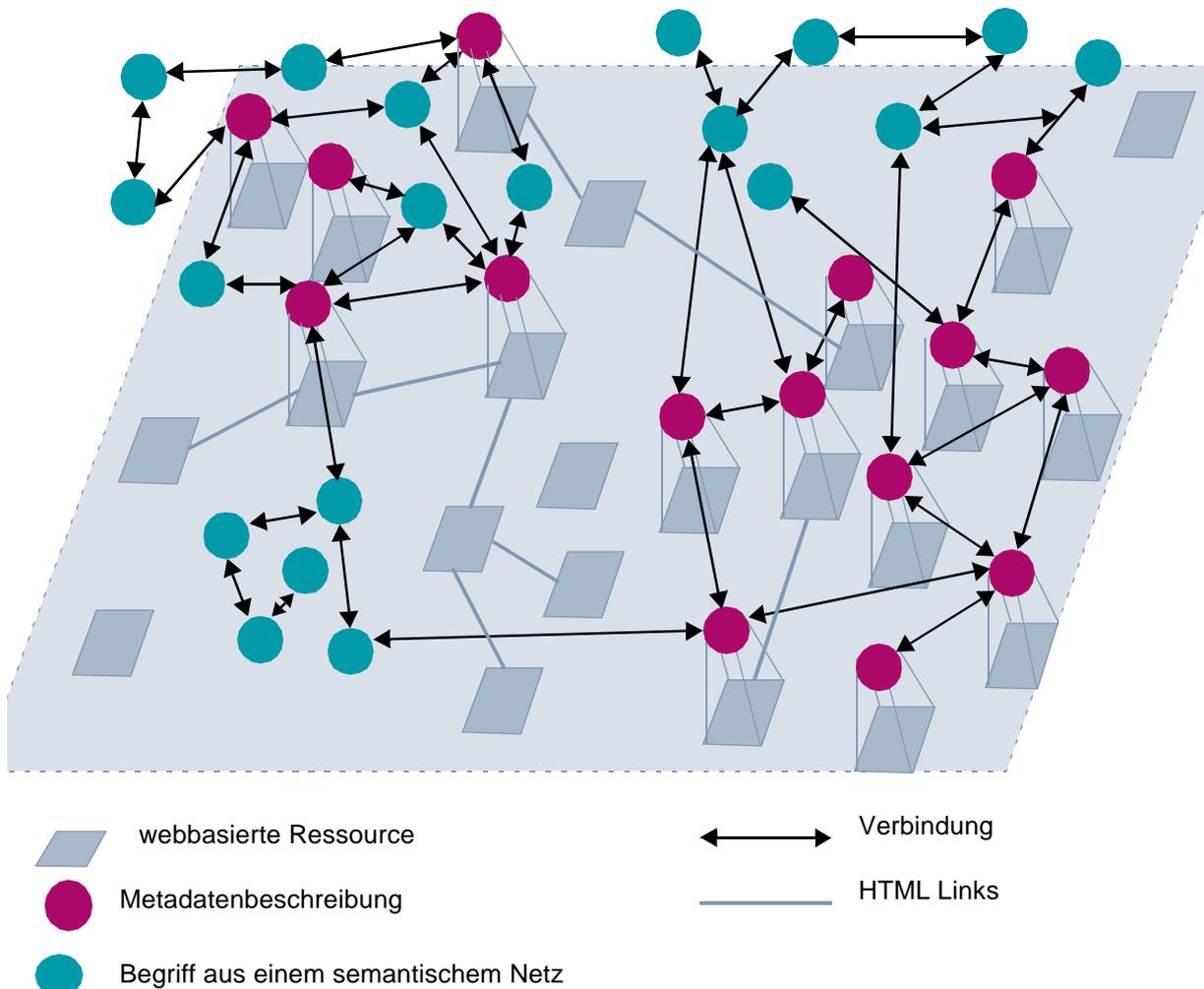


Abbildung 22: Kombination von LOM-Beschreibungen und semantischen Netzen

3.8.2 Kodierung

Während in den vorherigen Abschnitten die Kombination von LOM und semantischen Netzen auf einer abstrakten Ebene diskutiert wurde, wird in diesem Abschnitt untersucht, wie eine Kombination durch die Kodierung der einzelnen Elemente erfolgen kann. Die einfachste Möglichkeit stellt die Kodierung sowohl der semantischen Netze, als auch der LOM-Beschreibungen mit einem einheitlichen Schema dar. In diesem Fall können die Metainformationen durch die gleiche Datenstruktur repräsentiert werden, was die Implementierung von Algorithmen die die Strukturen untersuchen und vor allem die Speicherung der Informationen deutlich erleichtert. Betrachtet man die in Abschnitt 2.6, Abschnitt 3.3.2 und Abschnitt 3.7.4 beschriebenen Verfahren zur Kodierung so kann man feststellen, daß es prinzipiell möglich ist sowohl semantische Netze als auch LOM mit Hilfe einer RDF-Schema-Beschreibung zu kodieren. In der Praxis existiert aber weder eine endgültige Version von RDF-Schema des W3Cs, noch sind Beschreibungen semantischer Netze oder LOM-Beschreibungen, die mit RDF-Schema kodiert

sind, verfügbar. Der in Abbildung 22 gezeigte Idealfall der Kombination von semantischen Netzen und LOM-Beschreibungen lässt sich daher derzeit noch nicht durch eine einheitliche Kodierung umsetzen. Betrachtet man den Einsatz eines semantischen Netzes in einem Lernsystem, so kann festgestellt werden, daß zu keinem Zeitpunkt das vollständige Netz außerhalb des Systems das das Netz verwaltet zur Verfügung stehen muß. Dies bedeutet, daß nicht jeder Begriff des Netzes in Form einer eindeutigen URL zugreifbar sein muß. Aus diesem Grund relativiert sich der Nutzen einer einheitlichen Beschreibung erheblich. Während die roten Knoten in Abbildung 22, die die Metadaten repräsentieren physikalisch direkt ansprechbar sind, werden Begriffe aus einem semantischen Netz nur in Form einer Resultatmenge als Ergebnis einer Anfrage an die Inference Engine zur Verfügung stehen. Die blauen Knoten existieren daher praktisch immer nur für eine begrenzte Zeit und werden nur bei Bedarf generiert.

Steht kein einheitliches Schema zur Verfügung, so muß entschieden werden, ob die Verbindung durch einen expliziten Verweis in den jeweiligen Datenstrukturen erfolgen soll, oder die Verbindung nur indirekt realisiert werden soll. Bei einer indirekten Verbindung muß festgelegt werden, welches Element einer Metadatenbeschreibung für die Realisierung der Verbindung verwendet werden soll. Bei LOM bietet sich das *Keywords*-Element für dieses Element an. Dabei hat eine indirekte Verbindung zwischen Begriffen des semantischen Netz und den Medienbausteinen sowohl Vor- als auch Nachteile:

1 Vorteile

Wie bereits beschrieben, ist ein großes Hindernis bei der Beschreibung von multimedialen Ressourcen mit Metadaten der nötige Aufwand zur Erstellung der Beschreibungen. Eine implizite Verbindung zwischen den Begriffen eines semantischen Netzes und den Schlüsselwörtern eines Medienbausteins hat damit den Vorteil, daß die Verbindung nicht vom Autor einer Metadatenbeschreibung vorgenommen werden muß. Eine Suche nach Medienbausteinen kann daher in zwei Schritten ablaufen: Zuerst wird durch eine maschinelle Suche oder eine interaktive Exploration des Netzes ein Begriff ausgewählt. Dieser Begriff kann dann in einem zweiten Schritt als Suchkriterium nach Medienbausteinen verwendet bei einer Suchanfrage auf LOM-Beschreibungen werden. Auf diese Weise können die Medienbausteine völlig unabhängig vom semantischen Netz gespeichert und verwaltet werden und es ist daher möglich für die Speicherung der semantischen Netze und der Metadatenbeschreibungen unabhängige und spezialisierte Datenbanken zu verwenden. Zudem ist ein Medienbaustein, dessen Metadatenbeschreibung neu in einer Wissensbasis eingetragen wird, unmittelbar über eine Suche auf dem semantischen Netz verfügbar, ohne daß die Verbindung zwischen entsprechenden Begriffen des Netzes und dem Medienbaustein explizit etabliert werden muß. Ein weiterer Vorteil einer impliziten Verbindung liegt darin, daß die Kodierung des semantischen Netz unabhängig von der Kodierung der Metadatenbeschreibungen gewählt werden kann.

1 Nachteile

Der Hauptnachteil einer impliziten Verbindung liegt darin, daß nicht erzwungen werden kann, daß für die Begriffe des semantischen Netz und den Schlüsselwörtern der Medienbausteine das exakt gleiche Vokabular verwendet wird. Werden als Schlüsselwörter Begriffe verwendet, die nicht oder nur unter einem Synonym im semantischen Netz vorhanden sind, können diese Bausteine nicht ohne zusätzlichen Aufwand über eine Suche mit Hilfe des semantischen Netz gefunden werden.

Will man den Zusammenhang eines Medienbausteines zu einem Begriff im semantischen Netz

explizit kodieren, so erfordert dies, das beide Beschreibungsformate bestimmte Eigenschaften aufweisen:

- 1 Beziehungsfunktionalität
Diese Eigenschaft scheint offensichtlich, da keine Beziehung zwischen einem Begriff und einer Metadatenbeschreibung etabliert werden kann, wenn nicht beide Beschreibungsformate die Möglichkeit bieten, Beziehungen zu spezifizieren. Sie beschränkt jedoch die Auswahl an Kodierungsformate und in der Praxis werden zudem auch häufig Formate eingesetzt.
- 1 Verweis auf externe Beschreibungsformate
Wird für die Kodierung der semantischen Netze und der Metadatenbeschreibungen kein einheitliches Format verwendet, so muß es innerhalb eines Formates möglich sein, Verweise auf Metadaten, die in einem anderen Format kodiert sind, zu speichern.
- 1 Adressierbarkeit
Sowohl die Metadatenbeschreibung eines Medienbausteines, als auch ein einzelner Begriff des semantischen Netz müssen direkt adressierbar sein. Bei LOM ist hierfür in der *General*-Kategorie, das Metadatenelement *Identifier* vorgesehen. Bisher gibt es im offiziellen LOM-Dokument aber keinerlei Hinweis, wie dieser Identifier spezifiziert werden kann. Dabei gestaltet sich die eindeutige Bezeichnung einer Metadatenbeschreibung noch verhältnismäßig einfach. Wird eine LOM-Beschreibung als XML-Datei abgespeichert, so gehören alle Elemente der XML-Struktur zu derselben Metadatenbeschreibung. Adressiert werden muß demzufolge nur die komplette Struktur, wenn auf die Metadatenbeschreibung verwiesen werden soll. Deutlich schwieriger ist die Realisierung eines Verweises auf einen einzelnen Begriff des semantischen Netz, da bei semantischen Netzen einzelne Begriffe nicht separat kodiert werden, sondern nur das Netz als Ganzes.

Welches von den in Abschnitt 3.7 vorgestellten Kodierungsformaten für semantische Netze in der Praxis verwendet wird ist unerheblich. Alle weisen formal gesehen die gleiche Mächtigkeit auf und daher wird die Entscheidung von der Möglichkeit abhängen, wie Benutzer Anfragen an das semantische Netz stellen können, oder wie sich die Performance der Datenstruktur bei Anfragen verhält. [123].

Mit der in Abschnitt 3.3.2 entwickelten DTD für LOM steht ein Austauschformat für LOM-Beschreibungen zur Verfügung. Zudem ist XML vollständig textorientiert, sodass ohne Probleme Verbindungen zu Begriffen in einem semantischen Netz als Anfrage kodiert in einer LOM-Beschreibung gespeichert werden können. Wie beschrieben, kommt dem Kodierungsformat der semantischen Netze eine geringere Bedeutung zu. Wird das komplette semantische Netz bsp. für eine Replikation oder zur Datensicherung dennoch benötigt, dann stehen wie in Abschnitt 3.7.4 erwähnt verschiedene XML basierte Austauschformate zur Verfügung. Auch bei den einzelnen Begriffen des semantischen Netz können damit Verbindungen zu LOM-Beschreibungen verwaltet werden.

3.8.3 Architektur

Als Ausgangsbasis für die Entwicklung einer Architektur für ein Lernsystem kann die in Abschnitt 3.1 gezeigte Abbildung 12 dienen, die die Architektur der LTSC zeigt. In dieser

Architektur besteht die Wissensbasis allerdings nur aus einem Repository zur Speicherung der LOM-Beschreibungen. Zudem beschränken sich die im System aufgeführten Rollen auf den Lerner und Lehrer. Offen bleibt bei dieser Architektur die Frage, wer die Metadatenbeschreibungen der Medienbausteine erstellt und wie eine Unterstützung bei dieser Tätigkeit realisiert werden kann.

Betrachtet man die Erstellung und Verwaltung von Informationen in einer herkömmlichen Bibliothek, so kann man feststellen, daß die Aufgaben der Erstellung der Ressourcen und die Auszeichnung mit Metadaten in der Regel nicht von derselben Person vorgenommen werden. Bei den Metadaten handelt es sich ausschließlich um bibliographische Informationen, die von speziell ausgebildeten Bibliothekaren erstellt werden. Da es sich bei diesen Metadaten häufig nicht um inhaltsbezogene Daten handelt, ist der Autor der Ressource in den Auszeichnungsprozeß nicht direkt eingebunden. Bei der Beschreibung eines Medienbausteins mit LOM stellt sich die Situation anders dar, da LOM stark inhaltsbezogene Informationen zur Verfügung stellt. Beispielsweise lässt sich bei keinem Feld der Educational-Kategorie von LOM ein zutreffender Wert ermitteln, wenn man den Inhalt der Ressource nicht genau kennt. Dies bedeutet, daß der Autor einer Ressource bei der Erstellung der Metadaten stärker eingebunden werden muß, als das bei bisherigen MetadatenSchema erforderlich war. Nur wenn für eine Lernressource detaillierte und zutreffende Beschreibungen existieren, kann eine maschinelle Suche anhand von Benutzerpräferenzen erfolgreich sein.

Diese Aussage gilt in einem noch größeren Maße für die Erstellung eines semantischen Netz. Entsprechend kommt dem Autor des Netzes eine große Bedeutung zu. Während der Autor einer Ressource und der Autor der Metadatenbeschreibung nur über das Wissen verfügen muß, das in der Ressource vermittelt wird, muß der Autor des semantischen Netz detaillierte Kenntnisse über ein größeres Wissensgebiet und vor allem über die Zusammenhänge aller relevanten Begriffe eines Wissensgebietes besitzen, um das semantische Netz erstellen zu können. Eine eindeutige Bezeichnung für den Autor eines semantischen Netz existiert bisher nicht. Aus diesem Grund wird im folgenden der aus dem Bereich des Wissensmanagement bekannte Begriff des Wissensexperten oder Knowledge Engineer verwendet.

Abbildung 23 zeigt die Komponenten eines Lernsystems, bei dem sowohl die Rollen Ressourcenautor, Metadatenautor und Knowledge Engineer, als auch die Komponente Semantic Network mit in die Architektur aufgenommen sind. Zusätzlich ist in der Abbildung zu erkennen, daß die gemeinsame Speicherung der Metadatenbeschreibungen zusammen mit der Verwaltung der physikalischen Ressource zugunsten einer getrennten Verwaltung verändert wurde. Dies lässt sich durch die unterschiedlichen Aufgaben und Eigenschaften der beiden Komponenten in der Praxis begründen. Die zunehmende Mobilität der Lerner, erfordert es Lernmaterialien jederzeit an jedem Ort verfügbar zu machen. Enthält das Lernmaterial dabei eine große Menge an multimedialen Ressourcen, die eine große Bandbreite bei der Präsentation benötigen, lässt sich diese Forderung nicht mit einer zentralen Verwaltung der multimedialen Elemente realisieren. Liepert et. al beschreiben in [77] einen verteilten MedienServer, der für die Präsentation von Lerninhalten verantwortlich ist, sich dynamisch an wechselnde Dienstgütereigenschaften anpassen und durch Replikation der Ressourcen eine permanente Verfügbarkeit des Materials garantieren kann. Durch die Verwendung eines Medien Servers kann also nicht nur sichergestellt werden, daß auf alle Medienbausteine permanent zugegriffen werden kann. Die Fähigkeit eines Medien Servers, sich dynamisch an verschiedene Übertragungs- und Präsentationsmedien anzupassen, ermöglicht zusätzlich eine individuelle Präsentation des Lern-

materials unabhängig vom Inhalt der Medienbausteine.

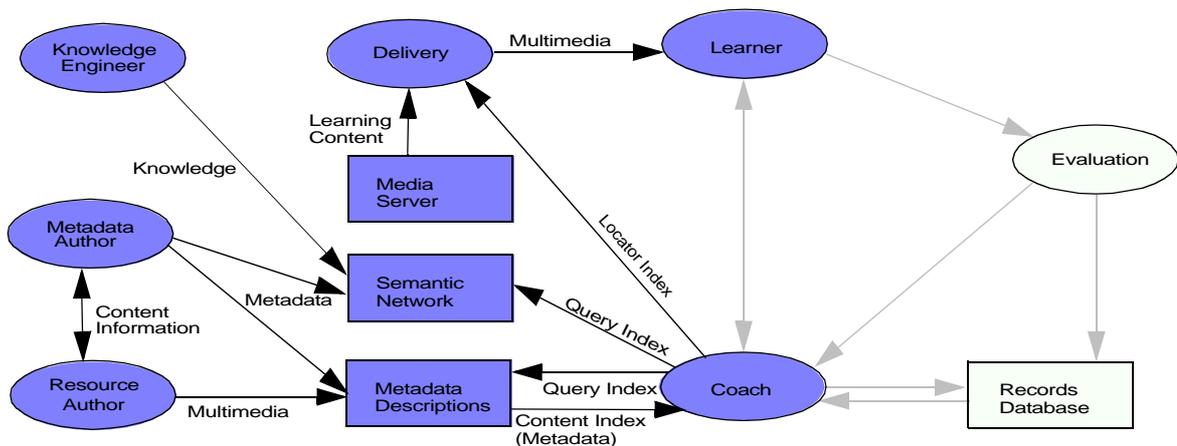


Abbildung 23: Komponenten eines Lernsystems

3.8.4 Werkzeuge zur Erstellung der Metadaten

Im bisherigen Teil der Arbeit wurde die Notwendigkeit und der Nutzen von semantischen Netzen und Metadatenbeschreibungen für multimediale Lernressourcen beschrieben. In Abschnitt 3.8.3 wurden die für die Erstellung der Metainformationen zuständigen Rollen eingeführt. Bisher offen blieb die Frage, auf welche Weise die Autoren die Beschreibungen erstellen und einer Suche zugänglich machen können. De Bra untersucht in [30] den Einsatz von Lernsystemen und kommt zu dem Schluss daß das Hauptproblem der derzeitigen adaptiven web-basierten Lernsysteme darin liegt, daß sie für und von Informatikern entwickelt werden. Der Authoringprozeß und die Auszeichnung der Ressourcen mit Metadaten wird demnach als zu kompliziert für ungeübte Benutzer angesehen. Sowohl die Einbringung einer Ressource in die Wissensbasis des Systems als auch die Erstellung der jeweiligen Metadatenbeschreibung müssen demzufolge durch Werkzeuge unterstützt werden, die die Komplexität der Wissensbasis und die verwendeten Datenstrukturen von den Benutzern fernhalten.

Auch wenn die Informationsmenge im WWW eine Auszeichnung von Ressourcen erzwingt und kommerzielle Szenarien, die Motivation für Metadatenbeschreibungen erheblich steigern, bleibt für die Beschreibung der Materialien ein signifikanter Aufwand bestehen. Dies bedeutet, daß man den Autoren der Metadaten so weit wie möglich mit einer teilautomatisierten und intuitiven Erstellung der Beschreibungen entgegenkommen muß. Berücksichtigen muß man dabei sowohl, die verschiedenen Rollen, die für die Erstellung der Metadaten und der semantischen Netze verantwortlich sind, als auch die unterschiedlichen Anforderungen an die Werkzeuge, aufgrund der Strukturen der LOM-Beschreibungen und der semantischen Netze. Abgesehen von den Kategorien Relation und Classification sind LOM-Metdaten direkt einer Ressource zugeordnet und beschreiben nur direkte Eigenschaften und keine Zusammenhänge. Die Erstellung einer Beschreibung kann daher nahezu unabhängig von bestehenden Ressourcen erfolgen. Selbst wenn diese Elemente, die einzigen sind, die zur Verfügung stehen um die Ressource zu beschreiben, können sie für eine Suche verwendet werden. Werden solche Informationen gesucht, so sind benachbarte Ressourcen bei der Präsentation der Suchergebnisse im ersten Schritt uninteressant. Ganz anders verhält es sich bei der Erstellung semantischer Netze.

Ein einzelner Begriff ohne Verbindung zu anderen Knoten im semantischen Netz ist praktisch nicht zu verwenden. Erst durch die semantischen Beziehungen die ein Begriff mit anderen Begriffen im Netz besitzt, kann er bei der Suche und bei einer Navigation einen Beitrag leisten. Entscheidend bei der Entwicklung eines Werkzeugs zur Erstellung eines semantischen Netzes, ist also die Möglichkeit neu hinzukommende Begriffe auf einfache und intuitive Weise, mit dem bestehenden Netz zu verbinden. Gleichzeitig muß die Möglichkeit geschaffen werden, interaktiv auf dem bestehenden Netz zu navigieren um neben einer maschinellen Suche, auch eine interaktive Detailsuche zu ermöglichen.

Sowohl der Wissensexperte, als auch der Autor der Metadatenbeschreibung einer Ressource benötigen einen Zugang zum semantischen Netz. Der Wissensexperte erstellt das eigentliche Netz; der Autor der LOM-Beschreibung stellt die Verbindung zwischen Medienbausteinen und Begriffen im semantischen Netz her. Beide Rollen benötigen daher eine Möglichkeit durch eine direkte Suche oder durch eine interaktive Navigation einzelne Begriffe des Netzes zu identifizieren, um dann entweder neue Begriffe hinzufügen zu können, oder die Verbindung zu den Medienbausteinen herzustellen. Da weder der Wissensexperte noch der Autor der Ressource, Experten in Wissensmodellierung sind, muß ein Werkzeug zur Verwaltung eines semantischen Netzes, die komplexe Struktur des zugrundeliegenden Datenmodells möglichst vollständig vor dem Benutzer verbergen.

Abbildung 24 zeigt einen Entwurf für ein Lernsystem, das auf LOM und semantischen Netzen beruht, bei dem die Entwicklung der Autorenwerkzeuge, im Gegensatz zu Abbildung 23, direkt mit in die Architektur aufgenommen wurde. Die Wissensbasis besteht in dieser Abbildung aus dem semantischen Netz, den Metadatenbeschreibungen der Ressourcen und den Ressourcen selbst. Verschiedenen Lernern kann aufgrund der Wissensbasis individuell und dynamisch generiertes Kursmaterial angeboten werden.

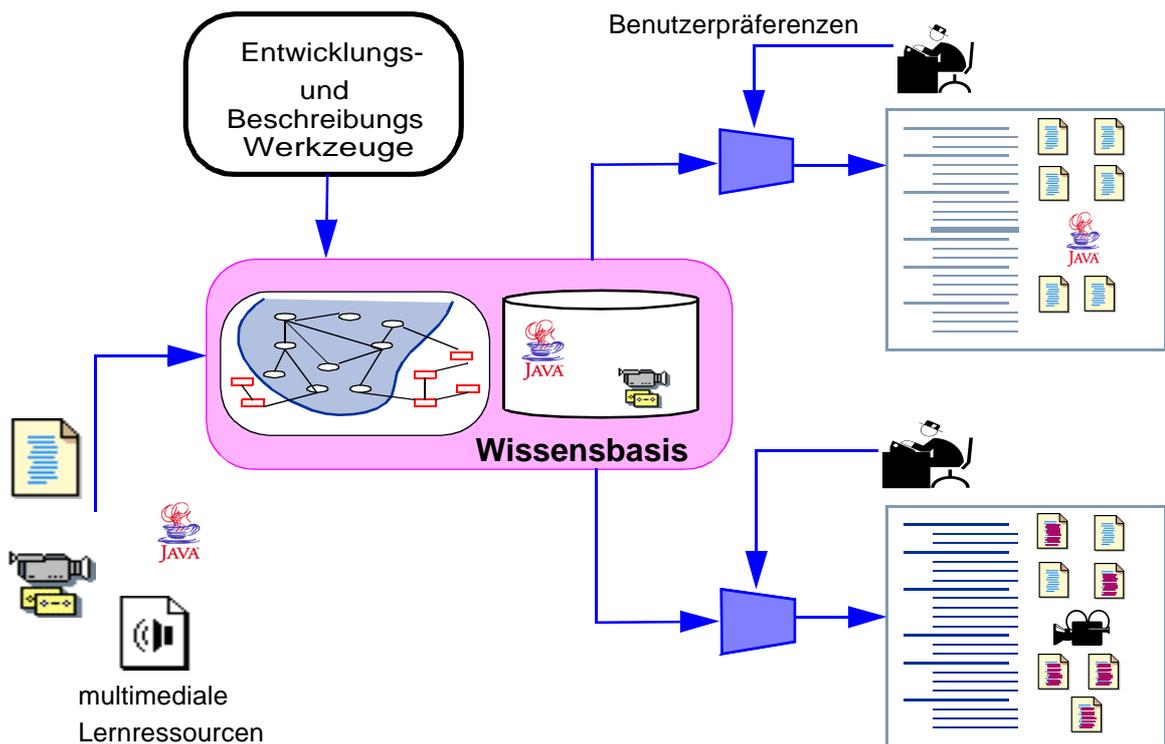


Abbildung 24: Architektur eines Lernsystems

Als erstes der im Rahmen dieser Arbeit entwickelten Werkzeuge soll in nächsten Kapitel der LOM-Editor beschrieben werden. Mit Hilfe des Editors ist es möglich LOM konforme Beschreibungen für multimediale Lernressourcen zu erstellen und in einem austauschbaren Format zu speichern. Der LOM-Editor wurde im Rahmen des Multibook-Projekts erstellt und wird mittlerweile in einer Vielzahl von anderen Projekten zur Beschreibung von Modulen eingesetzt. (Multibook [93], L3 [72], delite [31], OR-World [104], TeachwareOnDemand [133], Medibook [81], K-med [67]). Zusammen mit dem in Kapitel ZZ beschriebenen ConceptSpace-Editor stehen damit alle Werkzeuge zur Verfügung, die zum Aufbau einer Wissensbasis für web-basierte Lernsysteme benötigt werden. Der ConceptSpace-Editor ist ein Werkzeug zur Verwaltung eines semantischen Netzes, das zusätzlich die Erstellung einer Verbindung zwischen dem Begriff eines semantischen Netzes und einer LOM-Beschreibung ermöglicht. Da bestehende Werkzeuge zur Erstellung semantischer Netze, eine graphische Visualisierung entweder gar nicht, oder nur ungenügend umsetzen, werden in Kapitel 5 auch grundlegende Verfahren zur Informationsvisualisierung von Datenstrukturen beschrieben und untersucht, wie weit sie bei der Entwicklung eines Editors für die Verwaltung eines semantischen Netzes verwendet werden können.

4 Autoren- und Suchwerkzeuge für LOM

Ob modulares Lernmaterial in einem adaptiven Lernsystem verwendet wird und damit detaillierte Angaben über die Beziehungen zu anderen Bausteinen notwendig ist, oder als vollständiger Kurs eines virtuellen Studiengangs; den Modulen müssen LOM-Beschreibungen zugeordnet werden, die die benötigten Informationen über die Ressource enthalten. Obwohl, wie in Abschnitt 3.2 beschrieben, die Hälfte aller nötigen Einträge automatisch aus dem Material generiert werden kann, oder durch sich wiederholende Werte nicht mehr manuell eingegeben werden muß, bleibt eine große Zahl Metadatenelemente, deren Werte manuell eingegeben werden muß. Aus diesem Grund wurde im Rahmen dieser Arbeit ein Editor zur Erstellung von LOM-Beschreibungen entwickelt, der in Abschnitt 4.1 beschrieben ist. Als Programmiersprache wurde Java gewählt, sodaß der Editor auf beliebigen Plattformen ohne Änderungen lauffähig ist. Abschnitt 4.2 "Realisierung einer XML-basierten Suchmaschine für LOM" dokumentiert die Entwicklung einer web-basierten Suchmaschine zur Suche nach LOM-Beschreibungen.

4.1 LOM-Editor

4.1.1 Analyse und Design der Architektur

Die zunehmende Verbreitung von XML als universelles Datenaustauschformat und mittlerweile auch als Beschreibungsformat für Kommunikationsdienste, zeigt sich auch in der Verbreitung des Document Object Models (DOM) bei Implementierungen. Viele frei erhältliche XML-Parser in verschiedenen Programmiersprachen, bieten über eine DOM-API die Möglichkeit XML-Strukturen zu parsen. Da das DOM-Modell eine Struktur für beliebige Dokumente ist, wird die XML-Struktur in eine Baumform umgewandelt. Die Navigation auf diesen Strukturen ist langsam und komplex, da die DOM-Struktur einer XML-Beschreibung auch sämtliche Leerzeichen oder Zeilenumbrüche enthält, die aufgrund der besseren Lesbarkeit in jeder XML-Beschreibung vorhanden sind. Unabhängig von der Wahl der Programmiersprache, in der ein Editor für LOM-Beschreibungen implementiert wird, muß also eine Umsetzung der DOM-Repräsentation in ein Datenmodell der Programmiersprache vorgenommen werden.

Ähnlich wie bei der Entwicklung einer DTD für LOM (vgl. Abschnitt 3.3.2 "Document Type Definition für LOM"), lässt sich aus dem LOM-Draft direkt ein objektorientiertes Modell ableiten. Im Vergleich zu einer DOM-Struktur lassen sich bei einem objektorientierten Modell die Erweiterungsmöglichkeiten von LOM allerdings nur deutlich aufwendiger nachträglich implementieren. Bei einer XML-orientierten Struktur hingegen ist eine Erweiterung unabhängig von den Zugriffsmethoden, da der Typ der Elemente als Zeichenkette übergeben wird und daher im Zweifel ignoriert werden kann.

Eine weitere Randbedingung, die bei der Entwicklung des Editors berücksichtigt werden muß ist, daß neben einer Speicherung einer LOM-Beschreibung als XML-Datei, auch die Speicherung der Beschreibungen in einer Datenbank erforderlich ist. Erst durch den Einsatz einer Datenbank, kann eine Wissensbasis aufgebaut werden, die eine effiziente Verwaltung und eine Suche innerhalb von LOM-Beschreibungen ermöglicht. Neben der Verwendung einer relatio-

nen Datenbank, können zur Speicherung auch objektorientierte- oder XML-Datenbanken eingesetzt werden. Um eine möglichst breite Verwendung des Editors zu gewährleisten muß das verwendete Datenmodell daher vollständig unabhängig vom Typ der Datenbank sein, sodaß beliebige Datenbanken zur Speicherung der LOM-Beschreibungen eingesetzt werden können. Abbildung 25 zeigt wie diese Unabhängigkeit durch die Einführung einer Persistenzschicht, die eine Transformation des Objektmodells in eine geeignete Repräsentation zur dauerhaften Speicherung der LOM-Beschreibungen realisiert, erreicht werden kann.

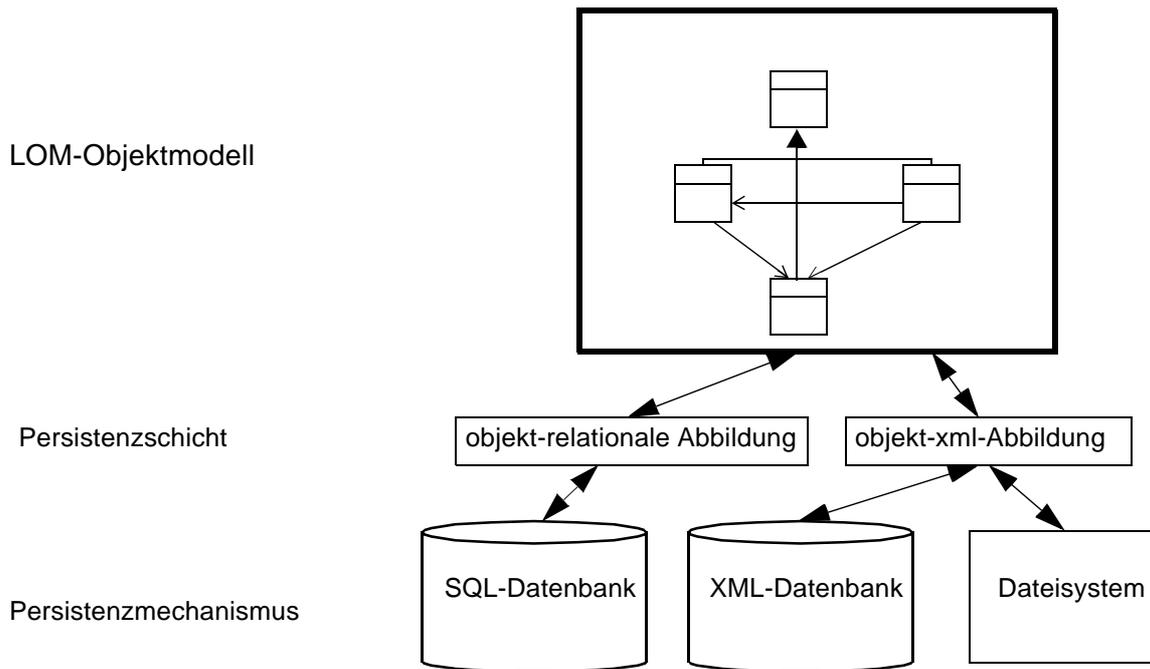


Abbildung 25: Abbildung des Objektmodells auf verschiedene Persistenzmechanismen

Für die Realisierung der Persistenzschicht kann auf frei erhältliche Implementierungen zurückgegriffen werden. Für die Programmiersprache Java existiert das Framework Castor, das eine Abbildung eines Java Objektmodells auf verschiedene relationale Datenbanken ermöglicht [22].

4.1.2 Verwendung vorgegebener Vokabulare

Wie beschrieben, erlaubt LOM für bestimmte Metadatenelemente keine freie Texteingabe, sondern spezifiziert ein festes Vokabular als mögliche Werte für die einzelnen Metadatenelemente. Steht für den Autor einer Metadatenbeschreibung nur eine feste Menge an möglichen Werten zur Verfügung, verringert sich die Gefahr, daß der Autor, der den Wert festlegt und ein Benutzer, der nach Ressourcen sucht, die gleiche Eigenschaft auf unterschiedliche Weise beschreiben. Ist dies der Fall, kann eine Ressource nicht gefunden werden. Ein weiterer Vorteil eines festen Vokabulars ist, daß die Menge an möglichen Werten einem Autor Hinweise darüber geben kann, wie eine Eigenschaft eines bestimmten Metadatenelements überhaupt

beschrieben werden kann. Durch die sprachenunabhängige Definition von LOM, werden als Werte für Vokabulare nur Zahlenwerte und keine Zeichenketten spezifiziert. Bei der Beschreibung einer Ressource kann aber nicht erwartet werden, daß ein Autor die Bedeutung der jeweiligen Werte kennt und daher bei der Erstellung der Beschreibung angeben kann. Das gleiche Problem stellt sich auch bei der Suche nach Ressourcen dar. Aus diesem Grund ist zusätzlich zur Festlegung der Zahlenwerte eines Vokabular-Elements, im LOM-Entwurf eine englische Beschreibung des Wertes mit spezifiziert. Die Übersetzungen des LOM-Entwurfs in verschiedenen andere Sprachen legen analog eine Abbildung auf Beschreibungen in der jeweiligen Sprache fest¹. Tabelle 3 zeigt dies anhand des *Structure*-Elements der General-Kategorie. In der Tabelle sind sowohl die englischen, als auch die deutschen Bezeichnungen für die möglichen Werte dieses Elements angegeben.

Wert	englisch	deutsch
0	Collection	Sammlung
1	Mixed	Gemischt
2	Linear	Linea
3	Hierarchical	Hierarchisch
4	Networked	Vernetzt
5	Branched	Verzweigt
6	Parceled	Aufgeteilt
7	Atomic	Atomar

Tabelle 3 Abbildung von Vokabularwerten auf Textbeschreibungen

Eine weitere Eigenschaft der Vokabulare bei LOM ist es, daß für bestimmte Elemente eine abgeschlossene Menge an Werten zur Verfügung steht, während andere Elemente durch benutzerdefinierte Werte, die zusätzlich zu den von LOM vorgeschlagenen Werten verwendet werden können. Beispielsweise sind die in Tabelle 3 gezeigten Werte des *Structure*-Elementes als festes Vokabular vorgegeben, während das Metadatenelement *Requirements.Type* der Technical-Kategorie beliebig erweitert werden kann. Will man diese Eigenschaft im LOM-Editor umsetzen, so bedeutet dies, daß die Menge an Werten für Vokabulare nicht fest im Editor implementiert werden kann, sondern über eine Konfigurationsdatei eingelesen werden muß. Durch Änderung der Konfigurationsdatei können dann für einzelne Elemente zusätzliche Werte spezifiziert werden. Ein Auszug der Konfigurationsdatei, die vom LOM-Editor verwendet wird, zeigt Abbildung 26

1. Ein Auszug aus der deutschen Übersetzung des LOM-Entwurfs findet sich im Anhang B; Übersetzungen in andere Sprachen (incl. der deutschen Version) unter [25]

```
# Syntax of the file:
# empty lines and comments (line starting with #) are ignored
# The beginning of a Section is identified with the path of the vocabulary
# through the categories in capitalized letters and separated with ".".
# Language sections of a vocabulary are identified with a line "lang='language'"
# End of a section is identified with "END 'Path for the Vocabulary'"
# While all languages of a vocabulary must appear together, there is no order for the categories
#
GENERAL.STRUCTURE restricted=true
lang=en
0=Collection
1=Mixed
2=Linear
3=Hierarchical
4=Networked
5=Branched
6=Parceled
7=Atomic
#
lang=de
0=Sammlung
1=Gemischt
2=Linear
3=Hierarchisch
4=Vernetzt
5=Verzweigt
6=Aufgeteilt
7=Atomar
#
lang=sp
0=coleccion
1=mixta
2=lineal
3=jerarquica
4=en redes
5=ramificada
6=empaquetada
7=atomica
END GENERAL.STRUCTURE
```

Abbildung 26: Auszug aus der Konfigurationsdatei für Vokabularwerte

Wie die Abbildung zeigt, ist die Konfigurationsdatei vollständig textorientiert und kann daher mit jedem beliebigen Text-Editor geändert werden. Die Abbildung zeigt auch, daß Bezeichnungen für Werte in beliebigen Sprachen in der Konfigurationsdatei spezifiziert werden können und beim nächsten Start des Editors unmittelbar zur Verfügung stehen. Bestehende, erweiterbare Vokabulare können um neue Werte ergänzt werden indem ein eindeutiger Zahlenwert zum entsprechenden Abschnitt hinzugefügt wird und eine Beschreibung spezifiziert wird. Beim Start des Editors wird die Konfigurationsdatei eingelesen und dem Benutzer in Form einer Auswahlliste bei den jeweiligen Elementen zur Verfügung gestellt, die je nach eingestellter Sprache, die möglichen Werte zeigt. Der Wert kann gelöscht werden, indem aus der Auswahlliste der leere Eintrag gewählt wird. Abbildung 27 zeigt die Präsentation der Werte für das *Structure*-Element in der Oberfläche des Editors

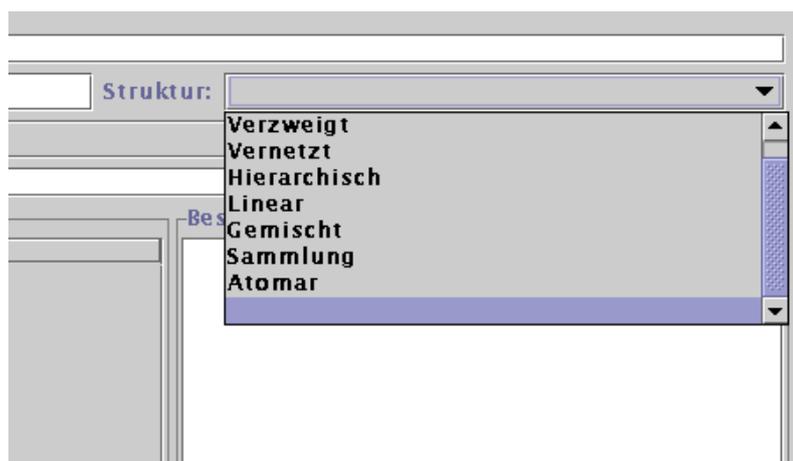


Abbildung 27: Anzeige der Vokabulare im Editor

4.1.3 Oberfläche des Editors

Nach dem Start des Editors, wird das Hauptfenster mit einer leeren LOM-Beschreibung geöffnet. Die verschiedenen LOM-Kategorien sind als Karteikarten angeordnet. Soweit ein Textfeld zur Verfügung steht, kann der Wert der LOM-Beschreibung direkt eingegeben werden. Können mehrere Werte eingegeben werden, so kann man über die bei dem Element angeordneten Neu- bzw. Löschen-Buttons neue Einträge erzeugen bzw. bestehende Einträge entfernen. Wie aus Abbildung 28 ersichtlich ist, die die Benutzeroberfläche des Editors mit der in Abbildung 14 gezeigten LOM-Beschreibung zeigt, werden nicht alle möglichen Elemente einer LOM-Beschreibung in der Benutzeroberfläche des Editors angezeigt. So werden bsp. die zu einem Vokabular gehörenden *Detail*-Elemente nicht in der Oberfläche angezeigt, da sie in der Regel nicht für jede Ressource erneut eingegeben werden müssen. Über das Datei-Menü können die in XX beschriebenen Schablonen eingelesen werden und entsprechend auch Beschreibungen, die im Editor angezeigt werden, als XML-Datei exportiert werden. Die beschränkte Auswahl an Elementen einer LOM-Beschreibung bedeutet also nicht, daß Werte, die nicht im Editor angezeigt werden, nicht verwendet werden können. Da als XML-Datei exportierte Beschreibungen mit jedem beliebigen textbasierten Editor nachträglich geändert oder neu erstellt werden können. Es kann also problemlos eine erste Version einer Beschreibung ohne Hilfe des Editors erzeugt werden. Diese Version kann dann in den Editor importiert werden. In der Oberfläche werden dann zwar nicht alle Elemente angezeigt; die Werte selbst werden aber nicht geändert oder gelöscht und finden bleiben entsprechend auch in einer durch den Editor geänderten und neu exportierten Version der Beschreibung erhalten.

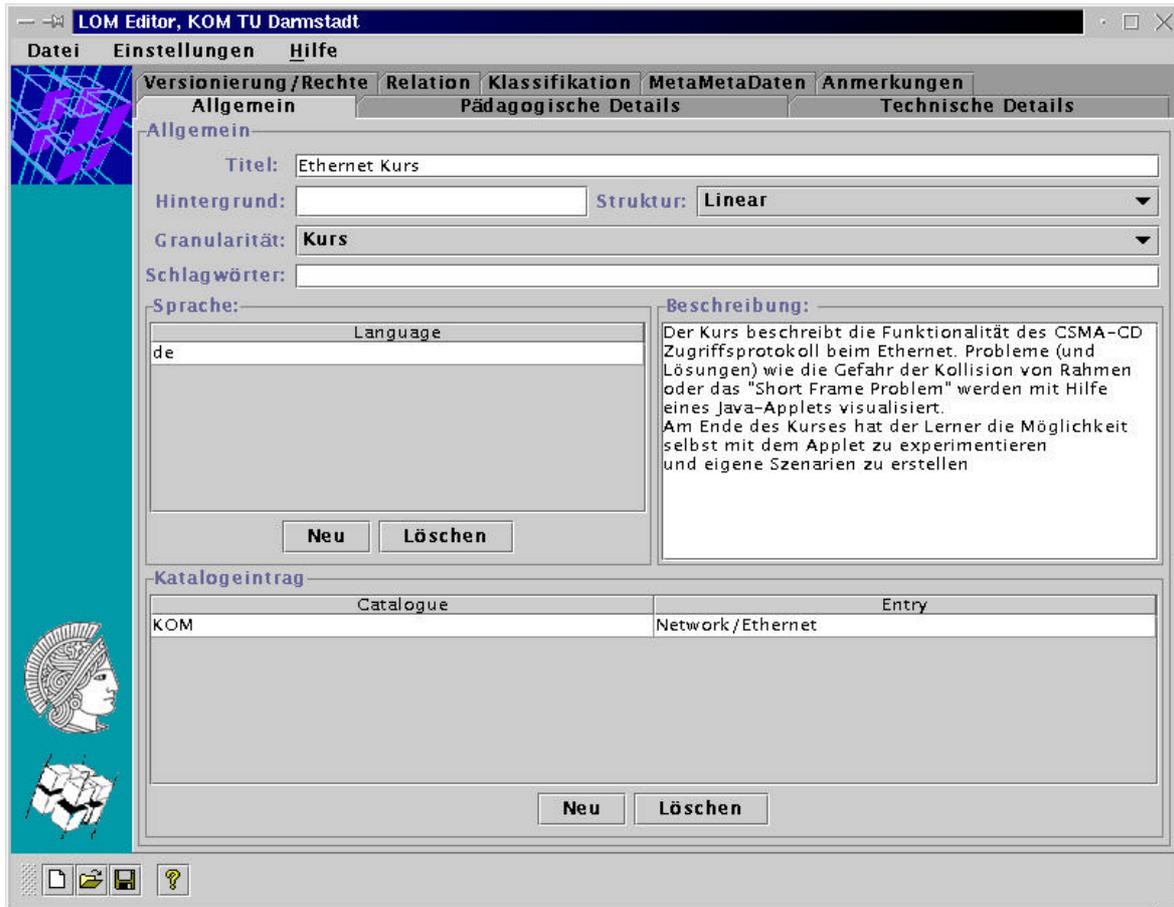


Abbildung 28: Benutzeroberfläche des Editors

4.2 Realisierung einer XML-basierten Suchmaschine für LOM

In diesem Abschnitt wird die Realisierung einer webbasierten Suchmaschine für Lernressourcen, die mit LOM ausgezeichnet sind, beschrieben. Als Datenbank wird eine XML-Datenbank verwendet, da damit ein direkter Export der LOM-Beschreibungen als XML-Datei erfolgen kann. Der LOM-Editor verfügt, wie oben beschrieben, neben der Anbindung an relationale Datenbanken über das Middleware-Konzept auch über die Möglichkeit die LOM-Beschreibungen in einer XML-basierten Datenbank zu speichern. Die Implementierung einer webbasierten Suchmaschine kann folgendermaßen strukturiert werden:

- 1 Der Client soll mit Hilfe eines Browsers befähigt werden, Suchanfragen an das Repository mittels HTTP-Requests an einen verarbeitenden Webserver zu übertragen.
- 1 Der Webserver muß die Suchanfrage an die Datenbank, die die LOM-Beschreibungen speichert, weiterleiten. Der Datenbankzugriff durch den Webserver muß hierbei das Kommunikationsprotokoll der Datenbank verwenden. Sofern dieses nicht HTTP-basiert ist, muß eine entsprechende Konvertierung des HTTP-Requests erfolgen, vorzugsweise durch ein java-basiertes Servlet, da damit die Suchmaschine auf jeder Plattform lauffähig

ist.

- 1 Die XML-basierte Datenbank liefert einen XML-Datenstrom an den Webserver zurück. Dieser Datenstrom muß - ebenfalls durch ein Servlet - nach HTML konvertiert werden, damit eine Darstellung der Suchergebnisse im Browser des Clients möglich ist.
- 1 Die HTML-Seite - mit einer Auflistung oder detaillierten Darstellung der Suchergebnisse - muß einen Verweis auf das eigentliche Lernobjekt enthalten. Durch diese Referenz wird der Client in die Lage versetzt, auf den entsprechenden Dokumentenserver zuzugreifen und sich das gewünschte Lernobjekt zu verschaffen.

4.2.1 Auswahl der Datenbank

XML-Dokumente lassen sich nach [13] in zwei Kategorien untergliedern. Diese werden mit den Begriffen "daten-orientiert" bzw. "dokumenten-orientiert" bezeichnet. Dabei werden als daten-orientierte XML-Dokumente Dokumente bezeichnet, bei denen XML hauptsächlich zum Datentransport verwendet wird und deren physikalische Struktur einen regelmäßigen Charakter aufweist. Im Gegensatz hierzu zeichnen sich die dokumenten-orientierte Dokumente durch eine unregelmäßige Struktur sowie gemischte Inhaltselemente aus. Zu dieser Dokumentenklasse gehören beispielsweise statische Web-Seiten. XML-Dokumente, die eine LOM-Beschreibung beinhalten sind demzufolge den daten-orientierten Dokumenten zuzuordnen. Ebenfalls in findet sich eine Kategorisierung existierender XML fähiger Datenbankprodukte und eine Beurteilung welches Produkt sich für welchen XML-Dokumententyp eignet. Aufgrund der dort aufgestellten Kriterien, eignet sich für die Speicherung von LOM-Metadaten eine native XML-Datenbank oder ein XML Server. Dabei unterscheidet sich eine XML-Datenbank von einem XML-Server nur insoweit, daß ein XML-Server neben dem reinen Datenmanagement noch weitere Dienste, anbietet, die aber für die Erbringung der Basisfunktionalität einer Suchmaschine nicht zwingend erforderlich sind. Für die Implementierung wurde daher der XML Server Tamino der Software AG ausgewählt der eine XQL-Schnittstelle beinhaltet und frei zur Verfügung gestellt wurde. Mit Tamino - der „Transaktions-Architektur zum Management von Internet-Objekten“ - wurde von der Software AG der weltweit erste Informationsserver entwickelt, bei dem Speicherung und Retrieval von Daten auf reinem, nativem XML-Format basieren. Kern der Datenbank ist die X-Machine, über die Speicherung und Aufruf von XML-Datenstrukturen aus dem XML-Speicher erfolgen. Diese Funktionen der X-Machine basieren auf den zuvor von einem Administrator im Data Map definierten Schemata. Das sogenannte Hybride Mapping ermöglicht es sogar die Speicherung von XML-Objekten teils in internen, teils in externen Datenspeichern zu realisieren. Hierzu wird die integrierte Schnittstelle X-Node benötigt, die den Zugriff auf externe Daten aus heterogenen Datenquellen, wie z.B. relationalen DBMS, ermöglicht. Darüber hinaus verfügt Tamino selbst über einen internen SQL-Speicher.

Die Definition eines Datenbankschemas erfolgt mit speziellen Schemasprache und wird mit Hilfe eines Schema Editors durchgeführt. Sie ist in der verwendeten Tamino Version 1.2.1.4 als DTD implementiert, die folgende Bestandteile aufweist:

```
<!ELEMENT Collection (Doctype*)>
<!ELEMENT Doctype (Node+)>
<!ELEMENT Node EMPTY>
```

Ein Datenbankschema wird durch die Vergabe von Attributwerten für diese Elemente defi-

niert. Eine Collection bezieht sich auf eine Datenbank und beinhaltet keinen oder mehrere Doctypes. Collections werden verwendet, um verschiedene Doctypes miteinander zu verbinden, so daß eine Datenbank parallel auf mehrere Doctypes zugreifen kann. Für die LOM-Datenbank wird eine entsprechende LOM Collection erzeugt. Diese besteht derzeit nur aus einem Doctype, dem RECORD. Ein Doctype stellt eine Datendefinition dar, die mit den Tabellen eines RDBMS zu vergleichen ist und einen oder mehrere Node-Elemente beinhaltet. Diese Node-Elemente selbst sind leer und drücken alle Informationen aus, die im Doctype enthalten sind, z.B. GENERAL, LIFECYCLE, RIGHTS etc. Die Node-Elemente können in einer Baumstruktur angeordnet werden, so daß Eltern-Kind-Beziehungen entstehen können. Hieraus resultiert ein ähnlicher Baum wie in Abbildung 29.

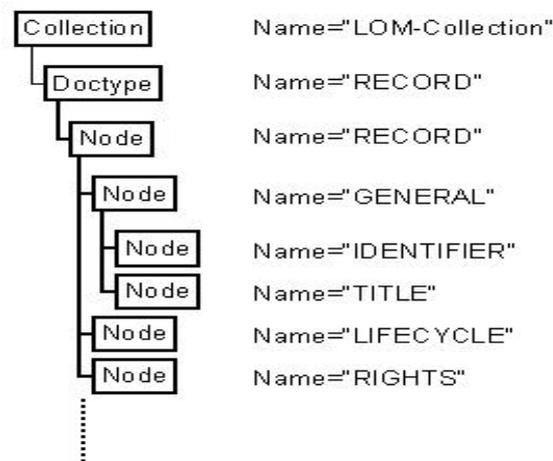


Abbildung 29: Collection-, DocType. und Node-Elemente des LOM Schemas

Die bislang als Branches bezeichneten Knoten werden in der Tamino Terminologie mit Intermediate Nodes bezeichnet. Die Leaves finden ihre Entsprechung in den Terminal Nodes. Die Beziehungen zwischen Root, Intermediate Nodes und Terminal Nodes werden mit Hilfe von Attributen ausgedrückt so daß für den Knoten GENERAL des LOM Datenbankschemas auszugswise folgende Informationen im XML-Format gespeichert werden:

```
<ino:node
  ino:name="GENERAL"
  ino:key="RECORD25"
  ino:obj-type="SEQ"
  ino:parent="RECORD2"
  ino:multiplicity="?"
  ino:search-type="no"
  ino:map-type="No"
/>
```

In diesem Fall bezeichnet ino:parent="RECORD2" den Knoten, der das RECORD Element der LOM-Struktur beinhaltet. Der Aufwand für die Generierung des Datenschemas ist nahezu null. Durch Einlesen der LOM DTD ist die Datenbank in der Lage, auf Basis der in der DTD deklarierten Strukturen, das zuvor beschriebene Schema im XML-Format, das die Relationen von Root, Intermediate Nodes und Terminal Nodes beschreibt, zu generieren. Einzig die Fest-

legung der Collection bzw. Doctype Attribute muß von Hand erfolgen. Im Anschluß an die Definition des Schemas kann dieses direkt in die LOM Datenbank geladen werden. Alle Dienste, die die Datenbank anbietet sind als HTTP-Requests spezifiziert, so daß der Zugriff auf die Datenbank mittels einer URL-Adresse erfolgen kann. Ein gültiger URL besteht hierbei aus der Adresse der Datenbank bzw. Collection, einem Befehlspräfix (z.B. ?_process, ?_xql) und - im Falle einer Query - einem abschließendem XQL-String. Taminos XQL Unterstützung basiert auf einem Subset der Empfehlungen des W3C [W3C2001f] und enthält zudem einige proprietäre Entwicklungen. Eine gültige Datenbankanfrage über den URL hat dann z.B. folgendes Format:

```
http://localhost/tamino/LOM-database/?_xql=RECORD[GENERAL/TITLE/LANGSTRING="Multimedia"]
```

Zur Einschränkung der Suchergebnisse kann der boolesche Operator AND genutzt werden. Dieser wird in XQL mit dem Ausdruck \$AND\$ symbolisiert. Zur Ausweitung der Suchoperation dient der boolesche ODER Operator, der entsprechend durch \$OR\$ dargestellt wird. Bei erfolgreicher Suche, liefert Tamino ein XML-Dokument zurück, die der in der folgenden Abbildung 30 gezeigten Struktur entspricht.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <ino:response xmlns:ino="http://namespaces.softwareag.com/tamino/response"
  xmlns:xql="http://metalab.unc.edu/xql/">
  <xql:query>RECORD[@ino:id='1']</xql:query>
  - <ino:message ino:returnValue="0">
    <ino:messageline>XQL Request processing</ino:messageline>
  </ino:message>
  - <xql:result>
    - <RECORD ino:id="1">
      + <GENERAL>
      + <METAMETADATA>
      + <LIFECYCLE>
      + <TECHNICAL>
      + <EDUCATIONAL>
      + <RIGHTS>
      + <RELATION>
      + <RELATION>
      + <RELATION>
      + <RELATION>
      + <ANNOTATION>
      + <CLASSIFICATION>
    </RECORD>
  </xql:result>
  - <ino:message ino:returnValue="0">
    <ino:messageline>XQL Request processed</ino:messageline>
  </ino:message>
</ino:response>
```

Abbildung 30: Ergebnis einer Suchanfrage

Die XQL-Anfragen erweisen sich als sehr komplex, da sie mit den zuvor genannten XPath Ausdrücken große Ähnlichkeit aufweisen. Der Benutzer ist daher gezwungen, die Struktur der gespeicherten XML-Instanzen genau zu kennen, um auf die von ihm gesuchten Elemente gezielt zugreifen zu können.

4.2.2 Suchoberfläche

Für das Design der Benutzeroberfläche muß es unerheblich sein, in welcher Art von Datenbank die LOM-Datensätze gespeichert werden. Ein webbasierter Client soll daher per HTTP-Request Suchanfragen an einen WebServer stellen können. Sofern die Anfrage vom Client nicht direkt als XQL-Anforderung übermittelt wird, muß sie vom Server noch entsprechend transformiert werden, bevor die Datenbank kontaktiert werden kann. Zudem kann der Server noch zusätzliche Dienste, wie bsp. eine Authentifizierung des Benutzers oder das Verwalten von Suchprofilen bestimmter Nutzer, realisieren. Wird eine XQL-Anforderung an den Server geschickt, so kann sie direkt an die Datenbank weitergeleitet werden. Für die Implementierung wurde die Variante gewählt, die beim Client direkt die XQL-Anfrage erzeugt, da diese den Vorteil bietet, daß neben der Generierung auch die Überprüfung des HTTP-Requests direkt beim Client durchgeführt werden kann. In diesem Kontext bedeutet Überprüfung, daß die eingegebenen Werte zu den erwarteten Werten konform sein müssen, so muß beispielsweise eine gültige Email-Adresse auf jeden Fall ein “@”-Zeichen enthalten.

Bei der serverseitigen Überprüfung dagegen müssen die Daten des Formulars zunächst zum Server übertragen und dort verifiziert werden. Im Falle einer ungültigen Eingabe muß dann vom Server eine entsprechende Aufforderung zur Eingabewiederholung an den Client zurückgeschickt werden. Dies bedeutet ein erhöhtes Datenaufkommen und längere Wartezeiten. Die Implementierung stellt daher das in Abbildung 31 gezeigte Suchformular für Learning Object Metadata zur Verfügung. Die HTML-Webseite enthält einen für die Auswertung der Formulareingaben verantwortlichen Java Script Anteil, der zum besseren Caching auf Clientseite in einer separaten Datei ausgegliedert ist und daher in der eigentlichen HTML-Datei nur referenziert wird.

The screenshot shows a web form for searching LOM (Learning Object Metadata) records. The form is organized into sections, each with a black header bar and a yellow background for the input fields:

- 1 General**: Contains fields for 1.1 Identifier, 1.2 Title, 1.3 CatalogEntry (Multi) (highlighted), 1.4 Language, 1.5 Description, 1.6 Keywords, 1.7 Coverage, 1.8 Structure (a dropdown menu with options: (0) Collection, (1) Mixed, (2) Linear), and 1.9 Aggregation Level (checkboxes for (0) atom, (1) collection of atoms, (2) collection of Level1, (3) course).
- 2 LifeCycle**
- 4 Technical**
- 5 Educational**
- 6 Rights**
- 7 Relation**
- 8 Annotation**
- 9 Classification**

At the bottom of the form, there are two buttons: "Send To Servlet" and "Reset Search Form".

Abbildung 31: Front-End der Suchmaschine

Dieses Java Script erzeugt einen HTTP-Request, der zunächst an den Server geschickt wird. Bestandteil des HTTP-Request ist die URL der Tamino Datenbank, der XQL-String, so wie - im weiteren noch näher zu erörtern - die Angabe eines entsprechenden Stylesheets:

```

databaseString=( "http://localhost/tamino/LOM-database" );
xqlString=( "RECORD[" + SUBSearchString + "]"
stylesheetString=( "<?xml-stylesheet href=\"http://localhost/lom/LOM -Stylesheet_Overview.xsl\"
type=\"text/xsl\"?>" );

```

Die Variable SUBSearchString beinhaltet hierbei einen XQL-konformen Suchbegriff für die LOM-Datenstruktur der XML-Datenbank. Obwohl der Client also eine Tamino konforme Anfrage als HTTP-Request an den Server schickt, wird die Anfrage nicht direkt weitergeleitet sondern, an einen Server-Prozeß gesendet, der als Java Servlet implementiert ist. Der Grund hierfür ist die Aufbereitung des Ergebnis der Suchanfrage. Wie beschrieben, kodiert Tamino die LOM-Datensätze, die zur Anfrage passen, als XML-Struktur, die in einem Datenstrom direkt an den Client zurückgesendet werden. Damit die Suchanfrage für den Benutzer nicht aussieht, wie in Abbildung 30, muß sie also noch bearbeitet werden. Prinzipiell steht dafür die im nächsten Abschnitt kurz eingeführte Extensible Style Sheet Language (XSL) zur Verfügung, mit Hilfe derer spezifiziert werden kann, wie eine XML-Struktur in HTML transformiert werden kann. Derzeit ist unter den gängigen Browsern aber nur der Internet Explorer in der

Version 5.5 in der Lage XSL-Transformationen durchzuführen. Um eine ansprechende Präsentation der Anfrage für Benutzer, die andere Browser verwenden, zu ermöglichen, muß der Antwortstrom der von der Datenbank gesendet wird also noch auf der Server Seite in HTML umgewandelt werden. Ein weiterer Grund für eine Bearbeitung der Antwort auf Server-Seite ist das von LOM verwendete Vokabular. Wie schon bei der Implementierung des LOM-Editor beschrieben, werden in LOM-Datensätzen für bestimmte Elemente nur, für einen menschlichen Benutzer wenig aussagekräftige, Integer-Werte gespeichert, die bei der Präsentation auf einen String-Werte in einer beliebigen Sprache abgebildet werden können. Da das Schema in der Tamino-Datenbank entsprechend nur Integer-Werte für die einzelnen Vokabulare vorsieht, muß die Abbildung vorgenommen werden, bevor das Suchergebnis an den Browser zurückgeschickt werden kann.

4.2.3 Server

Wie sich in der Erläuterung im vorherigen Abschnitt gezeigt hat, kann das Ergebnis der Suchanfrage nicht direkt an den Client weitergeleitet werden, sondern das Ergebnis muß vom Server noch wie folgt verändert werden:

- 1 Transformation der XML-Struktur nach HTML.
Diese Transformation beinhaltet nicht nur ein reines Umsetzen der XML-Elemente auf entsprechende HTML-Tags. Vielmehr sollte es aus Gründen der Übersichtlichkeit über diese Transformation bsp. auch möglich sein, bestimmte LOM-Elemente, nicht in die Präsentation mit aufzunehmen
- 1 Aufspalten des Datenstroms in mehrere unabhängige Einheiten.
Auch wenn durch LOM eine sehr viel detailliertere Suche auf Medienbausteinen möglich ist, als bei gängigen Suchmaschinen, kann das Ergebnis aus einer signifikanten Anzahl an LOM-Beschreibungen bestehen, die eine sehr lange und daher wenig komfortable HTML-Präsentation ergeben würde. Wird das Ergebnis in mehrere HTML-Seiten umgewandelt, die dem Benutzer sukzessive präsentiert werden, ist eine übersichtlichere Darstellung der Suchergebnisse möglich.
- 1 Abbildung der Vokabulare auf Zeichenketten
Analog zur Präsentation der Vokabulare als Zeichenketten im LOM-Editor (vgl. Abschnitt 4.1.2 "Verwendung vorgegebener Vokabulare"), muß auch bei der Transformation der Struktur nach HTML eine Umwandlung der Vokabulare in verschiedene Sprachen unterstützt werden

Für alle beschriebenen Transformationen, die innerhalb des XML-Dokuments benötigt werden, kann dieselbe Technologie verwendet werden, die vom W3C spezifizierte Extensible Style Sheet Language (XSL).

Extensible Style Sheet Language XSL

Der Vorteil von XML, die strikte Trennung von Inhalt und Struktur einerseits und Darstellung andererseits, impliziert, das Mechanismen zur Verfügung gestellt werden müssen, die die Definition von Formatierungsanweisungen für die Elemente eines XML-Dokumentes ermöglichen.

Mit der Extensible Stylesheet Language (XSL) stellt das W3C einen solchen Mechanismus zur

Definition von Stylesheets zur Verfügung XSL besteht aus zwei Teilen:

- 1 XSL Transformations (XSLT) ermöglichen die Transformation von XML-Dokumente.
- 1 Mit den XSL Formatting Objects (XSL-FO) wird ein Vokabular zur Verfügung gestellt, mit dessen Hilfe semantische Formatierungsanweisungen realisiert werden können.

Ein XSL Stylesheet spezifiziert die Präsentation einer Klasse von XML-Dokumenten. XSLT schreibt vor, wie eine Instanz dieser Klasse in ein neues XML-Dokument konvertiert werden kann, das das von den XSL-FO vorgegebene Vokabular verwendet und von einem entsprechenden Formatierer ausgewertet werden kann. XSL-Dokumente selbst sind wiederum als XML-Dokumente realisiert und enthalten eine Ansammlung von XSL-Elementen, die auf - durch XPath näher zu spezifizierende - Bereiche des XML-Dokumentes zugreifen

Eine Abkopplung der Transformation von der Präsentation erscheint aus verschiedenen Gründen sinnvoll. So müssen Transformationen von XML-Dokumenten nicht notwendigerweise zu Präsentationszwecken durchgeführt werden, z.B. können XML-Dokumente mit Hilfe von XSLT in eine andere Anwendungsdomäne transferiert werden. Außerdem existieren neben den XSL-FO weitere Beschreibungssprachen, mit denen die Präsentation beschrieben werden kann, so z.B. HTML und/oder CSS. Darüber hinaus sind die XSL-FO überaus komplex und nicht für die Verwendung im Web gedacht, vielmehr dienen sie der Layoutbeschreibung für die Druckmedien.

Ein XSLT-Prozeß benötigt zwei Dokumente, das zu transformierende XML-Dokument (Quelldokument) und ein Stylesheet mit entsprechenden Transformationsvorschriften. Das resultierende Zieldokument kann - muß aber nicht notwendigerweise - ein XML-Dokument sein. Der Transformationsprozeß traversiert das Quelldokument und beginnt bei der Dokumentwurzel, die jedoch nicht identisch ist mit dem obersten Element des XML-Dokumentes. Gemäß dem DOM ist die für die Transformation zu initialisierende Dokumentenwurzel ein Knoten, der nur einen einzigen Kindknoten vom Type Element Node besitzen kann.

Nach der Initialisierung des Dokumentes durchsucht der XSLT-Prozessor das Stylesheet nach einer Produktionsregel, welche die auf den jeweiligen Knoten anzuwendende Verarbeitungsanweisung beschreibt. Diese Beschreibung wird im folgenden als *Template Rule* bezeichnet. Mit Hilfe der Namespace Definition von XSLT zu Beginn des Stylesheets erkennt der XSLT-Prozessor, welche Elemente innerhalb des Stylesheets zur XSLT-Anwendungsdomäne gehören. Der standardmäßige Präfix für den XSLT-Namensraum ist hierfür *xsl*.

Der XSLT-Prozessor überprüft im folgenden, ob das Element zur seiner Anwendungsdomäne gehört, d.h. sobald ein Element den Präfix *xsl* aufweist, wertet der Prozessor das Element als XSLT-Konstrukt aus.

Wird ein Element des XSL-Stylesheet nicht als XSLT-Konstrukt erkannt, so ergeben sich nach [5] zwei Alternativen:

1. Das Element wird als zu einem anderen Namensraum, der für Erweiterungen innerhalb des Stylesheets deklariert wurde, zugehörig erkannt und an einen näher zu spezifizierenden Auswertungsprozeß weitergegeben.
2. Elemente ohne Namensraum-Deklaration werden unverändert in das Zieldokument übernommen.

Die folgende Tabelle gibt einen Überblick über die zur Verfügung stehenden XSL-Elemente: Die XSL-Elemente wählen in Verbindung mit einem Attribute (z.B. *select* oder *match*) und einem XPath-Ausdruck entsprechende Elemente des XML-Dokumentes aus und wenden auf diese die in der jeweiligen Template Rule definierte Verarbeitungsvorschrift an.

xsl:apply-templates	weist den XSL-Prozessor an, eine durch XPath näher spezifizierte Template Rule anzuwenden
xsl:attribute	erzeugt im Ausgangsdokument einen Attribute Node, der dem zuvor definierten Ausgabe-Element angehängt wird
xsl:choose	ermöglicht in Verbindung mit xsl:when und xsl:otherwise eine Fallunterscheidung
xsl:comment	erzeugt im Zieldokument einen Kommentar
xsl:copy	kopiert den ausgewählten Knoten in das Zieldokument
xsl:element	erzeugt ein neues Element im Ausgabedokument
xsl:for-each	ermöglicht die iterative Verarbeitung für gleichartige Knoten einer Ebene
xsl:if	dient der Überprüfung von näher zu spezifizierenden Bedingungen
xsl:otherwise	ist Unterelement von xsl:choose und ermöglicht die Fallunterscheidung
xsl:pi	erzeugt im Zieldokument eine Processing Instruction
xsl:stylesheet	steht zu Beginn eines Stylesheet und weist dieses als solches aus; beinhaltet eine Ansammlung von Template Rules
xsl:template	definiert die Template Rule für ein durch XPath definiertes Node Set
xsl:value-of	liest den Wert des durch XPath definierten Node Sets aus und übernimmt ihn ins Zieldokument
xsl:when	ist Unterelement von xsl:choose und ermöglicht die Fallunterscheidung

Tabelle 4: Übersicht der XSL-Elemente

Zum besseren Verständnis der XSL Transformation dient das folgende Stylesheet, das ein zur LOM-DTD konformes XML-Dokument in ein HTML-Dokument transformiert, welches von einem Formatierer (in diesem Fall ein Browser) dargestellt werden kann.

Ein XML-Dokument für die durch die in Abschnitt 3.3.2. definierte DTD sieht wie folgt aus:

```
<?xml version="1.0" ?>
<buch isbn-nummer="3-540-67332-6">
  <titel>Multimedia Technologie</titel>
  <information>
    <autor>Ralf Steinmetz</autor>
  </information>
  <kapitel>
    <ueberschrift>Multimediales Lernen</ueberschrift>
    <absatz>Dies ist ein Absatz.</absatz>
  </kapitel>
</buch>
```

Ein mögliches Stylesheet zur Generierung einer HTML-Webseite sieht folgendermaßen aus:

```

<?xml version="1.0"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <HTML>
      <HEAD><TITEL>Ein Buch</TITEL></HEAD>
      <BODY>
        <xsl:apply-templates select="buch"/>
      </BODY>
    </HTML>
  </xsl:template>
  <xsl:template match="buch">
    <H1><xsl:value-of select="child::title"/></H1>
    <H2>ISBN-Nummer: <xsl:value-of select="attribute::isbn-nummer"/></H2>
    <xsl:apply-templates select="child::information"/>
    <xsl:apply-templates select="child::kapitel"/>
  </xsl:template>
  <xsl:template match="information">
    <P>Autor: <xsl:value-of select="child::autor"/></P>
  </xsl:template>
  <xsl:template match="kapitel">
    <P>Kapitelüberschrift: <xsl:value-of select="child::ueberschrift"/><BR/>
    <xsl:value-of select="child::absatz"/></P>
  </xsl:template>
</xsl:stylesheet>

```

Das Zieldokument würde nach der Verarbeitung durch einen XSLT-Prozessor folgenden Inhalt vorweisen:

```

<HTML>
  <HEAD><TITEL>Ein Buch</TITEL></HEAD>
  <BODY>
    <H1>Multimedia-Technologie</H1>
    <H2>ISBN-Nummer: 3-540-67332-6</H2>
    <P>Autor: Ralf Steinmetz</P>
    <P>Kapitelüberschrift: Multimediales Lernen<BR/>
    Dies ist ein Absatz.</P>
  </BODY>
</HTML>

```

Zu beachten hierbei ist, daß die HTML-Elemente innerhalb des XSL-Stylesheets wohlgeformt sind, d.h. einen Start- und Endtag vorweisen. Aus diesem Grund wurde in dem vorausgegangenen Stylesheet statt
 ein
 Tag verwendet, eine verkürzte Schreibweise für
</BR>, die ein sogenanntes Empty Element markiert.

Cocoon

Cocoon wurde im Rahmen des Java-Apache-Projekts als Java Servlet zum XML-basierten Webpublishing entwickelt. Mittlerweile gibt es ein eigenes Apache-XML-Projekt mit dem Ziel, den Server XML-fähig zu machen. [1].

Cocoon ist als Bestandteil des Servers implementiert und basiert vollständig auf den W3C Standards. Als Servlet erstellt Cocoon anhand der Eingabeparameter eine komplette HTML-Seite, die es an den Browser des Clients ausliefern kann. Die serverinterne, für den Client

unsichtbare Verarbeitung läßt sich anhand der folgenden Grafik vereinfacht darstellen.

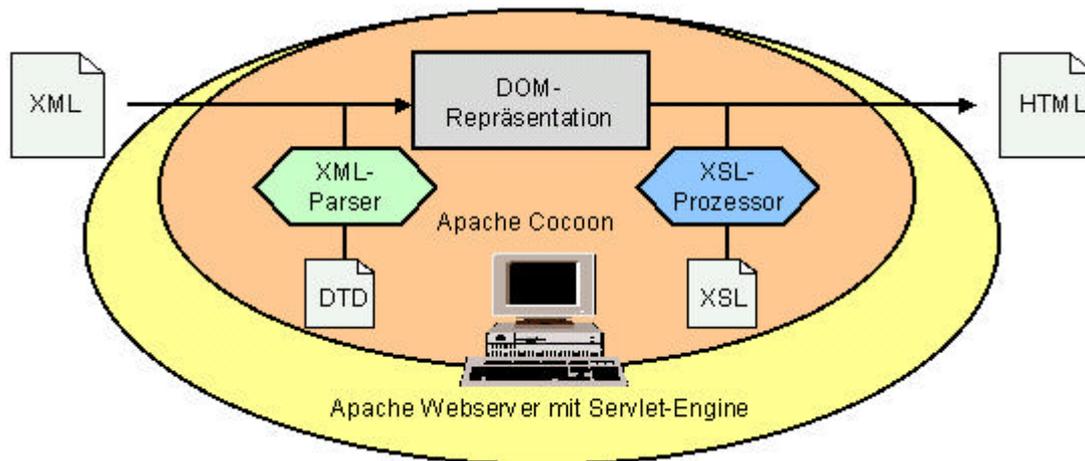


Abbildung 32: Vereinfachtes Verarbeitungsmodell von Cocoon ([87])

Ein XML-Dokument wird bei der internen Verarbeitung mittels des Parsers in eine DOM-Repräsentation überführt, das Vorhandensein einer DTD ist nicht zwingenderweise erforderlich. Der DOM-Baum dieser XML-Datei kann wie beschrieben, von einem XSL-Prozessor unter Angabe eines entsprechenden Stylesheets verarbeitet und in eine HTML-Datei umgewandelt werden. Darüber hinaus bietet Apache Cocoon die Möglichkeit, die verwendeten Komponenten (Parser und Prozessor) zu verändern. So kann z.B. als XSL Prozessor FOP eingesetzt werden. FOP bedeutet hierbei Formatting Objects to PDF, d.h. dieser Prozessor kann zur serverseitigen PDF-Erzeugung verwendet werden. Ein weiterer wichtiger Aspekt ist, daß Cocoon nicht nur auf dem Server gespeicherte und somit meist statische XML-Dokumente verarbeiten kann, sondern auch in der Lage ist, Zugriff auf die Tamino Datenbank zu nehmen und somit dynamische Inhalte zu generieren.

4.2.4 Vollständige Architektur der zu entwickelnden Applikation

Mit Hilfe von Cocoon, läßt sich die vollständige Architektur einer Suchmaschine für multimediale Lernressourcen wie in Abbildung 33 gezeigt abbilden. Der Client erzeugt mit Hilfe von Java Script aus dem Eingabeformular die für die Datenbankabfrage notwendigen Variablen *databaseString* und *xqlString* und übermittelt diese an den Webserver. Dort werden sie vom Apache Cocoon Servlet abgefangen und an die Tamino Datenbank weitergeleitet. Die zusätzlich vom Client übertragene Variable *styleSheetString* ist für Cocoon selbst bestimmt und dient dem Servlet zur Auswahl eines entsprechenden XSL-Stylesheets.

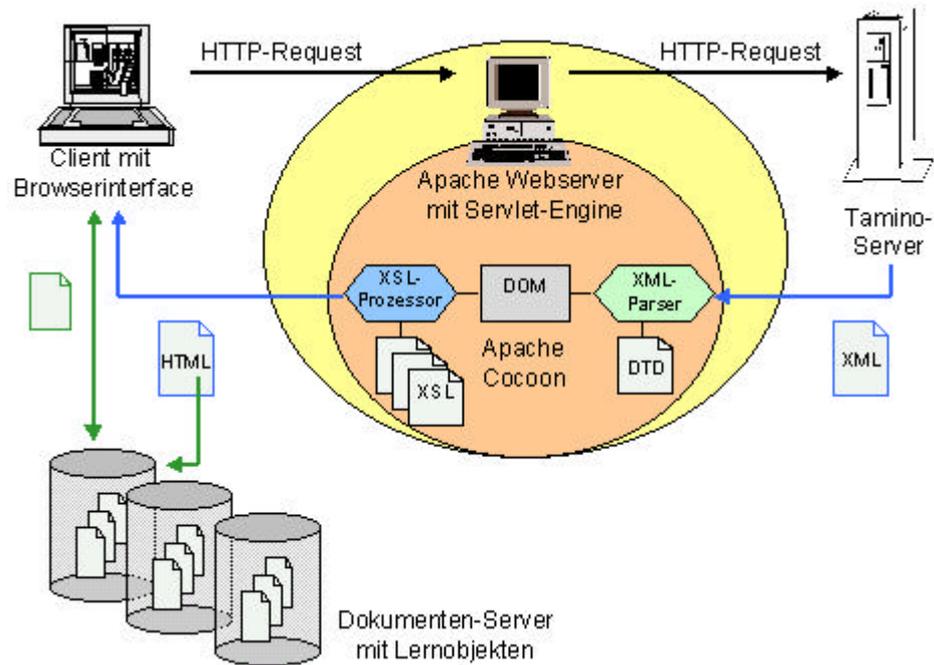


Abbildung 33: Architektur der Suchmaschine

Der von Tamino zurückgelieferte XML-Datenstrom wird ebenfalls vom Cocoon Servlet abgefangen und mit dem internen Parser in eine entsprechende DOM-Struktur umgewandelt. Im Anschluß hieran verwendet Cocoon das vom Client spezifizierte Stylesheet, um diese DOM-Struktur in HTML umzuwandeln. Bei der Suchanfrage des LOM-Formulars, weist der Client Cocoon an, das für die Erzeugung der Übersichtsseite zuständige Stylesheet zu verwenden. Dieses Stylesheet ist so gestaltet, daß die resultierende HTML-Seite nicht nur die eigentlichen Metadaten des XML-Dokumentes enthält, sondern auch einen Verweis auf das im XML-Datenstrom enthaltene Element *ino:id*. Dieses Element dient als Primärschlüssel einer Tamino Datenbank.

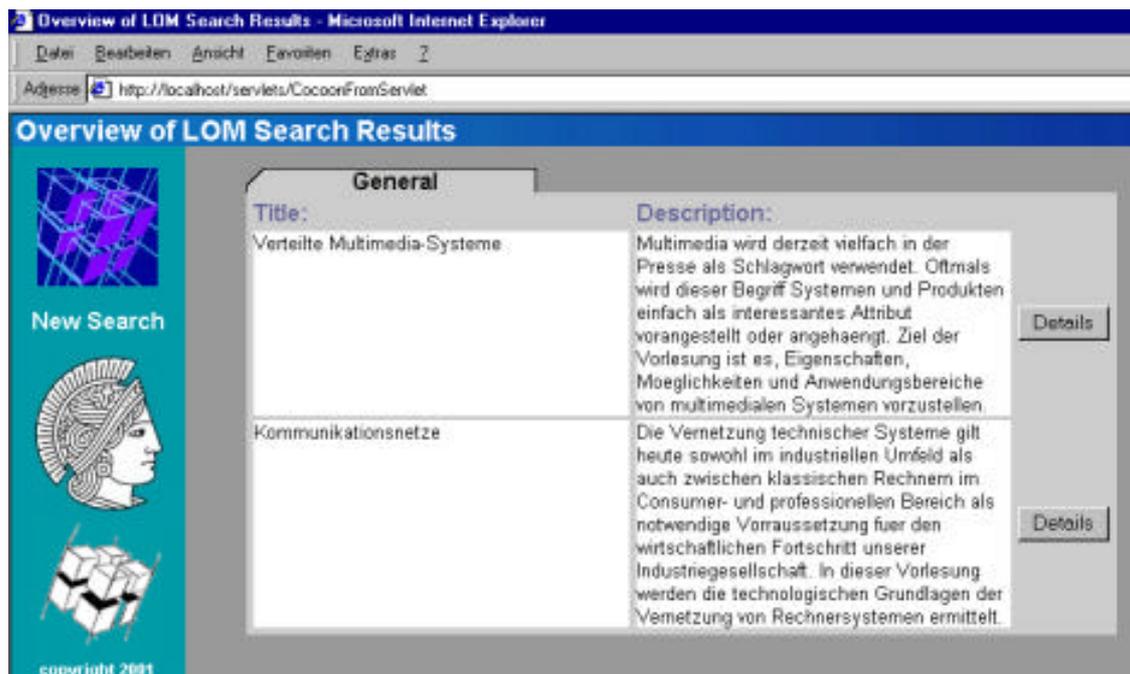


Abbildung 34: Übersichtsresultat einer Suchanfrage

Der Client kann nun aus dieser Übersicht einen für ihn relevanten Treffer auswählen. Sobald er den entsprechenden Details Button drückt, wird mittels der schon zuvor verwendeten Java Script Datei eine neue Datenbankabfrage gestartet. Der Client überträgt erneut die Variablen *databaseString* und *sqlString*. Der *sqlString* enthält nun die *ino:id*. Tamino kann dem Cocoon Servlet somit einen eindeutigen Treffer zurückliefern. Darüberhinaus weist der Client bei dieser zweiten Suchanfrage Cocoon an, ein anderes Stylesheet zu verwenden. Mit dessen Hilfe

kann Cocoon die detaillierte Ansicht eines Suchresultates erzeugen

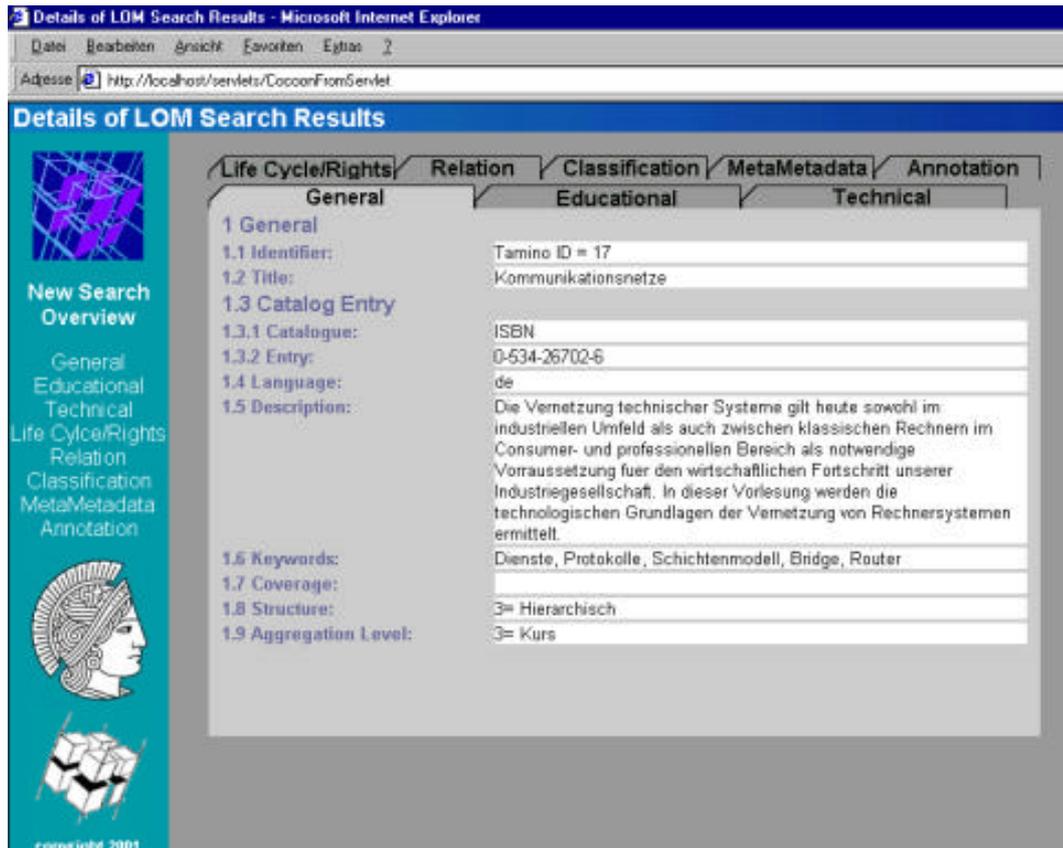


Abbildung 35: Detailansicht einer Suchanfrage

Die detaillierte Ansicht enthält einen Verweis auf die eigentliche Lernressource. Mit dieser URL-Adresse kann der Client auf das in einem Dokumentenserver gespeicherte Lernobjekt zugreifen.

5 Entwicklung eines Werkzeug zum Aufbau eines semantischen Netzes

5.1 Werkzeuge für semantische Netze

In [36] ist ein Rahmenwerk zur Beurteilung von Werkzeugen zur Erstellung und Verwaltung von semantischen Netzen beschrieben. Anhand der Kriterien Qualität der Benutzeroberfläche, Mächtigkeit der Schemata und Dienste, die ein kooperatives Erstellen eines semantischen Netzes ermöglichen, werden verschiedene Anwendungen untersucht und bewertet. Das Rahmenwerk ist in Tabelle 5 angegeben, die untersuchten Werkzeuge sind :

- 1 Ontolingua, WebOnto, Protege Win, OntoSaurus, ODE, KADS22 ([38], [33], [46], [62])

Allgemeine Kriterien	Semantisches Netz	Kollaborative Eigenschaften
Klarheit der Benutzeroberfläche	Mehrfachvererbung möglich	Können mehrere Benutzer gleichzeitig ein Netz editieren?
Konsistenz der Benutzeroberfläche	Konsistenzprüfung	Können Teile des Netz gesperrt werden?
Performanz	Komplexität bei der Verwendung von De-/Kompositionen	Kann auf gesperrten Teilen navigiert werden
Überblicksfunktionalität über das semantische Netz	Weiterverwendung bestehender Netze	Können Änderungen anderer Benutzer leicht erkannt werden?
Verständlichkeit der Befehle	Verfügbarkeit von Beispielnetzen	Kann das Netz exportiert werden?
Stabilität	Enthält das Hilfesystem Hinweise über die Verwendung von Begriffen?	Können bestehende Netze importiert werden?
lokale Installation	-	-
Funktionalität des Hilfe-Systems	-	-

Tabelle 5 Kriterien zum Vergleich von Ontologie-Editoren (vgl. [36])

Vor allem die allgemeinen Kriterien scheinen recht willkürlich gewählt und orientieren sich mehr an generellen Anforderungen des Software Engineering. So hat eine web-basierte Version eines Editors für semantische Netze offensichtliche Vorteile wie einfachere Wartung und leichtere Aktualisierung gegenüber einer Version, bei der eine lokale Installation erforderlich ist. Es lassen sich aber daraus keine Rückschlüsse für die Erstellung semantischer Netze ziehen. Auch die Kriterien Klarheit oder Konsistenz der Benutzeroberfläche sind sehr allgemein gehalten und machen keine Aussage darüber, ob eine Trennung der Funktionalität bezüglich der Anforderungen verschiedener Benutzer möglich ist. Eine Untersuchung, inwieweit ein Benutzer Kenntnis über das dem Werkzeug zugrundeliegende Datenmodell haben muss, findet zwar statt, schlägt sich aber nur indirekt in den Bewertungskriterien wieder. Duineveld et. al,

kommen zu dem Schluss, das sich Protege Win am ehesten für die Benutzung durch unerfahrene Benutzer eignet, da es eine klare Trennung zwischen Datenmodell und Sicht des Benutzers auf das Netz gibt. Dies wird erreicht durch die Aufteilung des Erstellungsprozess durch drei verschiedene Systemkomponenten. Zur Erstellung des Schemas wird ein Ontologie-Editor verwendet. Anhand dieses Schemas kann dann durch eine Layout-Editor genannte Komponente eine Oberfläche erzeugt werden, die die Eingabe von konkreten Begriffen und das Verbinden dieser Begriffe durch Relationen ermöglicht wird. Eine Exportmöglichkeit für das erzeugte Schema existiert nicht. Die dritte Layout-Interpreter genannte Komponente stellt die vom Layout-Editor generierte Oberfläche dar. In Form von Tabellen werden existierende Begriffe der Ontologie angezeigt, und der Benutzer kann durch das Hinzufügen neuer Zeilen zu bestimmten Tabellen neue Begriffe in das Netz einfügen. Eine graphische Sicht auf das bestehende Netz, oder ein Einfügen neuer Begriffe innerhalb einer graphischen Anzeige ist nicht möglich. Dazu muss gesagt werden, dass keines der getesteten Werkzeuge diese Möglichkeit anbietet. Überträgt man diese Situation auf Benutzerschnittstellen für relationale Datenbanken, so würde das bedeuten, dass nur Werkzeuge zur Verfügung stehen würden, die Benutzern die direkte Eingabe innerhalb der relationalen Datenstrukturen ermöglicht und die von den Benutzern erwarten, dass sie in der Lage sind, die zugrundeliegenden Zusammenhänge zwischen Einträgen verschiedener Spalten zu rekonstruieren, um zu erkennen, welche Einträge der entsprechenden Spalten in Zusammenhang stehen. Selbst bei einfachen relationalen Datenschema kann eine solche Leistung von einem Benutzer nicht erwartet werden. WebOnto ist das einzige Werkzeug, das in der Lage ist, ein Netz komplett zu visualisieren; ein Einfügen ist aber auch hier nicht möglich. Sobald ein Netz aber mehr als 20 Begriffe enthält, wird die Darstellung bei WebOnto zunehmend unübersichtlich. Dies ist darauf zurückzuführen, dass der Layout-Algorithmus Begriffe des Netzes, die als Knoten visualisiert werden, willkürlich auf der verfügbaren Oberfläche verteilt. Entsprechend müssen dann die zwischen den Begriffen bestehenden Beziehungen, die als Kanten zwischen den Knoten dargestellt werden, oftmals quer über die ganze Visualisierungsfläche - und demnach auch über andere Knoten hinweg - gezeichnet werden, was die Lesbarkeit des Netzes für einen Benutzer stark erschwert. Zudem ist der Algorithmus nicht in der Lage, nur Teile des Netzes zu visualisieren und eine interaktive Navigation anzubieten. Demzufolge überdecken sich Knoten auf der Visualisierungsfläche und machen die Visualisierung daher nahezu unbrauchbar. Andere Werkzeuge, wie beispielsweise Ontolingua sind nur in der Lage, die taxonomischen Teilstrukturen (soweit im Schema des Netzes eine Taxonomie enthalten ist) zu visualisieren, und geben daher einen verfälschten Eindruck des Gesamtnetzes wider. Während auf die Kollaborationseigenschaften der Werkzeuge hier nicht weiter eingegangen wird, kann man bei den letzten beiden Kriterien für die Leistungsfähigkeit des semantischen Netzes anmerken, dass damit nicht die Eigenschaften des Netzes beschrieben werden, sondern die Benutzerfreundlichkeit des Werkzeugs. Da es sich bei allen Editoren um Forschungsprojekte handelt, kann die verfügbare Hilfefunktionalität erwartungsgemäß kommerziellen Ansprüchen nicht genügen. Während die Eigenschaft der Mehrfachvererbung nur eine Aussage darüber macht, ob es sich bei dem zugrundeliegenden Schema um einen frame- oder einen regel-basierten Ansatz handelt, ist mit dem Kriterium "Weiterverwendbarkeit bestehender Netze" die Wiederverwendung eines Schemas gemeint und nicht die Weiterverwendung bestehender semantischer Netze, die selbstverständlich sein sollte. Der Punkt "Konsistenzprüfung" sollte eigentlich bei allen Werkzeugen eine Selbstverständlichkeit sein, vor allem Ontolingua weist dabei aber deutliche Schwächen auf.

5.1.1 Bewertung

Als eher ungeeignet für Anfänger werden die Werkzeuge Ontolingua und OntoSaurus bewertet, da bei diesen Editoren Kenntnisse der Datenmodellierungssprache LOOM zwingend für die Arbeit mit dem System erforderlich sind. Der Vergleich mit dem relationalen Datenbankmodell macht auch hier wieder die Limitierung dieses Ansatzes deutlich. Kenntnisse der Modellierungssprache bei der Eingabe von Daten für ein relationales Modell vorauszusetzen, würde bedeuten, dass einem Benutzer eine Schnittstelle angeboten wird, über die er SQL-Statements absetzen kann. Duinevald et. al. kommen zu dem Schluß, dass mit den bestehenden Werkzeugen nur Experten in Wissensmodellierung unterstützt wird, Wissensexperten jedoch nicht. Am ehesten kann noch Protege Win den Wissensexperten unterstützen, da hier die zugrundeliegende Struktur komplett vor ihm verborgen werden kann; die unzulängliche Visualisierung des Netzes erschweren aber auch hier die Verwendung. Für eine interaktive Navigation sind also alle Werkzeuge kaum geeignet.

Um sich in den Netzen zurechtzufinden, sind zumindest detaillierte Kenntnisse der modellierten Domänen erforderlich. Für einen Lerner, der durch die Arbeit mit dem Werkzeug eben genau dieses Wissen erwerben will, muss der Zugang zum semantischen Netz also in einer viel einfacheren und interaktiveren graphischen Art und Weise angeboten werden. Dies bedeutet, dass nur ein Experte in Wissensmodellierung Nutzen aus der Verwendung der Werkzeuge ziehen kann. Zum Aufbau einer Datenbasis für webbasierte Lernsysteme kann ein solcher Experte aber nur den ersten Schritt beitragen, indem er zusammen mit einem Wissensexperten ein Schema erstellt, auf das sich die Eigenschaften und Gegebenheiten des zu modellierenden Wissensgebiets möglichst vollständig und korrekt abbilden lassen. Für den Wissensexperten, der mit Hilfe des Schemas das semantische Netz baut und kein Experte in Wissensmodellierung ist, sind die Werkzeuge ungeeignet. Zudem unterstützen die untersuchten Werkzeuge nur die reine Erstellung des semantischen Netzes. Es ist nicht vorgesehen, daß einem Begriff im Netz die Metadatenbeschreibung eines Medienbausteins zugeordnet werden kann. Aus den in Abschnitt 3.8.4 "Werkzeuge zur Erstellung der Metadaten" beschriebenen allgemeinen Anforderungen an ein Werkzeug zum Erstellen semantischer Netze für die Verwendung in einem webbasierten Lernsystem können daher folgende konkrete Anforderungen für die Implementierung abgeleitet werden.

5.1.2 Anforderungen

- 1 Trennung der Schemaerstellung von der Erzeugung des semantischen Netz:
Diese Trennung ist nicht nur durch die unterschiedlichen Rollen bei der Erstellung eines semantischen Netzes motiviert. Sie ermöglicht es zudem, die Kodierung der Schemastruktur zu ändern, ohne dass der Benutzer über diese Änderungen informiert werden muss und sich für ihn der Erstellungsprozess ändert. Vor allem im Hinblick auf eine Wiederverwendung von Schemabeschreibungen, ist dieser Aspekt von Bedeutung. Wie in Kapitel 2.6.3 beschrieben, existiert derzeit zwar noch kein allgemein akzeptierter Standard für die Beschreibung eines Schemas, die Entwicklung von RDF Schema oder OIL zeigen aber deutlich den Bedarf für eine unabhängige Beschreibung.
- 1 Abstraktion von den verwendeten Datenstrukturen durch graphische Repräsentation des Netzes:
Unabhängig von der Kodierung des semantischen Netz, muß das Netz dem Benutzer in

einer konsistenten und von der Applikationsemantik unabhängigen Form präsentiert werden. Dies bedeutet das unter Verwendung der existierenden Modelle Navigationsfähigkeiten und Editoreigenschaften für die Rollen in web-basierten Systemen zur Verfügung gestellt werden müssen. Wie die Beurteilung der existierenden Werkzeuge zeigt, sollte den Benutzern entsprechend das semantische Netz als graphisches Netz angezeigt werden.

- 1 Anpassung der Oberfläche aufgrund des Schemas:
Ähnlich wie bei der Oberfläche von WinProtege darf nur die Menge der Begriffs- und Relationstypen in der Oberfläche angeboten werden, die für das ausgewählte Schema gültig sind.
- 1 Regeln und Axiome des Schemas müssen sich für den Benutzer in der Benutzeroberfläche des Editors widerspiegeln und damit nachvollziehbar sein:
Obwohl in der oben beschriebenen Evaluation, die Konsistenzprüfung des semantischen Netz noch als separates Kriterium aufgeführt wurde, ist dies nur die Grundvoraussetzung eines Editors. Viel wichtiger für eine komfortable Nutzung ist, daß der Editor in der Lage ist, aufgrund der Axiome und Regeln dem Benutzer Hinweise zu geben welche Operationen erlaubt sind. Dies kann nicht für alle Regeln, des Schemas geschehen, da es nicht zur Funktionalität der Visualisierungskomponente gehört, die Korrektheit des Schemas zu prüfen. Enthält das Schema jedoch einfache Regeln, wie beispielsweise die Angabe, dass zwischen zwei Begriffstypen keine Relation eines bestimmten Typs erzeugen lässt, so kann diese Regel dem Benutzer direkt verdeutlicht werden, indem ihm bei der Erstellung einer Beziehung zwischen zwei Begriffen nur die Relationstypen angeboten werden, die nicht zu einer Verletzung der Regeln führen.
- 1 Exportmöglichkeit des Netz zur Wiederverwendung:
Die Forderung nach einer Exportmöglichkeit bedingt nicht, daß für eine Implementierung konkrete Datenstrukturen vorgeschrieben werden. Es muß nur sichergestellt sein, daß die Möglichkeit besteht die verwendeten Datenstrukturen in ein wiederverwendbares Austauschformat wie bsp. RDF Schema oder Topic Maps realisiert werden kann.
- 1 Visualisierung des semantischen Netz und der Metadaten einzelner Medienbausteine:
Bisherige Werkzeuge ermöglichen nur die Erstellung eines semantischen Netzes. Es ist nicht möglich Begriffe des Netzes mit Metadatenbeschreibungen für Ressourcen in Verbindung zu setzen. Ein Werkzeug für die Erstellung einer Wissensbasis eines web-basierten Lernsystems muß aber genau diese Verbindung ermöglichen, damit anhand einer Suche auf dem semantischen Netz, Medienbausteine ausgewählt werden können.

5.2 Architektur des ConceptSpace-Editors

In [100] wird ein objektorientiertes Framework für *Begriffliches Browsern* beschrieben, auf dem der in Kapitel 5.3.2 beschriebene ConceptualBrowser basiert. Das Framework besteht aus fünf Komponenten, die zusammengestellt und entsprechend konfiguriert eine Navigationsmöglichkeit für strukturierte Suchergebnisse realisieren. Im einzelnen besteht das System aus folgenden Komponenten:

- 1 Die *Visualization*-Komponente ist für die Präsentation der Suchergebnisse verantwortlich.
- 1 Die *ConceptEngine*-Komponente ist für das Einleiten einer Suche und das Berechnen

von Strukturen anhand einer Analyse der Ergebnisse verantwortlich. Diese Komponente kann mit dem Begriff Inference Engine der wissensbasierten Systeme gleichgesetzt werden.

- 1 Für die Präsentation zusätzlicher Attribute der Suchergebnisse kann eine *Inspector*-Komponente verwendet werden.

- 1 *DataBaseAccess*

Die *DataBaseAccess*-Komponente wird von der *ConceptEngine* mit einer Suchanfrage gestartet. Sie ist für den Zugriff auf die eigentliche Datenbank verantwortlich und liefert die potentiellen Ergebnisse an die *ConceptEngine* zurück.

- 1 Als letzte Komponente wird ein *ScaleEditor* aufgeführt, der eine graphisch interaktive Manipulation von Skalen ermöglicht. Es wird aber weder erläutert um welche Art Skalen es sich handelt, noch in welcher Weise mit diesen Skalen interagiert werden kann.

Obwohl diese Architektur der für den *ConceptSpace*-Editor benötigten Struktur schon sehr nahe kommt, kann sie nicht direkt verwendet werden. Zum einen müssen die zusätzlichen Werte, die mit Hilfe der *Inspector*-Komponente angezeigt werden können, in der gleichen Datenbank wie die eigentlichen Dokumente verwaltet werden. Es ist also nicht möglich, zuerst eine Suche auf dem semantischen Netz durchzuführen und dann eine Anfrage nach Medienbausteinen zu stellen. Zum anderen ist die *ConceptEngine* nicht in der Lage, eine Suche anhand der bestehenden Ergebnis zu detaillieren. Die *ConceptEngine* kann nur die Ergebnisse einer Suchanfrage gruppieren und eine Navigation darauf anbieten. Eine interaktive Suche ist daher nicht möglich.

Aus den in Kapitel 5.1.2 beschriebenen Anforderungen lässt sich ableiten, daß die Datenstrukturen für die Speicherung des semantischen Netzes auf zwei verschiedene Arten von Repräsentationskomponenten der Benutzeroberfläche abgebildet werden müssen, die als Schemavisualisierung und Graphvisualisierung bezeichnet werden können. Zum einen müssen die Regeln und Beschränkungen des Schemas so abgebildet werden, daß sie für einen Benutzer nachvollziehbar sind. Die Realisierung dieser Komponente ist damit stark von der Ausdrucksfähigkeit des Schemas abhängig und kann daher nur sehr schwer allgemein spezifiziert werden. Sie wird daher im folgenden nicht weiter grundlegend untersucht, sondern erst in Abschnitt Kapitel 5.8.3, wieder aufgegriffen, der die tatsächliche Implementierung beschreibt. Einfacher gestaltet sich auf den ersten Blick die Abbildung der Datenstrukturen, durch die die Begriffe und Relationen des Netzes repräsentiert werden auf eine graphische Darstellung des Netzes in Form von Knoten und Kanten. Hier liegt das Problem nicht in der Entwicklung einer Datenstruktur, die die Knoten und Kanten des Netzes speichert. Vielmehr erfordert die graphische Präsentation und vor allem die Fähigkeit der Präsentation eine interaktive Navigation zu ermöglichen, die Entwicklung von Algorithmen und Verfahren, die in der Lage sind aus dynamisch generierten Daten, die die Antwortmengen enthalten eine graphische Repräsentation zu generieren. Welche prinzipiellen Möglichkeiten es gibt dynamisch graphische Strukturen zu realisieren wird daher in den nächsten Abschnitten beschrieben.

5.3 Visualisierungen von Baumstrukturen

Navigationssteuerungen für allgemeine Netze existieren bisher nur für spezielle Anwendungen oder sind in Systemen für spezielle Netztopologien in kommerziellen Anwendungen zu finden.

Insbesondere ist hier die Baumstruktur zu nennen. In dieser hat jedes Element eine Verknüpfung zu seinem übergeordneten Element, ausgenommen das Wurzelement. Beliebig viele Verknüpfungen können zu möglicherweise untergeordneten Elementen vorliegen. Benachbarte Elemente können jedoch keine Beziehung untereinander haben. Die Baumstruktur vereinfacht in vielen Fällen die tatsächlich vorliegende Struktur der Objektbeziehungen, da bei semantischen typischerweise eine vernetzte, und unter Umständen zyklische Struktur vorliegt und die Baumstruktur diese als grobe Vereinfachung darstellt. Mit dem Zwang einer Einordnung für jedes Element wird der Informationsraum in disjunkte Klassen zerlegt. Mit dieser Klassifikationsstruktur werden bestimmte Objektverknüpfungen verdeckt oder redundante Zusatzinformationen in die Baumstruktur eingefügt. Dazu kommt, daß auch bei großen Baumstrukturen der Überblick verloren gehen kann.

5.3.1 Microsoft Explorer

Einer der gebräuchlichsten Anwendung von Baumstrukturen ist das Inhaltsverzeichnis eines Buches. Quasi die moderne Form hierzu ist ein hierarchisches Dateisystem, eine Realisierung dieser Navigation zeigt Abbildung 36 mit dem Microsoft Explorer. Hierbei wird ein Baum nicht in der klassischen Ausrichtung angeboten, sondern in einer, wie auch im Buch, um 90 Grad gedrehten Variante. Der am Bildschirm vorhandene Platz wird hierbei optimal ausgenutzt.

Dieses Anwendungsbeispiel ist typisch für eine Navigation auf einem Graphen. Einzelne Ordner können geöffnet und geschlossen werden, somit kann interaktiv ein passender Ausschnitt gewählt werden. Diese interaktive Funktionalität trägt wesentlich zur Übersichtlichkeit und zum großen Erfolg dieser Navigationsmethode in einem hierarchischen Dateisystem bei.

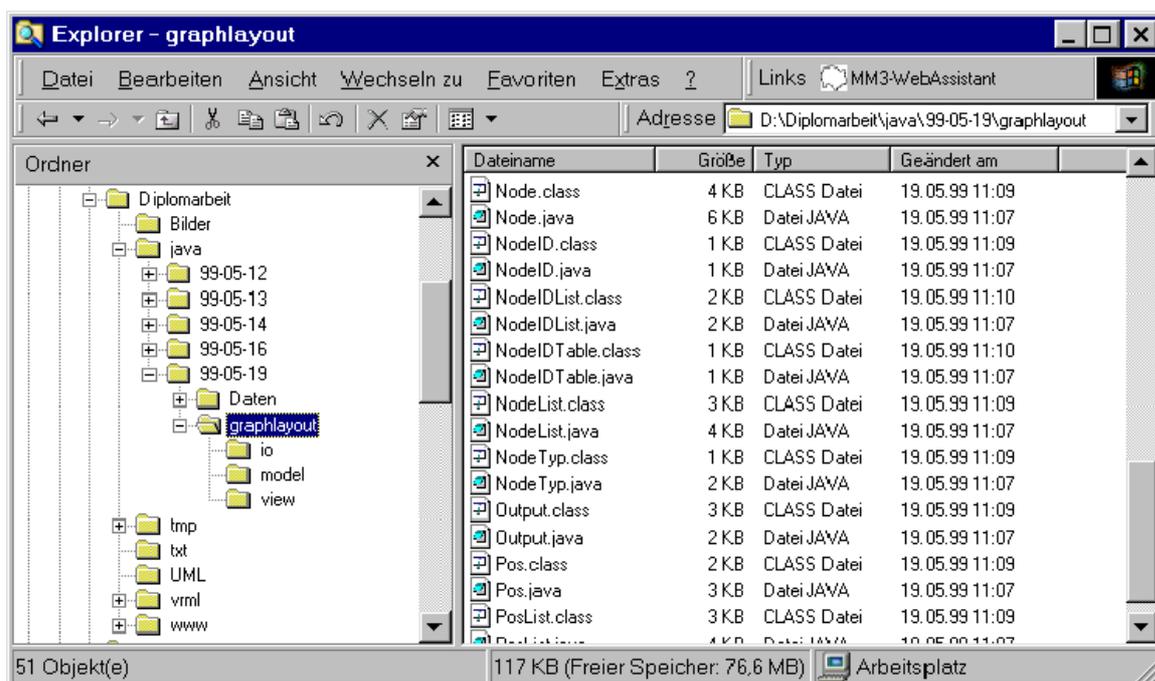


Abbildung 36: Baumansicht des Microsoft Explorers

Das Beispiel zeigt, dass die Baumsicht eine sehr gute Möglichkeit darstellt, auf einer hierarchischen Datenstruktur zu navigieren, sie sich allerdings nicht zur Visualisierung und Navigation auf einer Netzstruktur eignet. Zyklen oder nicht gerichtete Graphen können nicht adäquat wiedergegeben werden, da jeder Knoten genau einen Vaterknoten hat und dementsprechend nicht in zwei verschiedenen Ordnern enthalten sein kann.

5.3.2 HAWK

Der Conceptual Browser Hawk ist ein an der TU Darmstadt im Rahmen des MILAN-Projekts entwickelter Browser zur Navigation auf Suchergebnissen, die mit Hilfe einer formalen Begriffsanalyse auf gerichtete azyklische Graphen abgebildet werden [86]. Suchergebnisse können gruppiert und einzelnen Ordnern zugewiesen werden. Wie beim Explorer können Ordner beliebig viele Unterordner enthalten. Er unterscheidet sich vom Internet-Explorer insoweit, dass er die Limitierung aufhebt, daß jeder Knoten (oder Ordner) nur genau einen Vater-Knoten besitzt. So können Ordner und demnach alle in diesem Ordner enthaltenen Unterordner und Knoten eine Verbindung zu mehreren Vater-Knoten besitzen. Der Conceptual-Browser weist jedem Ordner ein Dreieck zu, das angibt, ob ein Ordner noch Unterordner besitzt. Zusätzlich wird angezeigt, welcher Prozentsatz an Treffern zur Suchanfrage in einem markierten Ordner zusammengefasst sind. Ein allgemeiner Graph, der Zyklen aufweist, kann mit dem Browser nicht dargestellt werden. Während beim Internet Explorer alle Ordner und Dateien durch Icons der gleichen Größe dargestellt werden und damit der zur Verfügung stehende Platz auf dem Bildschirm bei einem Ordner mit sehr vielen Einträgen sehr schnell belegt ist, kann der Conceptual Browser Ordner die weit von markierten Ordnern entfernt sind, durch kleinere Icons repräsentieren und so den verfügbaren Platz besser nutzen, ohne dass die Navigation für den Benutzer erschwert wird. Bei der Visualisierung geht man hier von der Annahme aus, dass für einen Benutzer bei der Navigation die direkt zu einem vom Benutzer markierten Knoten, benachbarten Knoten interessanter sind, als solche die sich weiter entfernt befinden. Das muss nicht generell für semantische Netze gelten, sondern wird von der semantischen Bedeutung der Beziehung zwischen den Knoten bestimmt. Abbildung 37 zeigt die Präsentation von Sucher-

gebnissen im ConceptualBrowser.

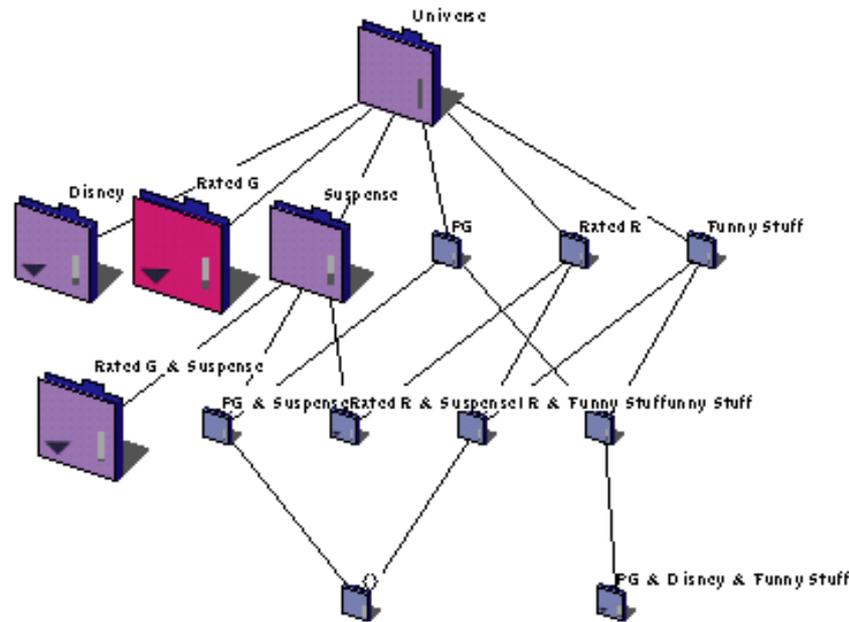


Abbildung 37: Präsentation der Suchergebnisse mit HAWK ([99])

Da der bei MILAN verwendete Algorithmus zur Gruppierung der Suchergebnisse auf die Zusammenfassung ähnlicher Treffer zu Ordnern hin entwickelt wurde, ist die Annahme aber in diesem Fall zutreffend. Der Ansatz, eine vergrößerte Darstellung der Gesamtstruktur gemeinsam mit einer Detailansicht zu visualisieren, ist auch als Focus+Context-Technik bekannt [21]. Ein sehr früherer Vertreter dieser Technik, verschiedene Knoten des Graphen in verschiedenen Größen zu visualisieren, um damit eine Navigation zu erleichtern, wurde vom PARC-Entwicklungslabor von Xerox entwickelt und ist in dem Produkt "The Hyperbolic Tree" von inxight umgesetzt [73].

5.3.3 The Hyperbolic Tree von inxight

Ein weiteres Beispiel einer Navigationsteuerung für eine hierarchische Baumstruktur ist das Applet "The Hyperbolic Tree" von inxight [59]. Mit diesem Applet wird ein Verzeichnisbaum wie ein Blick durch ein extremes Weitwinkelobjekt (fish eye) dargestellt. Durch diesen Effekt werden die im Blickzentrum liegende Objekte sehr groß und deutlich. Zu den Rändern hin werden die Objekte kleiner und überlappen sich zum Teil. Dadurch sind gleichzeitig Details zu erkennen und die Sicht auf das Ganze bleibt erhalten.

Das Öffnen von einzelnen Objekten ist möglich, dabei werden untergeordnete Objekte sichtbar. Es ist jedoch nicht möglich, diese wieder zu entfernen. Werden Knoten durch den Benutzer ausgewählt, wird das Objekt in das Zentrum des Blickfelds bewegt. Dabei ordnen sich die referenzierten Objekte kreisförmig bzw. in Segmenten um das Objekt im Zentrum an. Weiterhin können Knoten interaktiv bewegt werden. Dieser Bewegung folgen alle anderen Knoten. Damit kann eine Wanderung durch den Graphen durchgeführt werden. Der Überblick bei der Navigation geht jedoch schnell verloren. Abbildung 38a zeigt die Startseite mit der Gruppierung der Sparten von der Wochenzeitung "Die Zeit". Ausgehend von fünf Unterrubriken ist

Implementierungen beschrieben.

5.4.1 Ilog

Die Firma Ilog [57] bietet die Java-Klassenbibliothek JViews für fünf unterschiedliche Graph-typen an. Die Layout-Algorithmen sind auf die speziellen Graphstrukturen optimiert. Die Visualisierung einer Baumstruktur und die kreisförmige Anordnung von Objekten mit Satelliten erfolgt sehr performant. Die Algorithmen für rein zyklische Netze oder Mischformen von Graphen benötigen einige Rechenzeit, jedoch ist diese für eine interaktive Arbeit noch akzeptabel (Abbildung 39 in 7 Sekunden). Bei Veränderung der Struktur des Graphen wird das Layout neu berechnet, dadurch entsteht meist ein völlig geänderte und nicht vergleichbare Darstellung zu dem vorherigen Layout des Graphen. Weiterhin treten bei einigen Demonstrationsbeispielen von Ilog mit Mischform teilweise unbefriedigte Ergebnisse auf. Dies ist in der vergleichenden Darstellung zu entnehmen. Abbildung 39 zeigt das mit dem Ilog erzielte Layout und darunter ist in Abbildung 40 die anschaulichere kreuzungsfreie Darstellung des Graphen zu sehen

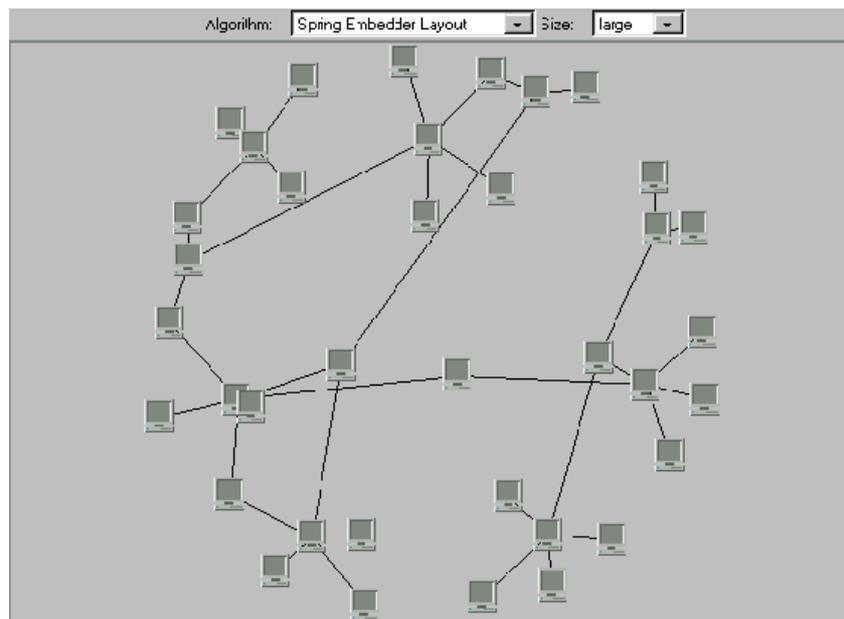


Abbildung 39: Visualisierung mit Ilog-Bibliothek

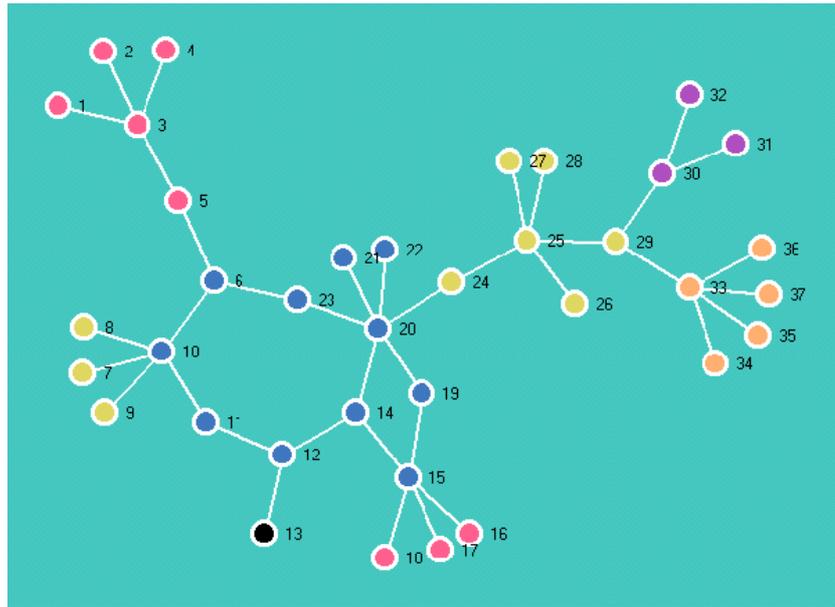


Abbildung 40: Kreuzungsfreie Visualisierung des ILog-Graph

5.4.2 Brain von Natrifical

Mit dem Programm Brain von Natrifical [98] können Dateien oder Notizen des Benutzers in einem Netz verwaltet werden. Durch Selektieren eines Objekts wird dieses in das Zentrum gerückt. Näher am neuen Zentrumsknoten liegende Objekte werden nun dargestellt und weiter entfernt liegende Objekte ausgeblendet. Kleine grüne Satelliten um die Objekte symbolisieren Links zu weiteren Objekten. Es werden maximal zwei Hierarchieebenen angezeigt, hierin unterscheidet sich der Ansatz von dem Hyperbolic Tree. Das Verfahren ist in Abbildung 41 dargestellt. Die Visualisierung wird auf eine nahezu sternförmige Struktur abgebildet. Hierbei können jedoch auch Objekte der gleichen Ebene miteinander verbunden sein. Um eine übersichtliche Darstellung zu erreichen, werden nur maximal zwei Hierarchieebenen dargestellt. Durch diesen stark begrenzten Ausschnitt kann praktisch ein einfaches allgemeines Netz dargestellt werden.



Abbildung 41: Visualisierung mit Nartificial Brain

5.4.3 i2 Analyst Notebook

Das Analyst's Notebook von i2 [54] dient zur Analyse von Beziehungsgeflechten z.B. im Bereich der Kriminalistik. Der verwendete Layout-Algorithmus richtet die Objekte nach Möglichkeit gleichmäßig horizontal nebeneinander oder vertikal untereinander aus (Abbildung 42). Weiterhin werden sternförmige Strukturen und 1:N:1- Verbindungen gesondert behandelt. Kanten werden nach Möglichkeit waagrecht, senkrecht oder in einem Winkel von 45 Grad angeordnet. Dabei werden auch Knickpunkte verwendet. Durch die Ausrichtung der Objekte, geraten diese gegenüber den Beziehungen visuell in den Vordergrund. Dadurch kann ein falscher Eindruck über die Struktur des Netz und die Beziehungen zwischen den Objekten entstehen.

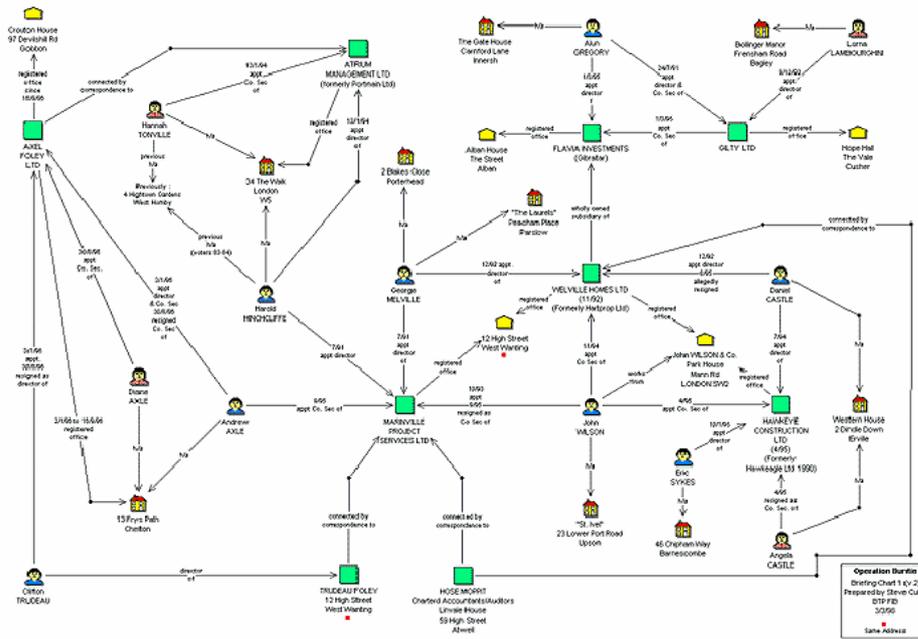


Abbildung 42: Visualisierung mit Analyst Notebook

Mit einer weiteren Darstellungsart, die in Abbildung 43 gezeigt ist, ist eine Überblicksdarstellung mit vielen Objekten möglich. Die Objekte werden hier als kleine Punkte symbolisiert. Bei dieser Ansicht hat es den Anschein, daß nur die Positionen der Hauptknoten berechnet und die Masse der Blattknoten einfach um diese Hauptknoten in Segmente angeordnet werden. Überschneidungen sind dabei möglich. Die Anordnung der Blattknoten erfolgt schuppenartig, so daß die Knoten möglichst wenig Platz einnehmen. Eine interaktive Navigation ist nicht möglich. Zudem kann sich das Aussehen des Layouts bereits bei einer kleinen Modifikation der Struktur komplett verändern.

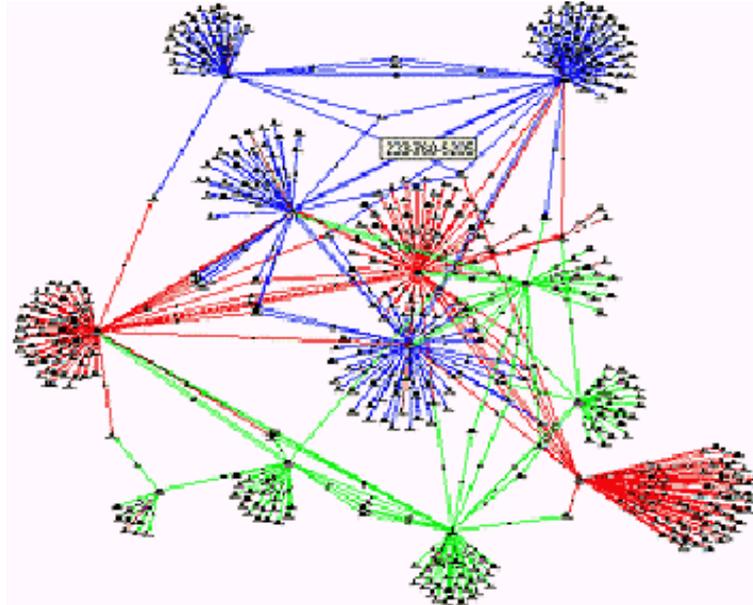


Abbildung 43: Visualisierung der Übersicht einer Analyst Notebook Struktur

5.5 Zusammenfassung

Das Beispiel “Explorer” zeigt, daß für die Visualisierung und Navigation in einer hierarchischen Baumstruktur effiziente Implementierungen vorliegen. Eine Sortierung nach verschiedenen Kriterien ist hier ebenfalls möglich. Existiert bei einer vorliegenden Baumstruktur keine Hierarchie ist eine sternförmige Anordnung vorzuziehen. Die Visualisierung mit dem “The Hyperbolic Tree” kann hierfür als Beispiel angesehen werden. Die Visualisierung von Netzen wurden anhand der Produkte von I2 und Ilog besprochen. Das Produkt Analyst’s Notebook von I2 ist eine konfigurierbare Anwendung, - JViews von Ilog eine Klassenbibliothek. Obwohl die Ausrichtung der Produkte deshalb sehr unterschiedlich ist, können folgende Feststellungen getroffen werden:

- 1 Eine interaktive Navigation ist nicht vorgesehen
- 1 Bereits bei geringen Veränderungen in der Netzstruktur kann sich das Layout komplett ändern.
- 1 Überschneidungen von Knoten und Kanten können zu einer fehlerhaften Interpretation führen.
- 1 Das in [34] zu findende Konzept der Visualisierung von inhaltlicher Nähe durch räumliche Nähe ist mit Einschränkungen nur beim Conceptual Browser des MILAN-Projekts realisierbar.

Aus den Beispielen ist zu entnehmen, dass die Verwendbarkeit von Konzepten zur interaktiven Navigation und Visualisierung heute bereits gut mit Baumstrukturen umsetzbar sind. Liegt in der Anwendung keine Baumstruktur vor, sollte demnach zuerst geprüft werden, ob diese sinnvoll auf eine Baumstruktur reduziert werden kann. Das in Kapitel Kapitel 3.7 beschriebene

Wissensnetz, das als Voraussetzung für adaptive web-basierte Lernsysteme gesehen wird, zeichnet sich aber dadurch aus, dass keine Baumstruktur vorliegt, sondern den Benutzern Möglichkeiten geboten werden müssen, allgemeine Graphen zu erzeugen und auf ihnen zu navigieren. Zudem darf es sich nicht um eine statische Visualisierung handeln, da sich das Layout durch das Hinzufügen neuer Knoten ändern kann. Als Rahmenbedingungen für die Entwicklung einer graphischen Komponente zur Erstellung von und Navigation auf semantischen Netzen wurden daher folgender Kriterien angenommen:

- 1 Navigation in allgemeinen Netzen,
- 1 Stabilität bei geringen Veränderungen der Netzstruktur,
- 1 Visualisierung von inhaltlicher Nähe durch räumliche Nähe

5.6 Darstellung des Graphen

Die unterschiedlichen Anforderungen an die Layoutberechnung und Anzeige des Graphen können nicht unabhängig voneinander betrachtet werden, da die Optimierung eines Punktes für die Visualisierung nicht optimal ist, wie sich leicht am Beispiel des Leiterplatten-Layout zeigen läßt. Hier erfolgt eine Optimierung hinsichtlich einer minimalen Anzahl von Überkreuzungen der Leiterbahnen. Dies führt im allgemeinen Fall für den menschlichen Betrachter jedoch nicht zu einem übersichtlichen Layout. Die Berechnung des Layout für den Graphen ist kein triviales Problem. Für einen Betrachter ist es möglich, einen vorgegebenen Graphen zu beurteilen und festzustellen, ob das Layout für ihn verständlich ist. Diese Betrachtung ist zum einen subjektiv und zum anderen können aus dieser Aussage nur bedingt allgemeine Konstruktionsregeln abgeleitet werden. Folgende Möglichkeiten werden gesehen: (Literatur!!)

- 1 Anordnung durch Optimierung einer Zielfunktion
Bei diesem Ansatz wird versucht, einige Kriterien für eine übersichtliche Layout-Gestaltung eines Graphen zu qualifizieren und in eine mathematische Optimierungsfunktion zu fassen. In der Zielfunktion sind die Abweichungen von den Kriterien zu bewerten. Im Optimum sind alle Kriterien erfüllt und die Abweichung ist gleich null. Damit wird bei diesem Ansatz der Funktionswert minimiert. Da jeder Knoten durch jeweils eine x- und y-Koordinate beschrieben wird, sind bei einem Graphen mit N Knoten somit 2N Parameter zu optimieren. Wie im weiteren Verlauf des Kapitels gezeigt wird, kann die Optimierungsfunktion aus einzelnen positiven Summanden aufgebaut werden. Jeder Summand bewertet die Abweichung von einem Kriterium für das Aussehen des Graphen. Durch Hinzunahme von weiteren Kriterien kann somit das Layout eines Graphen weitgehend beeinflusst werden. Von Nachteil bei diesem Ansatz ist, daß zum einen mit der Knotenanzahl der Rechenaufwand nichtlinear ansteigt und zum anderen, das Finden des globalen Minimum durch das Vorliegen von mehreren lokalen Minima erschwert wird.
- 1 Anordnung durch eine Strukturanalyse
Die vorliegenden Basisstrukturen des Graphen, wie Zyklen und sternförmige Anordnungen, werden durch Strukturanalyse ermittelt. Mit dieser Information wird der Graph systematisch vereinfacht. Zunächst werden alle Blattknoten eliminiert. Die Elimination wird so oft wiederholt, bis keine Blätter mehr existieren. Existiert im letzten Schritt nur noch ein Knoten, ist der zu visualisierende Graph ein Baum. Für diesen Fall kann ein spezielles Layout benutzt werden. Anderenfalls sind in dem nun vorliegenden Netz die Zyklen oder auch Kombinationen von Zyklen zu bestimmen. Zur Reduzierung wird jeder Zyklus durch einen virtuellen Knoten abstrahiert (siehe Abbildung 44). Dieses Ver-

fahren wird fortgesetzt bis kein Zyklus mehr existiert. Im allgemeinen Fall können dann wieder Blätter eliminiert werden. Mit diesem Vorgehen kann die Grundstruktur des Graphen ermittelt werden. Für die Visualisierung ist die Zusammenfassung der Knoten wieder aufzulösen. Dabei sind die zuletzt zusammengefaßten Knoten zuerst in die Layout-Berechnung einzubeziehen. Danach folgen sukzessive die anderen Knoten. Bei einem komplexen Graphen kann möglicherweise durch das Löschen bzw. Hinzufügen von Knoten dieser auf andere bestimmte Grundstrukturen reduziert werden. Damit kann sich das Layout wesentlich vom vorhergehenden unterscheiden. Die Kontinuität kann dadurch verloren gehen.

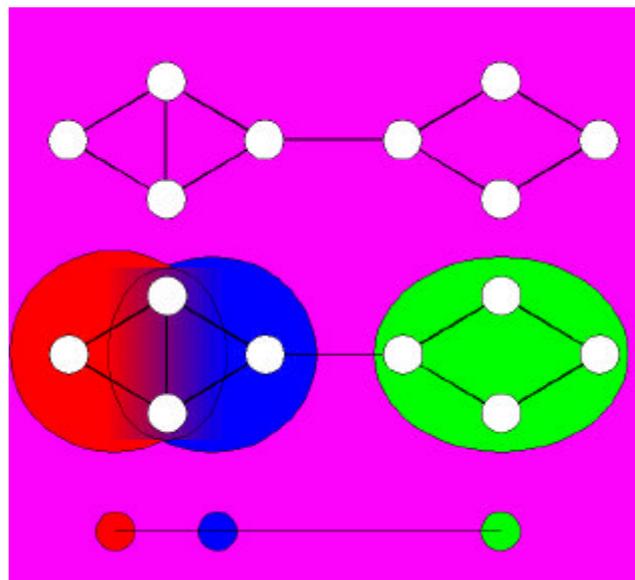


Abbildung 44: Zusammenfassung von Knoten zu virtuellen Knoten

1 Topologische Verfahren

Aus der Topologie eines Graphen kann der Abstand zweier Knoten bestimmt werden. Sind zwei Knoten direkt miteinander verbunden, beträgt der Abstand eins. Erfolgt dagegen die Verbindung über einen dritten Knoten, so haben sie den Abstand zwei. Allgemein wird der Abstand zwischen zwei Knoten definiert, als die minimale Anzahl der Kanten, die zwischen den beiden Knoten liegen. Zusätzlich können, wie bereits erläutert, die Kantenlängen zur Symbolisierung der Ähnlichkeit zweier Knoten herangezogen werden. Sind die Kantenlängen kleiner eins, sind die Knoten einander ähnlicher, dagegen unähnlicher, wenn die Kantenlänge größer eins ist.

1 Problemangepasste Darstellung

Ein Benutzer kann die Struktur eines Graphen wesentlich besser erfassen, wenn diese problemangepaßt dargestellt wird. Ein besonderes Beispiel stellt hier die baumförmige bzw. sternförmige Anordnung dar (vgl. Abbildung 45). Steht bei der sternförmigen Anordnung ein Knoten im Mittelpunkt, so ist es in der baumförmigen Anordnung die Hierarchie, die betont wird. Obwohl es sich bei beiden Abbildungen um denselben Graphen handelt, ist die hierarchische Struktur des Netzes in der sternförmigen Visualisierung für einen menschlichen Betrachter nicht offensichtlich zu erkennen.

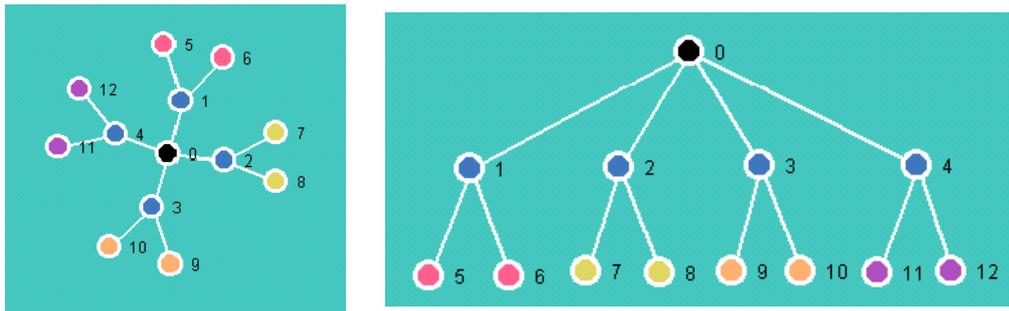


Abbildung 45: Stern- bzw. baumförmige Visualisierung eines Graphen

1 Kombinationsverfahren

Die bisher vorgestellten Verfahren lassen sich nicht ohne weiteres kombinieren. Der Ansatz, inhaltliche Nähe zweier Knoten durch die geometrische Länge der Relation zu symbolisieren und nicht verbundene Knoten möglichst weit auseinander zu positionieren, führt bsp. bei manchen Strukturen für die Visualisierung zu einer nicht optimalen Lösung. Beispiele sind hier die 1:n:1-Verbindung, die in Abbildung 46 gezeigt ist, oder ein aus Zyklen bestehendes Netz. Soll das Layout solcher Strukturen ebenfalls durch eine Optimierungsfunktion festgelegt werden, ist die Abweichung von der idealtypischen Struktur in die Zielfunktion aufzunehmen.

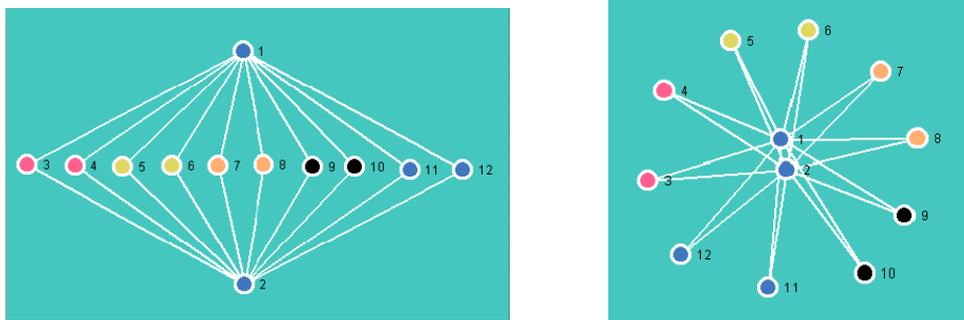


Abbildung 46: Visualisierungen für 1:n:1-Beziehungen

5.6.1 Restriktionen

Aufgrund der beschränkten Visualisierungsfläche auf einem Monitor und der begrenzten Aufnahmefähigkeit eines menschlichen Betrachters können große komplexe Graphen nicht in ihrer Gesamtheit dargestellt bzw. erfaßt werden. Folgende drei unterschiedliche Basisvarianten zur Navigation durch einen Graphen können unterschieden werden: (Literatur)

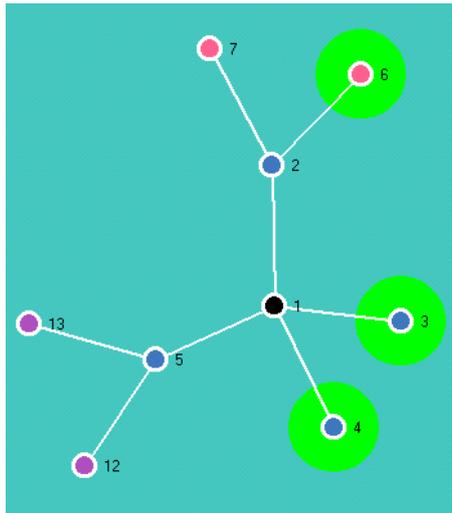
- 1 Verschiebung des sichtbaren Bereichs
- 1 Ausschnittvergrößerung bzw. Verkleinerung (zoomen)

1 Öffnen und Schließen von einzelnen Knoten

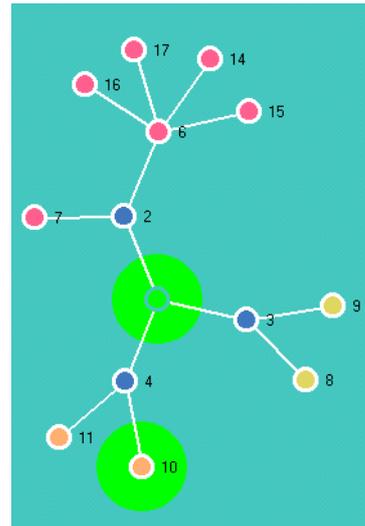
Daneben existieren Kombinationen der Basisvarianten, die weitere Möglichkeiten erschließen, zum Beispiel eine Navigation mit zwei Fenstern, wie sie bsp. in (GMD-ACM Comm) realisiert ist. In einem Fenster wird ein Überblick des gesamten Graphen dargestellt, mit der Möglichkeit auf einen einzelnen Bereich zu fokussieren. Dessen Detailansicht wird in einem zweiten Fenster dargestellt. Dieses Vorgehen entspricht einer Landkarte mit einer Überblicksdarstellung und einer zusätzlichen Detailansicht in einem Stadtplan. Eine weitere Möglichkeit ist die Fischaugen-Darstellung. Dabei wird die Ausschnittsvergrößerung auf einen Bereich beschränkt, ähnlich wie bei einem Blick durch eine Lupe.[Literatur aus IGD-Diss] Wie bereits diskutiert, hat sich die Navigation in hierarchischen Baumstrukturen nach der Vorgehensweise wie mit dem Explorer gut bewährt. Bei dieser Navigation werden interaktiv einzelne Knoten geöffnet bzw. geschlossen. Damit ist gleichzeitig eine Übersichts- und Detaildarstellungen des Graphen möglich. Diese ist quasi das Optimum zur Darstellung und Navigation auf einem begrenzten Raum. Sie soll deshalb auf allgemeine Netze übertragen werden. Das Vorgehen bei dieser Navigation entspricht einer Informationsreduktion des Graphen auf das Wesentliche. Es werden nur die Informationen angezeigt, die zur Zeit für den Benutzer zur Navigation wichtig sind. Bei diesen Operationen soll der visuelle Eindruck über alle anderen Knoten und Kanten im wesentlichen erhalten bleiben.

5.6.2 Strategie für Zusammenfassungen von Knoten

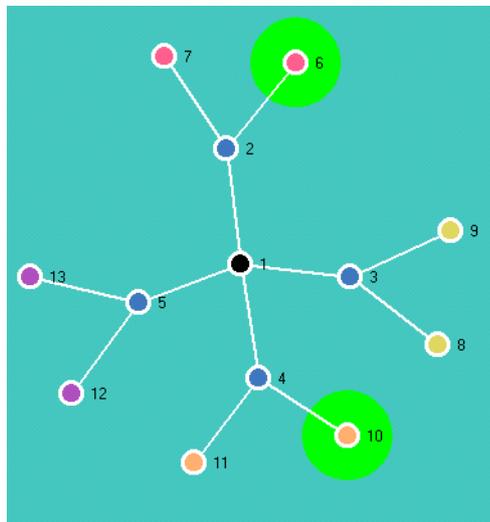
Durch das Öffnen eines Knotens sollen die darin enthaltenen Knoten sichtbar werden. In der Regel sind dabei andere Knoten zu schließen, da nur eine maximale Anzahl von Knoten gleichzeitig übersichtlich dargestellt werden kann. Die zu schließenden Knoten werden über den topologischen Abstand bestimmt. Zunächst wird der Knoten ausgewählt, der die größte Entfernung von zu dem öffnenden Knoten hat. In einem neuen virtuellen Knoten wird dieser und alle (direkt) verbundenen Knoten, die nicht mehr dargestellt werden können, zusammengefaßt. Dabei kann es vorkommen, daß Knoten geschlossen werden, die erst mit einer der letzten vorangegangenen Interaktion geöffnet wurden. Um dies zu verhindern, kann zusätzlich ein zeitlicher Aspekt bei der Auswahl der Knoten berücksichtigt werden. Zusätzlich kann der zeitliche Aspekt auch bei den umliegenden Knoten berücksichtigt werden. Weiterhin kann die Anzahl der minimal zurückliegenden Interaktionen für einen Knoten seit dessen Öffnung angegeben werden. Damit können nur Knoten geschlossen werden, die entsprechend lange sichtbar waren. Können aufgrund dieser Restriktion nicht genügend Knoten in dem virtuellen Knoten zusammengefaßt werden, kann der Auswahlprozeß wiederholt werden. Alternativ können die Knoten geschlossen werden, die bei der Interaktion "Öffnen" eines Knotens, mit dem ausgewählten Knoten zusammen entstanden sind. Nach dieser Strategie sind alle Knoten zu schließen, die bei der selben Aktion "Öffnen" entstanden sind. Dieser Ansatz ist für den Benutzer einfacher zu interpretieren, da er der logischen Undo-Funktionalität entspricht. Weiterhin soll eine Undo-Funktion ermöglichen, Interaktionen generell wieder zurückzunehmen. Eine interaktive Auswahl von Layout-Darstellungsarten sollte vorgesehen werden, da es nicht nur eine Darstellung für einen Graphen gibt und auch nicht die optimale apriori bestimmt werden kann. Ebenso soll das automatisch generierte Layout manuell veränderbar sein. Eine Sequenz von drei Interaktionen zeigt Abbildung 47. In Abbildung 47a ist der Ausgangsgraph dargestellt, in dem nacheinander zwei Knoten (3 und 4) geöffnet werden. Das Ergebnis zeigt Abbildung 47b. In einer weiteren Interaktion wird der Knoten 6 geöffnet. Hierbei werden vier Knoten (1, 5, 12, 13) in einem virtuellen Knoten zusammengefaßt (s. Abbildung 47c).



a) Ausgangsgraph



b) Graph nach Öffnen der Knoten 3 und 6



c) Graph nach dem Öffnen von Knoten 6 (Knoten 1,5,12,13 zusammengefasst)

Abbildung 47: Sequenz einer Interaktion mit Zusammenfassung von Knoten

5.6.3 Optimierung des Graphen

Die Aufgabe von Optimierungsverfahren ist das Auffinden der optimalen Werte für die Variablen eines Vektors x , für den eine Zielfunktion $f(x)$ ein globales Optimum ergibt. Im folgenden wird die Bestimmung des Optimum auf die Suche nach einem Minimum beschränkt. Meist besitzt die Zielfunktion mehrere lokale Minima. Da im voraus deren Anzahl nicht bekannt ist, entstehen hier offensichtlich Schwierigkeiten bei der Bestimmung des globalen Minimum. Denn selbst die Bestimmung der Lage eines lokalen Minimum ist keine triviale Aufgabe, da sie eine Suche im n -dimensionalen Raum erfordert. Numerische Optimierungsverfahren werden in vielen Gebieten der Technik eingesetzt, die auf unterschiedlichen Metho-

den basieren. Die bekannteste Methode zur Lösung einer Optimierungsaufgabe erfolgt über die Berechnung des Gradienten. Bei einer Vielzahl von Problemen liegt jedoch die zu optimierende Funktion nicht analytisch vor bzw. der Gradient kann aus dieser nicht analytisch abgeleitet werden. Selbst die numerische Berechnung ist schwierig bzw. vielfach nicht möglich. Deshalb werden meist gradientenfreie Verfahren eingesetzt. Diese lassen sich grob in klassische Verfahren, Zufallssuchverfahren, neuronale Netze und genetische Algorithmen klassifizieren.

Für die Implementierung des ConceptSpace-Editors wurde eine klassische Methode, welche von M.J.D. Powell [105] angegeben wurde, realisiert. die eine Variante der wohl ältesten gradientenfreien Methode darstellt der sukzessiven Variation der Variablen x_i . Hierbei werden ausgehend von einem Startpunkt u die Variablen x_1, x_2, \dots, x_n der Reihe nach einzeln solange verändert, bis jeweils das Minimum der Zielfunktion in der Suchrichtung gefunden ist. Es erfolgt also eine sukzessive Minimumsuche in Richtung der Koordinatenachsen. Sobald dieser Suchprozeß abgeschlossen ist, ist der Ausgangspunkt für den nächsten Durchlauf erreicht, bei dem erneut sukzessiv die Variablen verändert werden. In dieser Weise fährt man fort, bis das Minimum x hinreichend gut erreicht ist. Infolge der starren Wahl der Suchrichtungen ist das Verfahren schwerfällig und langsam. In einer von Powell vorgeschlagenen Variante wird diesem Umstand abgeholfen, in dem das Verfahren nur am Anfang die Koordinatenachsen als Suchrichtungen verwendet und danach konjugierte Suchrichtungen erzeugt. Die Basismethode nach Powell's Variante kann wie folgt beschrieben werden:

Ausgehend von den Anfangswerten

$$\begin{aligned} x_0 &= u \\ s_i &= e_i \end{aligned}$$

mit dem Startpunkt u und dem Einheitsvektor e_i besteht das Basisverfahren aus einem Iterationszyklus mit folgenden Schritten:

- Bestimme für alle n Suchrichtungen die optimalen Werte λ_i so, daß die Funktion $f(x_{i-1} + \lambda_i s_i)$ an der Stelle λ_i ihr Minimum entlang der Suchrichtung s_i erreicht. Definiere den Startpunkt für die nächste Suche zu:

$$x_i = x_{i-1} + \lambda_i s_i$$

- Ersetze die Suchrichtung

$$s_i = \begin{cases} s_{i+1} & \text{für } i=1,2,\dots,n \\ x_n - x_0 & \text{für } i=n \end{cases}$$

- Bestimme λ_n so, daß die Funktion $f(x_n + \lambda_n s_n)$ an der Stelle λ_n ihr Minimum entlang der Suchrichtung s_n erreicht und definiere

$$x_0 = x_n + \lambda_n s_n$$

Wie Powell zeigte, sind für eine quadratische Funktion die Suchrichtungen zueinander konjugiert. Damit wird spätestens nach n -Iterationszyklen bei einer quadratischen Funktion das Minimum gefunden. Bei einem quadratischen Funktionstyp kann mit Hilfe von drei Funktionswerten das Minimum angegeben werden. Für die im allgemeinen nichtquadratische Funktion entspricht dieser Wert dann einer Schätzung. Zum Finden des lokalen Minimum entlang der Suchrichtung wird dieses Schätzverfahren also mehrfach angewandt.

5.7 Funktionswertberechnung des Graphen

Wie beschrieben ist wird das Layout über einen Optimierungsansatz bestimmt. Hierzu wird jedes Layout bewertet und auf einen Funktionswert abgebildet. Eine Variation des Layouts führt zu unterschiedlichen Funktionswerten, die ein Optimierer bewertet und daraus neue Varianten generiert. Diese Sequenz wird gestoppt, wenn vom Optimierer keine Varianten mit niedrigerem Funktionswert gefunden werden. Hierfür ist zur Bewertung des Layouts des Graphen eine Zielfunktion zu definieren. Für diese wird gefordert:

Für alle betrachteten Graphen soll der niedrigste Funktionswert mit dem besten visuellen Eindruck korrelieren. Weiterhin soll die Zielfunktion ein stetiges Verhalten zeigen.

Die Schwierigkeit liegt in der Umsetzung dieser komplexen Forderung. Eine Methodik zur Festlegung ist, Details des Layouts durch Terme zu bewerten und diese dann additiv zur Zielfunktion zusammenzufassen. Die folgenden Anforderungen an das Layout können zur Bestimmung der Terme genutzt werden:

- 1 Einhaltung einer vorgegebenen Kantenlänge.
- 1 Topologische und räumliche Abstände von nicht direkt verbundenen Knoten sollen sich entsprechen.
- 1 Knoten dürfen nicht übereinander liegen.
- 1 Spezifikation der Lage von einzelnen Knoten durch geometrische Regeln.
- 1 Stabilität des Layouts gegenüber kleinen Änderungen.
- 1 Erkennen von Grundstrukturen.

Die Zielfunktion F setzt sich dann entsprechend aus den sechs Termen additiv zusammen:

$$f = \sum_i a_i T_i$$

Jeder Term soll dabei so normiert werden, daß dieser im Wertebereich von 0 bis 1 liegt. Damit wird erreicht, daß bei allen Termen die Abweichungen vom Optimum etwa gleich bewertet wird. Soll eine Anforderung in jedem Fall eingehalten werden, ist dieser Term stärker zu wic-

ten. Dies erfolgt durch den Faktor α_i . Entsprechend der Konstruktion der Terme T_i ist das Minimum der Zielfunktion null.

5.7.1 Topologischer Abstand eines Graphen

Ein allgemeiner Graph $G(V, W)$ besteht aus N Knoten V und M Kanten W . Zusätzlich kann jede Kante noch ein Gewicht besitzen. Für das Layout des Graphen sollen die Knoten so positioniert werden, daß der vorgegebene Abstand zwischen zwei Knoten nach Möglichkeit dem geometrischen Abstand im Layout des Graphen entspricht (siehe obige Anforderung). Zunächst kann der topologische Abstand zwischen den Knoten nach dem Algorithmus von Warshall [War95] berechnet werden. Dieser Algorithmus baut auf der allgemeinen Adjazenzmatrix A auf, die angibt, welche Knoten direkt miteinander verbunden sind:

$$A=(a_{ij}) \quad \text{mit } a_{ij} = \begin{cases} 1 & \text{falls } V_i \text{ mit } V_j \text{ direkt verbunden ist} \\ 0 & \text{sonst} \end{cases}$$

Ausgehend davon werden N Matrizen E_v berechnet, deren Elemente $e_{i,j}$ jeweils angeben, ob der Knoten i mit dem Knoten j über w Kanten verbunden ist.

Mit der Anfangsmatrix:

$$E_0 = A$$

können die Matrizen E_v sukzessive folgendermaßen berechnet werden:

$$E_v[i,j] = E_{v-1}[i,j] \text{ OR } (E_{v-1}[i,v] \text{ AND } E_{v-1}[v,j])$$

Dabei kann der Algorithmus zur Verbesserung der Performance noch so modifiziert werden, daß die Elemente $e_{i,j}$ die Anzahl der Kanten zwischen den Knoten V_i und V_j angeben. Damit wird nur eine Matrix E benötigt.

5.7.2 Einhaltung einer vorgegebenen Kantenlänge

Die Realisierung des Visualisierungskonzeptes, inhaltliche Nähe durch räumliche Nähe auszudrücken, erfordert die Einhaltung der vorgegebenen Kantenlängen. Die Abweichung von der Vorgabe kann durch folgenden Term bewertet werden:

mit:

- d_{ij} : vorgegebene Länge
- l_{ij} : aktuelle euklidische Länge
- k : Anzahl der direkt verbundenen Knoten

ergibt sich:

$$t_{ij} = \begin{cases} \left(\frac{d_{ij} - l_{ij}}{d_{ij}} \right)^2 & \text{für } e_{ij} = 1 \\ 0 & \text{sonst} \end{cases}$$

$$T_1 = \frac{1}{k} \sum_i \sum_j t_{ij}$$

Die Differenzbildung in t_{ij} bewirkt, daß bei Erreichung der geforderten Länge dieser Teilterm den Wert null annimmt. Mit der Quadrierung werden positive und negative Abweichungen vom Optimum gleichbewertet. Damit die Abweichungen der Teilterme bei unterschiedlichem d_{ij} relativ in die Summe T_1 eingehen, wird jeder Teilterm mit dem Faktor $1/d_{ij}$ normiert. Alle Abweichungen von den Vorgaben werden in dem Term T_1 zusammengefaßt. Der Aufbau und die Interpretation des Terms T_1 , kann sinngemäß auf die weiteren Terme T_i übertragen werden.

Mit dem angegebenen Term T_1 werden alle Abweichungen von einer vorgegebenen festen Länge bewertet. Eine Modifikation der Formel erlaubt, einen Wertebereich s vorzugeben. Bei vielen Graphstrukturen kommt diese Vorgabe der menschlichen Vorstellung eines optimalen Layouts näher:

5.7.3 Topologische Abstände nicht direkt verbundener Knoten

Der im folgenden beschriebene Term bewertet alle die Knoten, die nicht direkt verbunden sind. Auch hier wird der geometrische Abstand zwischen zwei Knoten betrachtet. Es wird gefordert, daß dieser mindestens dem gewichteten topologischen Abstand d_{ij} entsprechen soll. Der Term kann analog zu Kapitel 5.7.2 bestimmt werden:

mit:

- d_{ij} : max Kantenlänge, die über die Topologie ermittelt wurde
- l_{ij} : aktuelle euklidische Länge
- k : Anzahl der nicht direkt verbundenen Knoten

ergibt sich:

5.7.4 Überdeckungsfreie Anordnung der Knoten

Das Layout eines Graphen führt zu Fehlinterpretationen, wenn einzelne Knoten auf der selben

$$t_{ij} = \begin{cases} \left(\frac{d_{ij} - l_{ij}}{d_{ij}}\right)^2 & \text{für } l_{ij} < d_{ij} \text{ und } e_{ij} \neq 1 \\ 0 & \text{sonst} \end{cases}$$

$$T_2 = \frac{1}{k} \sum_i \sum_j t_{ij}$$

Position angeordnet sind. Zur Vermeidung dieser Situation kann eine Überdeckung von Knoten ermittelt und bewertet werden. Die Überlappung zweier Knoten V_i und V_j wird über einen Rechteck-Clipping-Vergleich berechnet. Hierzu wird angenommen, daß jeder Knoten neben seiner Position eine rechteckige Ausdehnung besitzt. Die maximal überdeckte Fläche ist somit die Fläche a_{ij} des kleineren Knotens. Die Berechnung der Überlappung erfolgt in zwei Stufen. Zunächst kann überprüft werden, ob sich die Rechtecke der beiden Knoten in der x Richtung überlappen. Trifft dies zu, wird auch die y Richtung überprüft. Liegt in dieser Richtung ebenfalls eine Überlappung vor, so wird die Überdeckungsfläche c_{ij} berechnet

mit:

a_{ij} : maximal überdeckte Fläche
 c_{ij} : überdeckte Fläche
 n : Anzahl der Überdeckungen

ergibt sich:

$$t_{ij} = \begin{cases} \left(\frac{a_{ij} - c_{ij}}{a_{ij}}\right)^2 & \text{für } c_{ij} > 0 \\ 0 & \text{sonst} \end{cases}$$

$$T_3 = \frac{1}{n} \sum_i \sum_j t_{ij}$$

5.7.5 Spezifikation der Lage einzelner Knoten durch geometrische Regeln

Durch die Vorgabe von Regeln kann die relative Lage von Knoten spezifiziert werden. In Abhängigkeit von der Lage eines Knotens sind folgende Regeln möglich:

- 1 unter und über,
- 1 links und rechts,
- 1 vertikal und horizontal.

Zunächst wird überprüft, ob für die zwei zu untersuchenden Knoten eine Regel existiert. Ist dies der Fall, werden die Positionen untersucht, ob sie den Regeln entsprechen. Bei einer Verletzung wird die Abweichung bewertet. Dies erfolgt in allen Richtungen analog und wird im anhand des folgenden Beispiel veranschaulicht werden, bei dem gefordert wird, daß der Knoten V_2 rechts vom Knoten V_1 liegt.

Falls die x-Koordinate des Knoten V_2 kleiner ist, als die des Knotens V_1 , wird die folgende Fehlerfunktion berechnet. Soll weiterhin sichergestellt werden, daß der Knoten V_2 nicht nur rechts vom Knoten V_1 liegt, sondern eine Mindestentfernung zwischen den Knoten eingehalten wird, muß zusätzlich eine Schranke s definiert werden. Für den Teilterm gilt:

$$t_{ij_y} = \begin{cases} \left(\frac{(x_1 + s) - x_2}{1} \right)^2 & \text{für } x_1 + s > x_2 \\ 0 & \text{sonst} \end{cases}$$

Soll dagegen Knoten V_2 links vom Knoten V_1 dargestellt werden gilt:

$$t_{ij_y} = \begin{cases} \left(\frac{(x_1 + s) - x_2}{1} \right)^2 & \text{für } x_1 - s < x_2 \\ 0 & \text{sonst} \end{cases}$$

Entsprechende Teilterme gelten für die anderen Regeln und die Teilterme können dann addiert werden

5.7.6 Stabilität des Layouts gegenüber kleinen Änderungen

Dieses Verhalten wird bereits im wesentlichen durch den Optimierungsansatz erreicht, da bei einer Modifikation des Netzes der Optimierer die Positionen des vorhergehenden Graphen als Startposition verwendet. Hierdurch ist eine kontinuierliche Veränderung des Graphen zu einem neuen Optimum sichergestellt. Sind die Modifikationen gering, wird das neue Optimum in der Nähe des vorhergehenden liegen. Die Stabilität wird durch einen weiteren Effekt, im Fall des Öffnen eines Knotens, verbessert. Die Startpositionen der neuen Knoten werden kreisförmig um den zu öffnenden Knoten angeordnet. Danach wird zunächst nur die Positionen der neuen Knoten optimiert. Erst in einem weiteren Schritt werden wieder alle Knoten in die Optimierung mit einbezogen. Durch diese zweistufige Optimierung wird eine bessere Stabilität erreicht.

5.7.7 Einbeziehung von Grundstrukturen

Das Layout eines Graphen besteht im allgemeinen aus mehreren Grundstrukturen. Die Grundstrukturen können allgemeingültiger oder anwendungsspezifischer Natur sein. Allgemeingültige Grundstrukturen sind z.B. Zyklen, baum- oder sternförmige Anordnungen. Bei einer initialen Strukturanalyse können anhand einer Bibliothek die vorliegenden Grundstrukturen ermittelt werden. Die Grundstrukturen stellen dann Bausteine des Graphen dar, deren Layout im allgemeinen festliegt. Die Optimierung beschränkt sich dann auf die Ermittlung der Position der Bausteine. Für spezielle Anordnungen können die Grundstrukturen jedoch auch in die Optimierung mit einbezogen werden. Die bei der Optimierung sich einstellenden Abweichungen von den Grundstrukturen können dann über einen Term T_G bewertet werden. Mit diesem Ansatz werden Grundstrukturen optimal dargestellt und die benötigte Zeit für die Optimierung reduziert. Statt der derzeitigen zufälligen Anordnung des Graphen in der Initialisierungsphase kann die Bibliothek von Grundstrukturen auch zu einer Verbesserung der Anfangsposition genutzt werden.

5.8 Implementierung

Wie zu Anfang des Kapitels diskutiert, sind die existierenden Werkzeuge zum Bau semantischer Netze nicht für den Einsatz in einer web-basierten Lernumgebung geeignet. Aus diesem Grund wurde in den vorherigen Abschnitten ein Verfahren vorgestellt, daß ein vorhandenes Netz für einen Lerner oder einen Autoren einer Wissensbasis in geeigneter Weise präsentieren kann. Ähnliche Probleme ergeben sich bei der Auswahl einer Sprache für die Erstellung des semantischen Netz. Eine Auswahl und Bewertung von Sprachen findet sich unter [28]. Unabhängig von den detaillierten Möglichkeiten Anfragen, an das Netz zu stellen, kommen die Autoren zu dem Schluß, daß beim Einsatz eines semantischen Netz in einer web-basierten Umgebung, auf jeden Fall auch eine web-basierte Sprache zur Modellierung verwendet werden sollte. Das Problem hierbei ist, daß die web-basierten Sprachen wie RDF Schema oder OIL bisher nur Entwurfscharakter haben und keine Implementierungen vorliegen. Entscheidend für die Auswahl einer Sprache ist also vielmehr, ob eine Trennung von Schema und

semantischem Netz möglich ist und ob sich beide Datenstrukturen als XML-Beschreibung speichern lassen. Ist dies der Fall, können sowohl Sprache als auch Netz zu einem späteren Zeitpunkt in eine webbasierte Sprache transformiert werden. Ein weiterer Punkt ist die mögliche Erweiterbarkeit der Software, die den Aufbau und die Präsentation des Netzes, realisiert. Da bei der Verwendung in einer webbasierten Lernumgebung nicht nur reine Begriffsnetze verwendet werden, sondern an die einzelnen Begriffe komplette Metadatenbeschreibungen angehängt werden müssen, kann keine Implementierung verwendet, die nur eine reine Verwaltung von Begriffen ermöglicht. Eine Software, die alle der eben genannten Forderungen erfüllt und deshalb für die Realisierung des Concept Space-Editors gewählt wurde, ist das von GMD IPSI entwickelte Smalltalk Frame Kit (SFK) [42]

5.8.1 SFK

Das SFK ist ein Framesystem, das in der objektorientierten Programmiersprache Smalltalk implementiert ist. Das SFK ist in die Klassenstruktur von Smalltalk eingebettet und stellt eine Erweiterung dar. Mit dem SFK können Strukturen aufgebaut werden, die Objekte beziehungsweise Klassen verwalten. Dazu werden Smalltalk-Klassen und spezielle im SFK angebotene Klassen verwendet. Die Klassen des SFK heißen Frame-Klassen und sind in einer Hierarchie angeordnet. Es ist keine multiple Vererbung möglich, so dass eine Frame-Klasse immer nur von einer Frame-Klasse direkt erben kann. Die Vererbung der Klasseigenschaften verhindert Redundanzen in den Definitionen der Frame-Klassen und ermöglicht so die praktische Wiederverwendung von Klasseigenschaften. Sowohl die Eigenschaften der Frame-Klassen, als auch die Relationen zwischen den Frame-Klassen werden in Slots gespeichert. Die Slots der Instanzen der Frame-Klassen sind mit Werten gefüllt, was nicht bedeutet, dass alle Slots einer Instanz gefüllt sein müssen. Ist diesbezüglich keine Einschränkung für einen Slot angegeben, dann kann ein Slot mit mehreren Werten gefüllt werden. Die Menge dieser Werte wird *ValueCollection* genannt. Die Eigenschaften der Slots werden in *Facets* angegeben. Soll mit dem SFK ein semantisches Netz aufgebaut werden, dann muß zunächst ein Schema implementiert werden. Dieses Schema kann dadurch erzeugt werden, daß direkt entsprechende Smalltalk-Klassen realisiert werden. Es ist zudem auch möglich eine, in einer proprietären XML-Beschreibung gespeicherte, Schemabeschreibung einzulesen und daraus die entsprechenden Smalltalkklassen automatisch generieren zu lassen. In [89] wird eine Untersuchung vorgenommen, inwieweit das SFK den Anforderungen an Sprachen zum Aufbau semantischer Netze genügt. Ohne hier auf Details einzugehen, kann festgestellt werden, daß es mit dem SFK möglich ist, sowohl Framesysteme, als auch Beschreibungslogiken zu realisieren. Zudem bietet das SFK neben seiner freien Verfügbarkeit im Quellcode, die Möglichkeit der Verwaltung des semantischen Netz in Form der impliziten Datenhaltung und der Möglichkeit Backups des Netzes zu erstellen.

die in [ROBB] beschrieben ist. Aufgrund der Verwendung von XML als Datenaustauschformat zur Kommunikation zwischen ConceptSpace-Editor und SFK, kann die Anwendung ohne Probleme auf ein RDF Schema oder OIL basiertes semantisches Netz umgestellt werden, sobald die Spezifikation dieser Sprachen abgeschlossen ist. In einem solchen Fall muß auf der Seite des ConceptSpace-Editors nur die Komponente Konsistenzprüfung geändert werden, die für die Abbildung des Netzes auf die im ConceptSpace-Editor verwendeten Datenstrukturen zur Verwaltung des Graphen, zuständig ist. Im folgenden soll nur auf die beiden Komponenten *SchemaVisualisierung* und *GraphVisualisierung* näher eingegangen werden. Eine detaillierte Beschreibung der anderen Komponenten samt Diskussion der verwendeten Datenstrukturen zur Verwaltung des Graphen findet sich unter [92]. Während die Komponente *GraphVisualisierung* für die eigentliche Darstellung der Knoten und Kanten verantwortlich ist, zeigt die Komponente *SchemaVisualisierung* das dem semantischen Netz zugrundeliegende Schema dem Benutzer in Form von Menüs zur Auswahl von Knoten- und Kantentypen.

5.8.2 GraphVisualisierung

Die Implementierung des Layout-Algorithmus wurde inspiriert durch eine Entwicklung eines in [66] beschriebenen Layout-Algorithmus (AVE) zur Visualisierung von Informationsnetzen, der als Prototyp für ein automatisches Layout einer Zeitungsseite genutzt wurde. Durch eine Strukturbeschreibung des Graphen kann das Layout berechnet werden. Zusätzlich können Regeln eingebracht werden, die die relative horizontale und die vertikale Ausrichtung für ausgewählte Knoten unter einander festlegen. Der Algorithmus ist ein Optimierungsverfahren nach der Spring-Model-Methode. Die Implementierung dieses Layout-Algorithmus lag in der Programmiersprache C vor. Als Grundlage für diese Arbeit erfolgte zunächst eine Transformation des Codes in die Programmiersprache Java. Ähnlich wie bei den bereits diskutierten Verfahren ist es auch mit diesem Layout-Algorithmus nicht möglich, dargestellte Graphen, so zu erweitern, daß der visuelle Eindruck für einen Betrachter bei Änderungen des Graphen stabil bleibt. Für eine interaktive Navigationssteuerung ist ein unempfindliches Verhalten bei Änderungen aber essentiell, damit der Graph beim Löschen oder Einfügen von Knoten nur lokale Veränderungen erfährt, jedoch das globale Aussehen nicht verändert wird. Um ein solches Verhalten bei AVE zu erreichen, wurden zunächst Regeln über die relative Lage der Knoten bestimmt. Die abgeleiteten Regeln dienten dann als Restriktion bei der Neuberechnung eines erweiterten Graphen. Es zeigte sich, daß dieses Vorgehen zu restriktiv war. Die Regeln erzwingen immer das Einhalten der Regel, selbst geringe Abweichungen wurden nicht toleriert. Im Sinne der benötigten Stabilität, ist zum Beispiel die Vorgabe, der Knoten *A* liege links von Knoten *B*, als erfüllt anzusehen, wenn sich die Knoten noch ein wenig überlappen. Dieses unscharfe Verhalten ist jedoch mit der vorliegenden Implementierung nicht möglich. Weiterhin zeigte sich, daß der AVE-Algorithmus einige vorgegebene Beispielgraphen nicht berechnen konnte. Diese Ergebnisse führten dazu, daß nur die Grundprinzipien des AVE-Algorithmus verwendet werden konnten. Das Grundprinzip beruht auf der Optimierung der Layout-Funktion. In der vorliegenden Implementierung von AVE wurde Optimierung und die Funktionsberechnung aus Effizienzgründen in einem Modul realisiert. Jedoch wurde durch diese Laufzeitverbesserung die Verständlichkeit so verschlechtert, daß eine Modifikation des AVE ausschied. Aufgrund dieser Erfahrung wurde bei der Neuimplementierung Funktionswertberechnung und Optimierung funktional getrennt, damit ist ein Austausch der beiden Module durch verbesserte oder andere Ansätze leicht möglich.

Öffnet ein Benutzer bei der Navigation auf dem semantischen Netz einen Knoten, so stellt sich

die Frage, wann der neu berechnete Graph auf der Visualisierungsfläche gezeichnet werden soll. Wird mit der Präsentation gewartet, bis die Layoutberechnung abgeschlossen ist, ergeben sich zwei Probleme:

1 Performance

Obwohl der Layoutalgorithmus aufgrund der verwendeten Datenstrukturen eine effiziente und damit performante Positionierung durchführen kann, können bis zum Finden des Optimums einige Sekunden vergehen. Wenn während dieser Zeit auf der Visualisierungsfläche keine Aktionen stattfinden, besteht die Gefahr, dass der Benutzer entweder ungeduldig wird, oder der Ansicht ist, daß Probleme bei der Visualisierung des Graphen aufgetreten sind.

1 Veränderte Lage der Knoten

Obwohl der Positionierungsalgorithmus Anordnungen von Knoten bevorzugt, die sich nur wenig von der vorherigen Anordnung unterscheiden, muß nach dem Öffnen eines Knoten für die neu hinzugekommenen Knoten Platz auf der Visualisierungsfläche geschaffen werden. Damit wird es aber unumgänglich daß zumindest die Knoten, die eine Verbindung zu den neu hinzugekommenen Knoten haben, bei der neuen Berechnung eine andere Position zugewiesen bekommen. Wird also nur das Endergebnis der Layoutberechnung visualisiert, führt dies dazu, daß bestimmte Knoten, Sprünge bei der Visualisierung aufweisen, was im schlimmsten Fall dazu führen kann, daß der Benutzer die Knoten nicht mehr auf Anhieb auf der Oberfläche findet.

Eine Lösung dieses Problem stellt ein Verhalten des Layoutalgorithmus dar, daß dem bekannten Prinzip einer Explosionseichnung ähnelt. Ein Knoten wird bei diesem Verfahren nicht erst bei Beendigung der Layoutberechnung präsentiert, sondern kontinuierlich während der Berechnung. Für den Benutzer ist bei diesem Verfahren zum einen permanent erkennbar, daß die Anwendung dabei ist, ein neues Layout zu berechnen. Zum anderen kann der Weg jedes Knoten von seiner Startposition bis zur endgültigen Position, direkt beobachtet werden, so dass einzelne Knoten nicht mehr für das Auge des Benutzers verloren gehen können. Eine Schwierigkeit bei der Umsetzung dieses Prinzips stellt allerdings die Arbeitsweise des Algorithmus dar. Obwohl die Forderung nach Stabilität des Layouts auch bei kleinen Änderungen dazu führt, daß der Optimierungsalgorithmus möglichen Anordnungen von Knoten beim Öffnen von Knoten und der dadurch notwendigen Neupositionierung des Graphen, einen sehr niedrigen Funktionswert zuweist, werden sie vom Algorithmus als mögliche Kandidaten in eine erste Berechnung mit einbezogen. Erst nach einer gewissen Zeit, fängt der Algorithmus an, sich an ein Optimum anzunähern und die Position der Knoten für die verschiedenen Möglichkeiten ändert sich nur noch geringfügig. Die Visualisierung der einzelnen Ergebnisse der Layoutberechnung können daher erst dann auf der Oberfläche präsentiert werden, wenn die Änderungen bei den einzelnen Versionen unter einen bestimmten Schwellenwert gefallen sind.

Der Zugriff auf die den einzelnen Begriffen zugeordneten Metadatenbeschreibungen erfolgt über ein PopUp-Menü, daß nach Auswahl eines Knoten zur Verfügung steht. Wie aus Abbildung 48 ersichtlich ist, verfügt die *GraphVisualisierungs*-Komponente über eine Schnittstelle zum LOM-Kommunikationsprotokoll. Über diese Schnittstelle können daher Suchanfragen nach Metadatenbeschreibungen abgesetzt werden. Standardmäßig werden die Bezeichnungen für den Bezeichnungen für den ausgewählten Begriff und die Bezeichnungen der unmittelbar benachbarten Knoten als Parameter für die Suchanfrage verwendet. Im Gegensatz zu der im vorherigen Kapitel beschriebenen Präsentation der Suchergebnisse in einem Web-Browser, werden im Fall einer Anfrage durch den ConceptSpace-Editor, die Suchergebnisse nicht nach

HTML transformiert, sondern vielmehr direkt als XML-Datenstrom an den Editor zurückgegeben. Sie können dort dann dem Benutzer in Listenform präsentiert werden. Wählt der Benutzer eine Metadatenbeschreibung aus dieser Liste aus, wird der LOM-Editor gestartet und die vollständige Metadatenbeschreibung angezeigt.

5.8.3 SchemaVisualisierung

Da Erstellung des Schemas und Knüpfen des semantischen Netzes zwei getrennte Aufgaben sind, die von unterschiedlichen Rollen bearbeitet werden, müssen diese Aufgaben nicht über eine einheitliche Oberfläche angeboten werden. Der Experte in Wissensmodellierung kann daher direkt das Schema im SFK implementieren und die Regeln für das semantische Netz festlegen. Er hat damit die Zugriff auf die gesamte Funktionalität des SFK und kann die Regeln und die Begriffstypen sehr detailliert spezifizieren. Damit aufgrund dieses Schemas ein Erstellen des Netzes möglich ist, muß dieses Schema separat gespeichert werden, damit es danach vom ConceptSpace-Editor eingelesen werden kann. Für die Kodierung des Schemas wurde eine XML-DTD spezifiziert. Eine entsprechende XML-Beschreibung des Schemas kann beim Start des Editors vom lokalen Dateisystem oder vom SFK angefordert werden.

5.8.4 Benutzeroberfläche des ConceptSpace-Editor

Abbildung 49 zeigt die Benutzeroberfläche des Editors. Die SchemaVisualisierungs-Komponente besteht aus den Menüleisten auf der linken Seite und zeigt die möglichen Begriffstypen und die Relationen des initialisierten Schemas. Bei dem semantischen Netz, das in der Abbildung gezeigt ist, handelt es sich um ein Netz, das den Bereich der Infektiologie modelliert und das im Projekt Medibook [81] entstanden ist. Über eine Suchschnittstelle, die nicht in der Abbildung gezeigt ist, kann der Benutzer nach bestimmten Begriffen suchen, die dann als Knoten angezeigt werden. Besitzen diese Knoten bei der Darstellung einen Halbkreis, der die obere Hälfte des Knoten umgibt, wird dem Benutzer dadurch angezeigt, daß dieser Knoten Verbindungen zu anderen Knoten hat, die derzeit nicht visualisiert sind. In der Abbildung findet sich dieser Halbkreis bspw. bei dem Begriff *Darminfektion*. Durch einfaches Doppelklicken, können diese Knoten geöffnet werden. Dies bedeutet, daß alle mit diesem Knoten verbundenen weiteren Knoten und die entsprechenden Relationen angezeigt werden. Um das Netz übersichtlich zu halten, können bestimmte Relationen oder Begriffstypen gefiltert werden. Diese Filterung erfolgt über die Deaktivierung eines bestimmten Begriffs- oder Relationstyp und ist ebenfalls in Abbildung 48 gezeigt. Man erkennt, daß in der linken Menüleiste der Begriffstyp *Zelle* mit einer hellgrauen Farbe gezeigt ist. Wird nach Deaktivierung eines Begriffstyps ein Knoten geöffnet, werden Begriffe auf die das Filterkriterium zutrifft nicht in die Visualisierung aufgenommen. Dem Benutzer wird durch einen Halbkreis, der in diesem Fall die untere Hälfte des Knotens umgibt angezeigt, daß der Knoten noch andere Verbindungen zu anderen Knoten hat, diese aber nicht angezeigt werden. Wird ein Filter für einen Begriffs- oder Relationstyp aktiviert und befinden sich auf der Visualisierungsfläche derzeit noch entsprechende Knoten oder Kanten, so werden diese bei der nächsten Positionierung des Graphen ebenfalls hellgrau eingefärbt und treten daher visuell gegenüber den anderen Knoten in den Hintergrund. Solche Knoten werden als erste Kandidaten herangezogen, wenn entschieden werden soll, welche Knoten während der Navigation auf dem semantischen Netz wieder gelöscht werden sollen. Übersteigt die Anzahl an neuen Knoten einen vom Benutzer eingestellten Schwellwert, kann er explizit angeben, welche Knoten neu in die Visualisierung aufge-

nommen werden sollen. Dieser Schwellenwert gibt zusätzlich auch die maximale Anzahl an Knoten an, die gleichzeitig auf dem Bildschirm angezeigt werden. Der Grund hierfür ist, daß die Darstellung des Netzes unübersichtlich wird, wenn ständig nur neue Knoten auf der Darstellungsfläche hinzugefügt werden. Ein weiterer Grund ist die Performance des Layout-Algorithmus. Je mehr Knoten bei der Positionierung der Knoten und bei der Optimierung des Layouts mit in die Berechnung einbezogen werden, desto länger benötigt der Algorithmus die optimale Position der Knoten zu berechnen. Die Anzahl der Knoten, die noch problemlos angezeigt und berechnet werden kann, ist also sehr stark von der zur Verfügung stehenden Fläche und damit von der Größe des Monitors und der Leistungsfähigkeit des Rechners abhängig. Es kann daher für den Maximalwert kein absoluter Wert angegeben haben. Untersuchungen mit verschiedenen Plattformen, Rechnern und Bildschirmen haben gezeigt, daß ein Wert der bei ungefähr 30 gleichzeitig visualisierten Knoten liegt, zu befriedigenden Ergebnissen führt. Bei 30 Knoten erhält der Benutzer ausreichend Information über die Umgebung eines Knoten und die Visualisierung kann auf allen Systemen so schnell erfolgen, daß eine interaktive Navigation möglich ist. Dennoch ist dieser Maximalwert beim Start konfigurierbar und kann auch während der Arbeit mit dem System vom Benutzer geändert werden. Übersteigt die Summe der Anzahl neuer und der bereits visualisierten Knoten den eingestellten Schwellenwert, werden automatisch Knoten von der Oberfläche entfernt nach der in Kapitel 5.6.1 beschriebenen Strategie entfernt. Neue Knoten können vom Benutzer durch Drag&Drop aus der linken Menüleiste hinzugefügt werden. Werden Kanten zwischen Knoten gezogen, wird überprüft, ob die Beziehung regelkonform mit dem Schema des semantischen Netz ist und zeigt sie entweder an oder weist den Vorgang zurück.

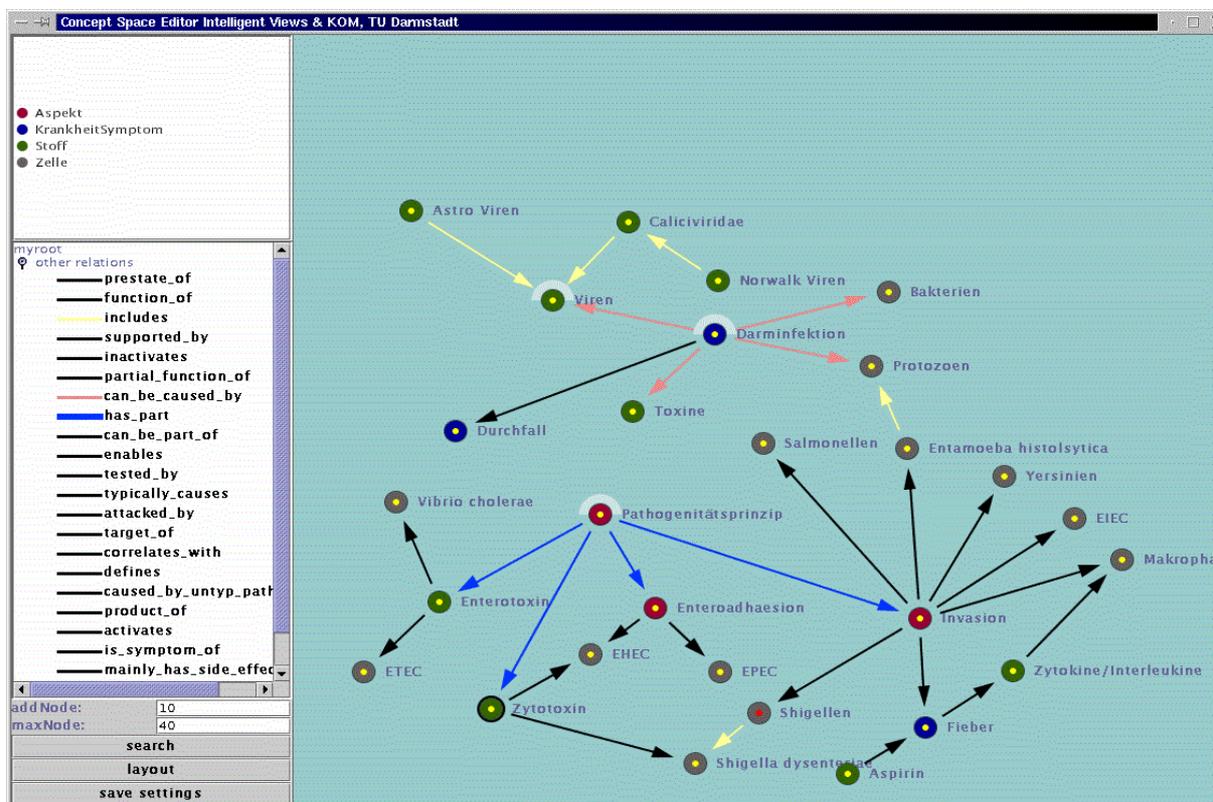


Abbildung 49: GUI des ConceptSpace-Editors

6 Zusammenfassung und Ausblick

Durch die Schaffung virtueller Studiengänge, die vollständig web-basiert organisiert sind, dem verstärkten Einsatz multimedialer Lernmittel als Ergänzung zu Präsenzveranstaltungen und der Aussicht, Material, das tagtäglich von Bildungseinrichtungen erstellt wird, kommerziell zu verwenden, ist das Interesse an unabhängigen, modularen und wiederverwendbaren Medienbausteinen sehr stark angestiegen. Medienbausteine bestehen dabei nicht nur aus der eigentlichen Ressource. Vielmehr ist jeder Ressource eine eindeutige Metadatenbeschreibung zugeordnet. Durch Analyse der Metadatenbeschreibung kann eine Ressource nicht nur in der zunehmenden Informationsflut im Web gefunden werden. Die Ressource kann zudem auf ihre Eignung hin beurteilt werden, ohne daß ein direkter Zugriff auf die Ressource nötig ist

Obwohl Metadaten kein neues Konzept darstellen, zeigt die in dieser Arbeit vorgenommene Untersuchung, daß für die Kodierung der Beschreibungen bisher kein einheitliches Vorgehen existiert und von einer einfachen XML-Kodierung durch die Spezifikation einer DTD, bis zum Aufbau komplexer semantischer Strukturen durch die Verwendung von RDF Schema, derzeit mehrere Lösungsansätze diskutiert werden. Bisher keine Berücksichtigung bei der Auswahl eines Datenmodells und eines Kodierungsschema findet die Frage, wer für die Erstellung der Beschreibungen verantwortlich ist. Zwar können insbesondere web-basierte Lernsysteme von semantischen Informationen, die über den Bezug zu einer einzelnen Ressource hinaus allgemeine Informationen speichern, profitieren; die Erstellung und Verwaltung dieser Information sollte aber unter Berücksichtigung der am Erstellungsprozeß beteiligten Rollen, separat von der Beschreibung einzelner Medienbausteine erfolgen. Aus diesem Grund wird in dieser Arbeit ein geteiltes Datenmodell vorgeschlagen, das die Metadatenbeschreibungen der Ressourcen mit einer einfachen XML-Beschreibung kodiert und für die semantischen Informationen, ein eigenes separates Modell, das der semantischen Netze, verwendet. Aufbauend auf diesem Modell wurde eine Architektur für ein web-basiertes Lernsystem entwickelt, das als zentralen Bestandteil eine Wissensbasis enthält, die die Metadatenbeschreibungen und das semantische Netz verwaltet. Ausgehend von dieser Architektur, wurden Anforderungen an Werkzeuge zur Erstellung und Pflege der Wissensbasis erarbeitet. Umgesetzt wurden diese Anforderungen durch die Implementierung des Concept Space Editors zur Erstellung des semantischen Netzes und des LOM-Editors für die Beschreibung der Medienbausteine mit Metadaten. Die Vielzahl von Projekten in denen der LOM-Editor seit seiner Freigabe eingesetzt wird (vgl. die Auflistung in Abschnitt 3.8.4 "Werkzeuge zur Erstellung der Metadaten") zeigt den Bedarf an Werkzeugen zur Erstellung von Metadaten für web-basierte Ressourcen und die Richtigkeit des in dieser Arbeit vorgestellten Datenmodells.

Neben der Entwicklung von Werkzeugen zur Erstellung und Verwaltung von Metadaten und den semantischen Netzen, sind die Suchmöglichkeiten ein entscheidendes Kriterium für die Nutzung einer Wissensbasis. Während für den Zugriff auf ein semantisches Netz, mit dem Concept-Space Editor ein Werkzeug entwickelt wurde, daß sowohl eine gezielte Suche nach einzelnen Begriffen, als auch eine interaktive Navigation ermöglicht, gilt dies für den LOM-Editor nicht. Es ist mit der derzeitigen Oberfläche nicht möglich Suchkriterien mit booleschen Operationen zu verknüpfen, oder für das gleiche Metadatenelement verschiedene Suchkriterien zu spezifizieren. Auch die in Abbildung 31 in Abschnitt 4.2.2 "Suchoberfläche" gezeigte web-basierte Suchoberfläche ermöglicht keine explizite Verknüpfung der Suchkriterien, sondern geht implizit von einem logischen AND aus. Zwar ist es möglich Suchoberflächen zu

bauen (vgl. [93]), die die interaktive Erstellung einer Suchanfrage ermöglichen; diese Oberflächen setzen allerdings zumindest Grundkenntnisse in der Anfragesprache voraus, die von einem normalen Benutzer nicht erwartet werden können. Dies stellt allerdings kein spezifisches Problem einer Suche nach web-basierten Lernmaterialien dar. Wie ein Blick auf die Suchoberfläche gängiger Suchmaschinen zeigt, existiert für dieses Problem derzeit auch für allgemeine Suchanfragen keine befriedigende Lösung.

Ein anderes Problem, das sich bei der Suche mit Hilfe von LOM-Beschreibungen ergibt, ist die Menge an Metadatenelementen von LOM und damit die Menge der möglichen Suchkriterien, die von einem Benutzer angegeben werden können. Da alle Projekte in denen der LOM-Editor eingesetzt wird, noch nicht abgeschlossen sind, stehen bisher keine Erfahrungsberichte von Benutzern zur Verfügung, welche der Metadatenelemente von LOM am häufigsten zur Suche eingesetzt werden. Untersuchungen, wie sie derzeit im Projekt k-med durchgeführt werden, können helfen, die Suchoberfläche übersichtlicher zu gestalten, indem nur die wichtigsten Elemente direkt in die Oberfläche mit aufgenommen werden und weniger benötigte Elemente nur optional für eine Suche präsentiert werden [67].

Die in dieser Arbeit entworfene Architektur geht von einer zentralen Wissensbasis aus, in der sämtliche Metadaten verwaltet werden. Denkbar ist aber auch, daß lokale Repositories entstehen, die unter Umständen Metadatenbeschreibungen thematisch verwandter Ressourcen oder Ressourcen eines bestimmten Typs, verwalten. Dies erfordert die Entwicklung eines Anfrageprotokolls für verteilte Wissensbasen. Neben bekannten Ansätzen, wie bsp. dem ISO-Standard Z39.50 für den Netzwerkzugriff auf verteilte Informationen, sind hier vor allem die unter dem Label "Santa Fe Convention" zusammengefaßten Spezifikationen der Open Archives Initiative (OAI) interessant. Das primäre Bemühen der OAI konzentriert sich auf den Zugriff auf Archive, die Veröffentlichungen in elektronischer Form enthalten. Da zur Beschreibung der Publikationen Metadaten wie bsp. Dublin Core zum Einsatz kommen, bieten sich diese Protokolle für den Zugriff auf eine Wissensbasis die LOM-Beschreibungen verwaltet an. Da sich die Spezifikationen der Zugriffsprotokolle allerdings noch in der Entwicklung befindet, können derzeit bestenfalls Prototypen realisiert werden. Die Realisierung eines solchen Prototyp ist in [94] beschreiben. Steht eine stabile Version der OAI-Protokolle zur Verfügung, kann die Architektur angepaßt werden, sodas ein transparenter Zugriff auf verteilte Repositories möglich wird.

Eine weiterer Aspekt der Verteilung, der genauer untersucht werden sollte, ist, neben der Verteilung der Metadatenbeschreibungen, die Verwaltung und Verteilung der tatsächlichen Ressourcen. Die Verwaltung der Ressourcen ermöglicht nicht nur die Garantie, daß die Ressource, für die eine Metadatenbeschreibung vorhanden ist, auch tatsächlich existiert; zudem können detaillierte Zugriffsrechte ermöglicht werden. Ein weiterer Vorteil einer Verwaltung der Ressourcen liegt darin, daß durch den Einsatz verteilter Dateisysteme oder anderer Replikationsstrategien eine lokale Verfügbarkeit der Ressourcen ermöglicht werden kann. Untersucht werden muß daher inwieweit die in dieser Arbeit entwickelten Modelle und Architekturen mit Modellen für eine verteilte Speicherung und Zugriffskontrolle, wie sie bsp. in [76] untersucht und beschreiben sind, kombiniert werden können.

Anhang A: Document Type Definition für LOM

```
!ELEMENT LANGUAGE (#PCDATA)*>
<!ELEMENT LANGSTRING (#PCDATA)>
<!ATTLIST LANGSTRING LANG CDATA #IMPLIED>
<!ELEMENT STRING (#PCDATA)*>
<!ELEMENT CATALOGUE (#PCDATA)*>
<!ELEMENT DATETIME (#PCDATA)*>
<!ELEMENT METADATAScheme (#PCDATA)*>
<!ELEMENT SIZE (#PCDATA)*>
<!ELEMENT LOCATION (#PCDATA)*>
<!ELEMENT MINIMUMVERSION (#PCDATA)*>
<!ELEMENT MAXIMUMVERSION (#PCDATA)*>
<!ELEMENT SEMANTICDENSITY (VOCABULARY)?>
<!ELEMENT SOURCE (#PCDATA)*>
<!ELEMENT ID (#PCDATA)*>
<!ELEMENT IDENTIFIER (#PCDATA)*>
<!ELEMENT VCARD (#PCDATA)*>
<!ELEMENT VOCENTRY (#PCDATA)*>
<!ELEMENT DETAIL (LANGSTRING?)>
<!ELEMENT VOCABULARY (VOCENTRY?, DETAIL?)>
<!ELEMENT COST (VOCABULARY?)>
<!ELEMENT COPYRIGHTOROTHERRESTRICTIONS (VOCABULARY?)>
<!ELEMENT DIFFICULTY (VOCABULARY?)>
<!ELEMENT AGGREGATIONLEVEL (VOCABULARY?)>
<!ELEMENT INTENDEDENDUSERROLE (VOCABULARY*)>
<!ELEMENT INTERACTIVITYLEVEL (VOCABULARY)?>
<!ELEMENT INTERACTIVITYTYPE (VOCABULARY?)>
<!ELEMENT FORMAT (LANGSTRING*)>
<!ELEMENT STATUS (VOCABULARY?)>
<!ELEMENT STRUCTURE (VOCABULARY?)>
<!ELEMENT CENTITY (VCARD?)>
<!ELEMENT PERSON (VCARD?)>
<!ELEMENT ENTRY (LANGSTRING*)>
<!ELEMENT PURPOSE (VOCABULARY?)>
<!ELEMENT TITLE (LANGSTRING?)>
<!ELEMENT DESCRIPTION (LANGSTRING*)>
<!ELEMENT KEYWORDS (LANGSTRING*)>
<!ELEMENT COVERAGE (LANGSTRING*)>
<!ELEMENT VERSION (LANGSTRING?)>
<!ELEMENT TYPE (VOCABULARY?)>
<!ELEMENT NAME (VOCABULARY?)>
<!ELEMENT INSTALLATIONREMARKS (LANGSTRING?)>
<!ELEMENT OTHERPLATFORMREQUIREMENTS (LANGSTRING?)>
<!ELEMENT ROLE (VOCABULARY?)>
<!ELEMENT LEARNINGRESOURCETYPE (VOCABULARY?)>
<!ELEMENT CONTEXT (VOCABULARY*)>
<!ELEMENT TYPICALAGERANGE (LANGSTRING*)>
<!ELEMENT KIND (VOCABULARY?)>
<!ELEMENT TAXON (ID?, ENTRY?)>
<!ELEMENT DATE (DATETIME?, DESCRIPTION?)>
<!ELEMENT DURATION (DATETIME?, DESCRIPTION?)>
<!ELEMENT TYPICALLEARNINGTIME (DATE?)>
<!ELEMENT TAXONPATH (SOURCE?, TAXON*)>
<!ELEMENT CATALOGENTRY (CATALOGUE?, ENTRY?)>
<!ELEMENT CONTRIBUTE (ROLE?, CENTITY*, DATE?)>
<!ELEMENT RESOURCE (IDENTIFIER?, TITLE?, DESCRIPTION?, CATALOGENTRY*)>
<!ELEMENT REQUIREMENTS (TYPE?, NAME?, MINIMUMVERSION?, MAXIMUMVERSION?)>
<!ELEMENT ANNOTATION (PERSON?, DATE?, DESCRIPTION?)>
<!ELEMENT CLASSIFICATION (PURPOSE?, TAXONPATH*, DESCRIPTION?, KEYWORDS*)>
<!ELEMENT EDUCATIONAL (INTERACTIVITYTYPE?, LEARNINGRESOURCETYPE*,
```

```
INTERACTIVITYLEVEL?, SEMANTICDENSITY?, INTENDEDENDUSERROLE*, CONTEXT*,
TYPICALAGERANGE*, DIFFICULTY?, TYPICALLEARNINGTIME?, DESCRIPTION?, LANGUAGE?)>
<!ELEMENT GENERAL (IDENTIFIER?, TITLE?, CATALOGENTRY*, LANGUAGE*, DESCRIPTION*,
KEYWORDS*, COVERAGE*, STRUCTURE?, AGGREGATIONLEVEL?)>
<!ELEMENT LIFECYCLE (VERSION?, STATUS?, CONTRIBUTE*)>
<!ELEMENT METAMETADATA (CATALOGENTRY*, CONTRIBUTE*, METADATAScheme*, LANGUAGE?)>
<!ELEMENT RELATION (KIND?, RESOURCE?, IDENTIFIER?)>
<!ELEMENT RIGHTS (COST?, COPYRIGHTOROTHERRESTRICTIONS?, DESCRIPTION?)>
<!ELEMENT TECHNICAL (FORMAT?, SIZE?, LOCATION*, REQUIREMENTS*,
INSTALLATIONREMARKS?, OTHERPLATFORMREQUIREMENTS?, DURATION?)>
<!ELEMENT RECORD (METAMETADATA?, GENERAL?, LIFECYCLE?, TECHNICAL?, EDUCATIONAL?,
RIGHTS?, RELATION*, ANNOTATION*, CLASSIFICATION*)>
```

Anhang B: Grundschemata der deutschen Übersetzung des LOM-Entwurfs

Nummer	Name	Beschreibung	Struktur	Format	Typ	Beispiel
1	Allgemein	Diese Kategorie enthält allgemeine Informationen, die diese Ressource als solche beschreiben.	Einzelne Instanzen	-	-	-
1.1	Kennziffer	Eine globale, eindeutige Kennzeichnung, mit der sich die Ressource identifizieren lässt. Dieses Datenelement wird (noch) nicht verwendet, da es noch keinen spezifizierte Methode für die Erschaffung einer globalen, eindeutigen Kennzeichnung gibt	Einzelner Wert	-	reserviert	-
1.2	Titel	Der Name, der dieser Ressource gegeben wurde	Einzelner Wert	-	Sprach-String	-

Tabelle 6 Kategorienschema des LOM-Entwurfs

Nummer	Name	Beschreibung	Struktur	Format	Typ	Beispiel
1.3	Katalog eintrag	Dieses Datenelement beschreibt einen Eintrag dieser Ressource in einen Katalog (also ein Verzeichnis zur Identifizierung). Diese Unterkategorie beschreibt die Ressource im Zusammenhang mit bekannten Katalogsystemen, so daß extern nach ihr gesucht werden kann, und man sie gemäß der Methodik des speziellen Systems lokalisieren kann. Diese Unterkategorie soll die Funktion des bisherigen Elementes 1.1:General.Identifier ersetzen BEMERKUNG: Einige der Katalogeinträge können automatisch von Indizierungswerkzeugen generiert werden..	Mehrere ungeordnete Instanzen (minmax: 10 Elemente)	-	-	-
1.3.1	Katalog	Der Name des Katalogs (Des Verzeichnisses zur Identifizierung)	Einzelner Wert	ISO 10646-1	String	ISBN, Ariadne
1.3.2	Eintrag	Aktueller String Wert des Eintrags im Katalog.	Einzelner Wert			14-06-93

Tabelle 6 Kategorienschema des LOM-Entwurfs

Nummer	Name	Beschreibung	Struktur	Format	Typ	Beispiel
1.4	Sprache	Die hauptsächlichste Sprache (oder Sprachen), die in dieser Ressource zur Kommunikation mit dem Benutzer benutzt wird (bzw. werden). BEMERKUNG: Ein Indizierungswerkzeug könnte eine sinnvolle Grundeinstellung anbieten	Ungeordnete Liste (minmax: 10 Elemente)	SprachID = Langcode ('-'Subcode)*, wobei Langcode ein Sprachcode aus zwei Buchstaben ist, gemäß ISO 639 und Subcode ein Ländercode gemäß ISO 3166. BEMERKUNG: Diese Vorgehensweise ist kompatibel mit dem xml:lang Attribut und wird definiert durch RFC1766 BEMERKUNG: ISO 639 beschäftigt sich mit »antiken« Sprachen, wie Griechisch oder Latein. Der Sprachcode sollte in Kleinbuchstaben, und der Ländercode (falls vorhanden) in Großbuchstaben geschrieben werden. 'Keine' sollte auch ein möglicher Wert sein	String	"en", "en-GB", "de", "fr-CA", "it"
1.5	Beschreibung	Eine schriftliche Beschreibung des Inhalts der Ressource.	Ungeordnete Liste (minmax: 10 Elemente)	-	Sprach-String	-
1.6	Schlagwörter	Schlagwörter oder Redewendungen, die diese Ressourcen beschreiben. Dieses Element sollte nicht für Charakterisierungen benutzt werden, die mit anderen Elementen beschrieben werden könnten.	Ungeordnete Liste (minmax: 10 Elemente)	-	Sprach-String	-

Tabelle 6 Kategorienschema des LOM-Entwurfs

Nummer	Name	Beschreibung	Struktur	Format	Typ	Beispiel
1.7	Hintergrund	Ein zeitlicher, kultureller, geographischer oder regionaler Rahmen, der auf diese Ressource zutrifft.	Ungeordnete Liste (minmax: 10 Elemente)	-	Sprach-String	Circa, 16. Jahrhundert, Frankreich
1.8	Struktur	Die zu Grunde liegende Organisationsstruktur dieser Ressource.	Einzelner Wert	Eingeschränktes Vokabular: 3=Sammlung 4=Gemischt 5=Linear 6=Hierarchisch 7=Vernetzt 8=Verzweigt 9=Aufgeteilt 10=Atomar	Vokabular	-
1.9	Granularität	Die Zusammensetzung einer Ressource.	Einzelner Wert	Eingeschränktes Vokabular: 0 = Hohe Granularität, z.B. unbearbeitete Mediendaten oder Fragmente. 1 = Eine Ansammlung von atomaren Bausteinen, z.B. ein HTML Dokument mit eingebetteten Bildern, oder eine Lektion. 2 = Eine Ansammlung von Stufe 1 Ressourcen, z.B. ein 'Netz' von HTML Dokumenten mit einem Inhaltsverzeichnis, das die Seiten zu einer Einheit verbindet. 3 = Höchste Geschlossenheit, niedrigste Granularität, z.B. ein Kurs	Sprach-String	-

Tabelle 6 Kategorienschema des LOM-Entwurfs

Nummer	Name	Beschreibung	Struktur	Format	Typ	Beispiel
2	Lebenszyklus	Diese Kategorie beschreibt die Geschichte und den derzeitigen Zustand der Ressource, sowie diejenigen, die zu dieser Ressource einen Beitrag geleistet haben.	Einzelner Wert	-	-	-
2.1	Version	Die Version dieser Ressource	Einzelner Wert	-	Sprach-String	3.0 vorläufig
2.2	Status	Der Status oder Zustand dieser Ressource	Einzelner Wert	Eingeschränktes Vokabular: 3=Entwurf 4=Endversion 5=Korrigierte Version 6=Nicht verfügbar	Vokabular	-
2.3	Beitrag	Dieses Datenelement nennt die Personen oder Organisationen, die zu dieser Ressource einen Beitrag geleistet haben BEMERKUNG: Dieses Element unterscheidet sich von 3.3:MetaMetadaten.Beitrag	Mehrere ungeordnete Instanzen (minmax: 10 Elemente)	-	-	-

Tabelle 6 Kategorienschema des LOM-Entwurfs

Nummer	Name	Beschreibung	Struktur	Format	Typ	Beispiel
2.3.1	Rolle	In welcher Funktion wurde der Beitrag zur Ressource geliefert. Wenigstens der Autor/ die Autoren sollte(n) hier spezifiziert werden.	Einzelner Wert	Empfohlene, bisher Verwendete Ressourcen Liste: 3=Autor 4=Verleger 5=Unbekannt 6=Initiator 7=Verfasser der Endversion 8=Verantwortlicher 9=Herausgeber 10=Grafikdesigner 11=Technischer Implementierer 12=Inhalt Lieferant 13=Technisch Verantwortlicher 14=Didaktiker 15=Drehbuchautor 16=Verantwortlich für das Instruktionsdesign	Vokabular	-

Tabelle 6 Kategorienschema des LOM-Entwurfs

Nummer	Name	Beschreibung	Struktur	Format	Typ	Beispiel
2.3.2	Person/ Organisation	Die Identifizierung und Beschreibung der Personen oder Organisationen, die Beiträge geleistet haben. Die relevantesten zuerst. Wenn 2.3.1:Lebenszyklus.Beitrag.Rolle gleich Autor ist, dann soll Person/Organisation eine Person oder Personen sein. Wenn 2.3.1: Lebenszyklus.Beitrag.Rolle gleich Verleger ist, dann soll Person/Organisation eine Organisation sein. Wenn Person/Organisation eine Organisation ist, dann sollte es das universitäre Institut, die Agentur, die Firma, die Institution, usw. sein in deren Verantwortung der Beitrag gemacht wurde	Geordnete Liste (min-max: 40 Elemente)	vCard	String	-
2.3.3	Datum	Das Datum des Beitrags	Einzelner Wert	-	Datum	-
3	MetaMeta-Daten	Diese Kategorie beschreibt die spezifischen Informationen über die Metadatenansammlung an sich. Das sind keine Informationen über die Resource, die diese Ansammlung beschreibt	Einzelner Wert	-	-	-

Tabelle 6 Kategorienschema des LOM-Entwurfs

Nummer	Name	Beschreibung	Struktur	Format	Typ	Beispiel
3.1	Kennziffer	Eine globale, eindeutige Kennzeichnung, mit der sich die Metadatenansammlung identifizieren läßt. Dieses Datenelement wird (noch) nicht verwendet, da es noch keine spezifizierte Methode für die Erschaffung einer globalen, eindeutigen Kennzeichnung gibt	Einzelner Wert	-	reserviert	-
3.2	Katalogeintrag	Dieses Datenelement beschreibt einen Eintrag dieser Metadaten in einen Katalog (also ein Verzeichnis zur Identifizierung). Diese Unterkategorie beschreibt die Metadaten im Zusammenhang mit bekannten Katalogsystemen, so daß extern nach ihnen gesucht werden kann, und man sie gemäß dem speziellen System lokalisieren kann. Dieses Element kann als Ersatz für das bisherige 3.1:MetaMeta-Data. Identifiziert werden BEMERKUNG: Einige der Katalogeinträge können automatisch von Indizierungswerkzeugen erschaffen werden.	Mehrere ungeordnete Instanzen	-	-	-

Tabelle 6 Kategorienschema des LOM-Entwurfs

Nummer	Name	Beschreibung	Struktur	Format	Typ	Beispiel
3.2.1	Katalog	Der Name des Kataloges (Des Verzeichnisses zur Identifizierung). BEMERKUNG: Wird normalerweise vom System erzeugt.	Einzelner Wert	ISO 10646-1	String	Ariadne
3.2.2	Eintrag	Aktueller String Wert des Eintrags im Katalog BEMERKUNG: Dieses Element wird normalerweise vom System erzeugt.	Mehrere geordnete Instanzen	-	-	-
3.3	Beitrag	Dieses Datenelement beschreibt diejenigen Personen, oder Organisationen, die zum Status dieser Metadaten einen Beitrag geleistet haben. BEMERKUNG: Dieses Element unterscheidet sich von 2.3:Lebenszyklus.Beitrag	Mehrere geordnete Instanzen	-	-	-
3.3.1	Rolle	In welcher Funktion wurde der Beitrag zu den Metadaten geliefert. Genau ein Erzeuger sollte angegeben sein.	Einzelner Wert	Offenes Vokabular. Empfohlen: 3=Erzeuger 4=Verantwortlicher	Vokabular	-
3.3.2	Person/ Organisation	Die Identifizierung und Beschreibung der Personen oder Organisationen, die zu diesen Metadaten Beiträge geleistet haben. Die relevantesten zuerst	Geordnete Liste	vCard	String	-
3.3.3	Datum	Das Datum des Beitrags	Einzelner Wert	-	Datum	-

Tabelle 6 Kategorienschema des LOM-Entwurfs

Nummer	Name	Beschreibung	Struktur	Format	Typ	Beispiel
3.4	Metadatenformat	Der Name und die Version der autorisierten Spezifikation, nach der diese Metadaten Instanzen erschaffen wurden. BEMERKUNG: Dieses Element kann vom Benutzer bestimmt oder vom System generiert werden. Bei mehreren Einträgen müssen die Metadaten auch verschiedenen Metadatenformaten entsprechen.	Ungeordnete Liste	ISO 10646-1	String	LOM-1.0
3.5	Sprache	Sprache dieser Metadateninstanz. Das ist die DefaultEinstellung für alle SprachString Werte bei diesen Metadateninstanzen	Einzelner Wert	siehe 1.4:Allgemein. Sprache BEMERKUNG: Für dieses Datenelement ist der Eintrag 'Keine' nicht möglich	String	-
4	Technische Details	Diese Kategorie beschreibt die technischen Details und Grundvoraussetzungen für diese Ressource.	Einzelne Instanzen	-	-	-
4.1	Format	Der technische Datentyp dieser Ressource (und aller ihrer Komponenten). Dieses Element wird benötigt, um festzulegen, welche Software benötigt wird, um die Ressource darzustellen.	Ungeordnete Liste	MIME Typen oder 'non-digital'	String	video/ mpg text/html

Tabelle 6 Kategorienschema des LOM-Entwurfs

Nummer	Name	Beschreibung	Struktur	Format	Typ	Beispiel
4.2	Größe	Die Größe der digitalen Ressource in Bytes. Nur die Ziffern '0'...'9' dürfen benutzt werden; die Einheit ist Bytes, nicht Mbytes, GB, usw. Dieses Element bezieht sich auf die aktuelle Größe dieser Ressource, nicht die Größe einer komprimierten Version der Ressource.	Einzelner Wert	ISO 10646-1 (aber nur mit den Ziffern '0'..'9')	String	
4.3	Adresse	Ein String, den man benötigt, um Zugriff zu der Ressource zu haben. Entweder eine Adresse (z.B. Universal Resource Locator) oder ein Verfahren, eine Adresse zu erhalten (z.B. Universal Resource Identifier). Die bevorzugte Adresse zuerst. Der Ort, wo sich die Lernressource, die mit diesen Metadaten beschrieben wird, physikalisch befindet.	Geordnete Liste	ISO 10646-1	String	

Tabelle 6 Kategorienschema des LOM-Entwurfs

Nummer	Name	Beschreibung	Struktur	Format	Typ	Beispiel
4.4	Voraussetzungen	Diese Unterkategorie beschreibt die technische Ausstattung, die benötigt wird um die Ressource zu benutzen. Wenn es mehrere Voraussetzungen gibt, dann werden alle benötigt, d.h. die logische Verbindung ist UND.	Mehrere ungeordnete Instanzen	-	-	-
4.4.1	Typ	Die technische Ausstattung, die benötigt wird, um diese Ressource zu benutzen, d.h. Hardware, Software, Netzwerk	Einzelner Wert	Offenes Vokabular. Empfohlen: 3=Betriebssystem 4=Browser	Vokabular	-
4.4.2	Name	Name der benötigten technische Ausstattung BEMERKUNG: Der Wert dieses Elementes kann automatisch aus 4.1:Technische Details.Format abgeleitet werden. Aus »video/mpeg« folgt z.B. »Verschiedene-Betriebssysteme«	Einzelner Wert	Wenn Typ='Betriebssystem', dann ist zu empfehlen: 3=PC-DOS 4=MS-Windows 5=MacOS 6=Unix 7=Verschiedene-Betriebssysteme 8=Kein Wenn Typ='Browser', dann ist zu empfehlen: 100=Beliebig 101=Netscape Communicator 102=Microsoft Internet Explorer 103=Opera 104=Amaya Bei anderen Typen: Offenes Vokabular	Vokabular	-
4.4.3	Minimale Version	Version der technischen Ausstattung, die mindestens für diese Ressource benötigt wird	Einzelner Wert	ISO 10646-1	String	-

Tabelle 6 Kategorienschema des LOM-Entwurfs

Nummer	Name	Beschreibung	Struktur	Format	Typ	Beispiel
4.4.4	Maximale Version	Neueste Version, von der bekannt ist, daß sie die Ressource unterstützt	Einzelner Wert	ISO 10646-1	String	-
4.5	Installationshinweise	Beschreibung, wie die Ressource zu installieren ist.	Einzelner Wert	-	Sprach-String	
4.6	Weitere technische Ausstattung	Information über weitere Hardware und Software, die benötigt wird.	Einzelner Wert	-	Sprach-String	Soundkarte
4.7	Dauer	Die Zeit, die eine ununterbrochene Ressource bei der für sie vorgesehenen Geschwindigkeit läuft. Dieses Element ist besonders nützlich für Sounddateien, Filme oder Animationen.	Einzelner Wert	-	Datum	PT1H30M, PT1M45S
5	Pädagogische Details	Diese Kategorie beschreibt die pädagogischen Charakteristika dieser Ressource. Es handelt sich um die pädagogischen Informationen, die entscheidend sind für alle, die eine qualitativ bessere Lehre erreichen wollen. Die Zielgruppe für dieses Metadaten sind Lehrer, Manager, Autoren und Lernende.	Einzelne Instanzen	-	-	-

Tabelle 6 Kategorienschema des LOM-Entwurfs

Nummer	Name	Beschreibung	Struktur	Format	Typ	Beispiel
5.1	Art der Interaktion	<p>Wieviel Interaktion gibt es zwischen Ressource und Benutzer.</p> <p>In einer erklärenden Ressource fließt die Information fast nur von der Ressource zum Lernenden.</p> <p>Erklärende Dokumente werden vor allem für Learning-by-reading eingesetzt.</p> <p>In einer aktiven Ressource fließt die Information zusätzlich vom Lernenden zur Ressource. Aktive Ressourcen werden für Learning-by-doing eingesetzt.</p> <p>Da Benutzen von Links in Hypertexten stellt keinen Informationsfluß dar. Daher sind Hypertextdokumente erklärend.</p>	Einzelner Wert	Eingeschränktes Vokabular 3=Aktiv 4=Erklärend 5=Gemischt 6=Undefiniert	Vokabular	<p>Erklärende Dokumente sind z.B. Essays, Video-clips, alle Arten von graphischem Material und Hypertext-dokumente.</p> <p>Aktive Dokumente sind z.B. Simulationen, Fragebögen und Übungen.</p>
5.2	Ressourcentyp	Verschiedene Formate, die in der Ressource auftauchen. Die häufigsten zuerst	Geordnete Liste	<p>Offenes Vokabular. Empfohlen:</p> <p>3=Übung 4=Simulation 5=Fragebogen 6=Diagramm 7=Bild 8=Graph 9=Index 10=Folie 11=Tabelle 12=Erzählender Text 13=Prüfungstext 14=Experiment 15=Problembeschreibung 16=Selbstbewertung</p>	Vokabular	-

Tabelle 6 Kategorienschema des LOM-Entwurfs

Nummer	Name	Beschreibung	Struktur	Format	Typ	Beispiel
5.3	Interaktionsstufe	Der Grad der Interaktion zwischen dem Benutzer und der Ressource.	Einzelner Wert	Eingeschränktes Vokabular: 0=sehr wenig 1=wenig 2=mittel 3=hoch 4=sehr hoch	String	-
5.4	Semantische Dichte	Die Menge an Information, die diese Ressource enthält, im Verhältnis zu ihrer Größe und Dauer.	Einzelner Wert	Eingeschränktes Vokabular: 0=sehr wenig 1=wenig 2=mittel 3=hoch 4=sehr hoch	String	-
5.5	Zielgruppe	Die Zielgruppe(n), für die diese Ressource entwickelt wurde. Die dominierendste zuerst. Ein Lerner arbeitet mit einer Ressource um etwas zu lernen. Ein Autor erschafft, oder veröffentlicht eine Ressource. Ein Verwalter stellt eine Ressource zur Verfügung, z.B. eine Universität. Das Dokument für einen Verwalter ist typischerweise ein Lehrplan. BEMERKUNG: Wenn man die Zielgruppe durch die erforderlichen Fähigkeiten oder Eigenschaften des Nutzers definieren will, sollte die Kategorie 9:Klassifikation benutzt werden.	Geordnete Liste	Eingeschränktes Vokabular: 3=Lehrer 4=Autor 5=Lerner 6=Verwalter	Vokabular	-

Tabelle 6 Kategorienschema des LOM-Entwurfs

Nummer	Name	Beschreibung	Struktur	Format	Typ	Beispiel
5.6	Kontext	Die hauptsächliche Umgebung, in der das Lernen und der Gebrauch dieser Ressource stattfinden soll.	Ungeordnete Liste	Offenes Vokabular. Empfohlen 3=Grundschule 4=Mittelstufe 5=Oberstufe 6=Universität Grundstudium 7=Universität Hauptstudium 8=Universität Promotionstudium 9=Fachhochschule Grundstudium 10= Fachhochschule Hauptstudium 11=Professionelle Weiterbildung 12= Kontinuierliche Weiterbildung 13=Berufliche Weiterbildung	Vokabular	-

Tabelle 6 Kategorienschema des LOM-Entwurfs

Nummer	Name	Beschreibung	Struktur	Format	Typ	Beispiel
5.7	Alterstufe	<p>Das Alter der repräsentativen Zielgruppe</p> <p>Dieses Element sollte sich auf die Entwicklungsstufe beziehen, wenn diese von der Altersstufe verschieden ist.</p> <p>BEMERKUNG: Das Alter des Lernenden ist vor allem bei Kindern wichtig, um für sie geeignete Ressourcen zu finden.</p> <p>Wenn möglich, sollte der String als 'Mindestalter-Höchster', oder 'Mindestalter-' formatiert sein.</p> <p>(BEMERKUNG: Das ist ein Kompromiß zwischen dem Wunsch nach drei Feldern (minAlter, maxAlter und Beschreibung) und der Verfügbarkeit von nur einem Textfeld.)</p> <p>BEMERKUNG: Die verschiedenen Kategorien für Altersstufen, IQ oder Entwicklungsstufe sollten in der Kategorie 9:Klassifikation auftauchen</p>	Ungeordnete Liste	-	Sprach-String	0-5,7-9, 15, 18-, geeignet für Kinder ab 7, nur für Erwachsene
5.8	Schwierigkeit	Dieses Element definiert, wie schwer es für die Zielgruppe ist, sich durch die Ressource zu arbeiten.	Einzelner Wert	Eingeschränktes Vokabular: 0=sehr leicht 1=leicht 2=mittel 3=schwer 4=sehr schwer	String	-

Tabelle 6 Kategorienschema des LOM-Entwurfs

Nummer	Name	Beschreibung	Struktur	Format	Typ	Beispiel
5.9	Durchschnittliche Lernzeit	Geschätzte oder durchschnittliche Zeit, die man benötigt, um sich durch die Ressource zu arbeiten	Einzelner Wert	-	Datum	
5.10	Beschreibung	Bemerkung, wie man die Ressource zu benutzen hat.	Einzelner Wert	-	Sprach-String	Richtlinie für Lehrer, in Verbindung mit einem Lehrbuch
5.11	Sprache	Die natürliche Sprache, die von der Zielgruppe gesprochen wird. BEMERKUNG: Für eine Ressource in Französisch, die sich an deutschsprachige Benutzer richtet (Z.B. ein Sprachkurs) wäre der Wert von 1.4:Allgemein.Sprache Französisch, der von 5.11:Pädagogisch. Sprache Deutsch .	Ungeordnete Liste	Siehe 1.4:Allgemein. Sprache	String	"en", "en-GB", "de", "fr-CA", "it"
6	Rechte	Diese Kategorie beschreibt die Urheberrechte und Nutzungsbedingungen für diese Ressource. BEMERKUNG: Hier sollen die Ergebnisse der andauernden Arbeit auf dem Gebiet der Urheberrechte im Hinblick auf das Internet berücksichtigt werden. Gegenwärtig bietet diese Kategorie nur das absolute Minimum an Details.	Einzelne Instanzen	-	-	-

Tabelle 6 Kategorienschema des LOM-Entwurfs

Nummer	Name	Beschreibung	Struktur	Format	Typ	Beispiel
6.1	Kosten	Gibt an, ob man für den Gebrauch dieser Ressource bezahlen muß.	Einzelner Wert	Eingeschränktes Vokabular: 3=ja 4=nein	Vokabular	-
6.2	Urheberrecht und andere Einschränkungen	Gibt an, ob diese Ressource Urheberrechten, oder anderen rechtlichen Einschränkungen unterliegt.	Einzelner Wert	Eingeschränktes Vokabular: 3=ja 4=nein	Vokabular	-
6.3	Beschreibung	Erklärung der rechtlichen Einschränkungen.	Mehrere ungeordnete Instanzen	-	Sprach-String	-
7	Verwandte Ressource	Diese Kategorie definiert die Beziehungen zwischen dieser Ressource und anderen, wenn vorhanden.. Um verschiedene Beziehungen darzustellen, kann es verschieden Instanzen in dieser Kategorie geben. Wenn es mehr als eine Zielressource gibt, muß für jedes Ziel eine eigenen Beziehungsinstanz angegeben sein.	Mehrere ungeordnete Instanzen	-	-	-

Tabelle 6 Kategorienschema des LOM-Entwurfs

Nummer	Name	Beschreibung	Struktur	Format	Typ	Beispiel
7.1	Art	Die Art der Beziehung zwischen dieser Ressource und der in 7.2: Verwandte Ressource. Ressource angegeben.	Einzelner Wert	Empfohlen, nach Dublin Core: 3=Ist ein Teil von 4=Enthält Elemente 5=Ist eine Version von 6=Enthält Versionen 7=Ist ein Format von 8=Hat das Format 9=Verweist zu 10=Wird darauf verwiesen von 11=Hat als Grundlage 12=Ist Grundlage für 13=Benötigt 14=Wird benötigt von	Vokabular	-
7.2	Ressource	Die Zielressource, auf die diese Beziehung verweist.	Einzelner Wert	-	-	-
7.2.1	Kennziffer	Eindeutige Kennziffer der Zielressource. Kann und soll (noch) nicht benutzt werden.	Einzelner Wert	-	reserviert	-
7.2.2	Beschreibung	Beschreibung der Zielressource.	Einzelner Wert	-	Sprach-String	-
7.2.3	Katalogeintrag	Siehe 1.3: Allgemein. Katalogeintrag.	Mehrere ungeordnete Instanzen	-	-	-
8	Anmerkung	Diese Kategorie enthält Anmerkungen bezüglich des pädagogischen Gebrauchs dieser Ressource, wer diese Anmerkung gemacht hat und wann. Wenn mehrere Anmerkungen vorliegen, können mehrere Instanzen dieser Kategorie verwendet werden.	Mehrere ungeordnete Instanzen	-	-	-

Tabelle 6 Kategorienschema des LOM-Entwurfs

Nummer	Name	Beschreibung	Struktur	Format	Typ	Beispiel
8.1	Person	Die Person, die diese Anmerkung gemacht hat.	Einzelner Wert	vCard	String	-
8.2	Datum	Das Datum, wann die Anmerkung gemacht wurde.	Einzelner Wert	-	Datum	-
8.3	Beschreibung	Der Inhalt der Anmerkung	Einzelner Wert	-	Sprach-String	-
9	Klassifikation	Diese Kategorie beschreibt, welcher Aspekt dieser Ressource sich auf ein bestimmtes Klassifikationssystem bezieht. Um mehrere Klassifikationen angeben zu können, kann man mehrere Instanzen in diese Kategorie eintragen.	Mehrere ungeordnete Instanzen	-	-	-
9.1	Zweck	Was man an dieser Ressource klassifiziert hat.	Einzelner Wert	Offenes Vokabular. Empfohlen: 3=Fachgebiet 4=Thema 5=Voraussetzungen 6=Lernziele 7=Zugriffsbedingungen 8=Pädagogisches Niveau 9=Fähigkeiten 10=Sicherheitsstandard	Vokabular	-
9.2	Taxonomischer Pfad	Diese Unterkategorie beschreibt den Klassifikationspfad in einem bestimmten Klassifikationssystem. Es kann im selben System verschiedene Pfade geben, die dieselbe Ressource beschreiben.	Mehrere ungeordnete Instanzen	-	-	-

Tabelle 6 Kategorienschema des LOM-Entwurfs

Nummer	Name	Beschreibung	Struktur	Format	Typ	Beispiel
9.2.1	Quelle	<p>Der Name des Klassifikationssystems. Dieses Element kann sowohl »offizielle« Taxonomien, als auch selbstdefinierte akzeptieren. BEMERKUNG Ein Indizierungswerkzeug könnte sicherstellen, daß die etablierten Klassifikationen (LOC, UDC, DDC, usw.) zuerst eingetragen werden.</p>	Einzelner Wert	ISO 10646-1	String	ACM, MESH, ARI-ADNE

Tabelle 6 Kategorienschema des LOM-Entwurfs

Nummer	Name	Beschreibung	Struktur	Format	Typ	Beispiel
9.2.2	Eintrag	Diese Unterkategorie beschreibt einen genauen Ausdruck in einer Taxonomie. Ein Taxon ist ein Knoten mit einer genauen Bezeichnung oder Definition. Ein Taxon kann auch eine alphanumerische Kennzeichnung haben, für Standardverweise. Die Kennzeichnung oder der Ausdruck können einzeln oder zusammen verwendet werden, um den Taxon zu bestimmen. Eine geordnete Liste von Taxons stellt einen Taxonpfad dar, d.h. Eine 'Taxon Treppe', also ein Pfad von einem allgemeinen zu einem immer spezielleren Eintrag in der Klassifikation. Ein Taxonpfad sollte eine Tiefe zwischen 1 und 9 haben, wobei die durchschnittliche Tiefe bei 2 bis 4 liegt.	Mehrere geordnete Instanzen	-	-	Physik/ Akustik/ Instrumente/ Stethoskop, Medizin/ Diagnose/ Instrumente/ Stethoskop
9.2.2.1	Kennung	Die Kennziffer des Taxon. Eine Buchstaben- oder Zahlenkombination die der Quelle der Taxonomie entspricht.	Einzelner Wert	ISO 10646-1	String	320, 4.3.2 , BF180
9.2.2.2	Eintrag	Die textliche Bezeichnung des Taxon.	Einzelner Wert	-	Sprach-String	medizinische Wissenschaft

Tabelle 6 Kategorienschema des LOM-Entwurfs

Nummer	Name	Beschreibung	Struktur	Format	Typ	Beispiel
9.3	Beschreibung	Das ist die Beschreibung der Ressource, dem 9.1:Klassifikation. Zweck des Klassifikationssystems entsprechend, also Fachgebiet, Anspruch, usw.	Einzelner Wert	-	Sprach-String	Ein medizinisches Instrument zum Hören, Stethoskop genannt.
9.4	Schlagwörter	Die Schlagwörter oder Phrase, die diese Ressource dem 9.1:Klassifikation. Zweck des Klassifikationssystems entsprechend beschreiben	Geordnete Liste	-	Sprach-String	-

Tabelle 6 Kategorienschema des LOM-Entwurfs

SprachString

Nummer	name	Beschreibung	Struktur	Format	Typ	Beispiel
1	SprachString	String in einer oder mehrerer natürlicher Sprachen.	Mehrere ungeordnete Instanzen	-	-	-
1.1	Sprache	Natürliche Sprache, in der der String verfaßt wurde.	Einzelner Wert	Wenn keine Sprache angegeben ist, dann sollte Sprach-String.String interpretiert werden wie 3.5:Meta-Dataen.Sprache.	String	"en", "en-GB", "de", "fr-CA", "it"

Nummer	name	Beschreibung	Struktur	Format	Typ	Beispiel
1.2	String	Tatsächlicher Stringwert. Ein String sollte immer wenigstens einen Buchstaben enthalten. Für interne Operationen können Strings der Länge Null zugelassen werden, diese dürfen aber nicht von Elementen ohne Wert zu unterscheiden sein.	Einzelner Wert	ISO 10646-1	String	

Tabelle 7 Schema für LOM SprachString

Datum

Nummer	name	Beschreibung	Struktur	Format	Typ	Beispiel
1	Zeit	Das Datum, nach dem ISO 8601 Standard formatiert.	Einzelner Wert	ISO 8601	String	1999-06-22
2	Beschreibung	Beschreibung des Datums	Einzelner Wert	-	Sprach-String	circa 1300 v.Chr., Wintersemester 1999

Tabelle 8 Schema für LOM Datum

Vokabular

Nummer	name	Beschreibung	Struktur	Format	Typ	Beispiel
1	Eintrag	"Benutzerdefiniert", "Gemäß_Klassifikation", oder ein Eintrag aus dem Vokabular des Datenelementes.	Einzelner Wert	ISO 10646-1	String (minmax: 4 Zeichen) 1=Benutzerdefiniert 2=Gemäß_Klassifikation oder Werte, wie in den Schemata des Datenelementes definiert.	
2	Detail	Zusätzlich Detail des Eintrages vom Typ Vokabular.	Einzelner Wert	-	Sprach-String	

Tabelle 9 Schema für LOM Vokabular

Literaturverzeichnis

- [1] Apache XML Project, *Aapache Cocoon*, <http://xml.apache.org>, 2001
- [2] Advanced Distributed Learning, *Sharable Courseware Object Reference Model (SCORM)*, <http://www.adlnet.org/SCORM/docs>
- [3] ARIADNE Project, *The Ariadne Foundation*, <http://www.ariadne-eu.org>, 2001
- [4] Beaumont I. *User Modelling in the Interactive Anatomy Tutoring System ANATOM-TUTOR*, Adaptive Hypertext and Hypermedia, Kluwer Academic Publishers 1998, S. 91-117
- [5] Bach M. *XSL und XPath - Verständlich und praxisnah*, Addison Wesley Verlag, 2000
- [6] Baumert A. *Kritische Grammatik*, Doktorarbeit, Fakultät der Geistes- und Sozialwissenschaften der Universität Kanner, 1989
- [7] Beard K. *A strucutre for organizing metadata collection*, http://ncgia.ucsb.edu/conf/SANTA_FE_CD-ROM/sf_papers/beard_kate/metadatapaper.html
- [8] Bechhofer S., et-al. *An informal description of Standard OIL and Instance OIL*, <http://www.ontoknowledge.org/oil/download/oil-whitepaper.pdf>, November 2000
- [9] Berners-Lee T., Hendler J. Lassila O., *The Semantic Web*, Scientific American, <http://www.scientificamerican.com/2001/0501issue/0501berners-lee.html>, 2001
- [10] Bild der Wissenschaft, *Studenten lernen am leichtesten von Webangeboten, die im Aufbau an Bücher erinnern*, bild der Wissenschaft online, <http://www.wissenschaft.de/sixcms/detail.php?id=89486>, 2001
- [11] Blumstengel A. *Entwicklung hypermedialer Lernsysteme*, Dissertation Universität Paderborn, http://dsor.uni-paderborn.de/organisation/blum_diss/, 1998
- [12] Boll S., Klas W. and Sheth A. *Overview on Using Metadata to manage multimedia data*, In: W. Klas and A. Sheth (hrsg.): *Multimedia Data Management - using Metadata to Integrate and Applay Digital Media*. McGraw-Hill, Hightown, USA, 1998
- [13] Bourret R, *XML and databases*, <http://www.rpbouret.com/xml/XMLandDatabases.htm>, 2000
- [14] Brase J. *Entwurf für eine deutsche Version der Learning Objects Metadata*, <http://www.cenorm.be/iss/workshop/lt/LOM-localization/default.htm>, 2000
- [15] Bray T. *RDF and Metadata*, O'Reilly xml.com, <http://www.xml.com/pub/a/98/06/rdf.html>, 1999

- [16] Brickley D., Guha R.V. *Resource Description Framework (RDF) Schema Specification 1.0*, W3C Candidate Recommendation, <http://www.w3.org/TR/2000/CR-rdf-schema-20000327>, März 2000
- [17] Brin S., Page L. *The Anatomy of a large-Scale Hypertextual Web Search Engine*, Proceedings of the 7th WWW conference, Brisbane Australien, 1998
- [18] Brusilowsky P. Schwarz E., Weber G. *ELM-ART: An intelligent tutoring system on World Wide Web*, Intelligent Tutoring Systems Lecture Notes in Computer Science Vol.1986, Springer Verlag Berlin 1996, Seite 261-269
- [19] Bund-Länder Kommission, *Globalisierung des Bildungsmarktes durch Neue Medien - Auswirkungen auf die Hochschulen*, Heft 81 der BLK-Reihe Materialien zur Bildungsplanung und zur Forschungsförderung, ISBN 3-934850-02-2, 2000
- [20] Bund-Länder Kommission, *Multimedia in der Hochschule*, Heft 85 der BLK-Reihe Materialien zur Bildungsplanung und zur Forschungsförderung, ISBN 3-934850-08-1, Juni 2000
- [21] Card S., Robertson G., Mackinlay J. *The Information visualizer, an information workspace*, In Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems, Information Visualization, Seite 181-188, 1991
- [22] CASTOR, *Open Source Data Binding Framework for Java*, <http://castor.exolab.org>
- [23] Charlesworth, J.P.A., Garner P.N. *Sopken content metadata and MPEG-7*, Proceedings of the ACM multimedia 2000 workshops, S.81-84
- [24] Caumanns J. *Automatisierte Komposition von wissensvermittelnden Dokumenten für das World Wide Web*, Dissertation Brandenburgische Technische Universität Cottbus, 2000
- [25] CEN/ISSS, *Learning Technologies Workshop LOM Localization*, <http://www.cenorm.be/iss/workshop/lt/LOM-localization/default.htm>, 2001
- [26] CLIX, *Corporate Learning & Information eXchange*, <http://www.im-c.de/clix.htm>, 2001
- [27] Computerwoche, *Online-Lernen gewinnt an Fahrt*, Computerwoche 22/2001
- [28] Corcho O. Gomez-Perez A. *Evaluating Knowledge Representation and Reasoning Capabilities of Ontology Specification Languages*, Proceedings of the EACI-00 Workshop on Applications of Ontologies and Problem Solving Methods, August 2000, Berlin Germany
- [29] DCMI, *Dublin Core Metadata Element Set Version 1.1*, Reference Description, <http://purl.org/dc/elements/1.1>
- [30] De Bra P. *Design Issues in Adaptive Hypermedia Application Development*, Proceedings of the 2nd Workshop on Adaptive Systems and User Modeling on the World Wide Web, Toronto Mai

- 1999
- [31] delite, *Virutal Information and Knowledge Environmentes*, <http://darmstadt.gmd.de/delite>
- [32] Decker S. et al. *The Semantic Web: The Roles of XML and RDF*, Internet Computing
- [33] Tadzebao and WebOnto, *Discussing, Browsing and Editing Ontologies on the Web*, 11th Knowledge Acquisition for Knowledge-Based Systems Workshop Banff, Canada, April 1998
- [34] Däßler R., Palm H., *Virtuelle Informationsräume mit VRML*, d.punkt Verlag Heidelberg, 1997
- [35] Der Duden, *Das Fremdwörterbuch*, Duden Verlag Mannheim, 1997
- [36] Duinevald A., Stoter R., Weiden M., Kenepa B., Benjamins V.R. *Wondertools? A comparative study of ontological engineering tools*, Proceedings of the Twelfth Workshop on Knowledge Acquisition, Modeling and Management <http://sern.ucalgary.ca/KSI/KAW/KAW99/papers/Duinevald1/wondertools.pdf>, 1999
- [37] Endres A., Fellner D.W., *Digitale Bibliotheken Informatik Lösungen für globale Wissensmärkte*, dpunkt.verlag Heidelberg, 2000
- [38] Farquhar A., Fikes R., Rice J., *The Ontolingua Server: A Tool for Collaborative Ontology Construction*, Knowledge Systems Laboratory, ftp://ftp.ksl.stanford.edu/pub/KSL_Reports/KSL-96-26.ps, 1996
- [39] Faatz A. et. al, *Multimedia und Wissen: Unser Weg zu einem produktiven Umgang mit Wissensdurst*, Thema Forschung Technische Universität Darmstadt 2/00, 2000
- [40] Fensel D. et.al, *OIL in a Nutshell*, Proceedings of the EACI-00 Workshop on Applications of Ontologies and Problem Solving Methods, August 2000, Berlin Germany
- [41] Ferber R. *Data Mining and Information Retrieval*, Skript zur Vorlesung an der TU Darmstadt, WS 98/99
- [42] Fischer D., Rostek L. *SFK: A Smalltalk Frame Kit: Concepts and Use*, <http://www.darmstadt.gmd.de/~rostek/sfkman3.1/index.htm>, 1997
- [43] Furrie B. *Undertstanding MARC Bibilographics: Machine Readable Cataloging* ,<http://lcweb.loc.gov/marc/umb/>, 2000
- [44] Gallenbacher J. *Web komplett*, Computer & Literatur, 2000
- [45] Ghavam A. *Applying IEEE's LOM and W3C's RDF to a multimedia learning system*, Diplomarbeit KOM, TU Darmstadt, September 1999
- [46] Grosso. W., Erikson H., Ferguson R.W. et. al, *Knowledge Modeling at the Millenium (The Design and Evolution of Protege-2000*, SMI Report, SMI-1999-0801, 1999
- [47] Gangemi A., Pisanelli D.M., Steve G. *An overview of the ONIONS project: Applying ontologies to the integration of medical terminol-*

- ogies, Data and Knowledge Engineering 31, 1999
- [48] Gruber T.R. *A Translation Approach to Portable Ontology Specifications*, Knowledge Acquisition, 5, 1993
- [49] Guarino N. *Understanding Building and using Ontologies, A commentary to 'Using explicit Ontologies in KBS Development by van Heijst, Schreiber and Wielinya*, International Journal of Human-Computer Studies, 1997
- [50] Hemforth B. *Einführung in die Kognitionswissenschaft*, Universität Freiburg 2000, www.iig.uni-freiburg.de/cognition/teaching/veranstaltungen/ws00/kw2/kw2-03b.pdf
- [51] Hakala J., Husby O., Koch T. *Warwick framework and Dublin Core set provide a comprehensive infrastructure for network resource description*, <http://www.lub.lu.se/tk/warwick.html>
- [52] Hilt V. Schremmer C. Vogel J. *Lehrsznarien-übergreifende Erzeugung und Verwendung Multimedialer Teachware*, Technical Report 17-2000, Department for Mathematics and Computer Science, University of Mannheim, 2000
- [53] Hunter J. *A Comparison of Schemas for Video Metadata Representation*, Proceedings of the WWW8 Conference, ISBN 0-444-50264-5
- [54] I2 Visualize and Analyze Communicate, <http://www.i2.co.uk>,
- [55] IEEE P1484.1/D8, *Draft Standard for Learning Technology-Learning Technology Systems Architecture (LTSA)*, http://ltsc.ieee.org/wg1/IEEE_1484_01_D08_LTSA.pdf, 2001
- [56] IEEE P1484.12, *Draft Standard for Learning Object Metadata*, http://ltsc.ieee.org/wg12/LOM_WD6.pdf
- [57] ILOG Software Components, <http://www.ilog.com>, 2001,
- [58] IMS Global Learning Consortium, *Designing the future of global distributed learning*, <http://www.imsproject.org>, 2001
- [59] Inxight Software, <http://www.inxight.com>, 2001
- [60] International Organization for Standardization ISO, *ISO 11179 Specification and Standardization of Data Elements*, <http://www.schemas-forum.org/workshops/ws1/ws1-presentations/ISO-IEC11179/>, Mai 2000
- [61] International Organization for Standardization ISO, *ISO/IEC FCD 13250:1999 - Topic Maps*, <http://www.ornl.gov/sgml/sc34/document/0058.htm>, 1999
- [62] ISX Corporation, *LOOM Users Guide Version 1.4*, 1991
- [63] Iannella R., Waugh A., *Metadata : Enabling the Internet*, <http://www.dstc.edu.au/RDU/reports/CAUSE97/Metadata>
- [64] Jacobson I., Christerson M., Jonnson P., *Object Oriented Software Engineering: A use case driven approach*, Workingham, England Addison-Wsley

- [65] Jasper R. Uschold M. *A Framework for Understanding and Classifying Ontology Applications*, Proceedings of the Twelfth Workshop on Knowledge Acquisition, Modeling and Management 1999 Canada <http://sern.ucalgary.ca/KSI/KAW/KAW99/papers/Uschold2/final-ont-apn-fmk.pdf>
- [66] Kamps T., Kleinz J., *Constraint Based Spring Model Algorithm for Graph Layout*, Technical Report GMD IPSI
- [67] k-med Knowledge Based Multimedia Medicine Education, <http://www.k-med.org>, 2001
- [68] Krämer B., Steinmann U. *A market place for multimedia components*, Proceedings of the 15 th International Conference on DataEngineering ICDE'99 <http://www.cse.unsw.edu.au/icde99/index.html>
- [69] Krämer B. *Forming Federated Virtual University Through Course broker Middleware*, LearnTEC 2000, Schriftenreihe der Karlsruher Kongreß- und Ausstellungs-GmbH, Band I. S. 137-148, Februar 2000
- [70] Kashyap V., Shah K., Sehth A. *Issues and Research Directions, Kapitel Metadata for Building the Multimedia Patch Quilt MMDBMSs*, Springer-Verlag Berlin-Heidelberg, 1996
- [71] Kiriazidis I. *Automatische Generierung von Metadaten für multimediale Lernressourcen*, Diplomarbeit KOM, TU Darmstadt KOM-D-105, 2000
- [72] L-3 Lebenslanges Lernen: Weiterbildung als Grundbedürfnis, <http://www.l-3.de>,
- [73] Lamping J., Rao R., Pirolli P. *A Focus + Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies*, ACM Conference on Human Factors in Computing Systems (CHI'95), Denver Colorado, 1995
- [74] Lassila O., Swick R.R. *Resource Description Framework (RDF) Model and Syntax Specification*, W3C Recommendation, Februar 1999, <http://www.w3.org/TR/REC-rdf-syntax>
- [75] Lienhart R. Pfeiffer S., Effelsberg W. *Video abstracting*, Communications of the ACM, Volume 40. No.12 S.54-62, Juni 1997
- [76] Liepert M. *TTTTTTTT*, Dissertation Technische Universität Darmstadt, 2001
- [77] Liepert. M. et.al, *A distributed media server for the support of multimedia teaching*, Proceedings of the SPIE Photonics East, Boston September 1999
- [78] Marchiori M. *The limits of web metadata and beyond*, Computer Networks and ISDN Systems, Vol.30 Seite 1-9, 1999
- [79] Marti R. *Wissensbasierte Systeme*, Vorlesungsskript Wissensbasierte Systeme, ETH Zurich 2001, www.inf.ethz.ch/wbs/script/Einleitung.pdf

- [80] Martinez J.M. *Overview of the MPEG-7 Standard Version 5.0*, ISO/IEC JTC1/SC29/WG11 N4031, <http://www.cselt.stet.it/mpeg/standards/mpeg-7/mpeg-7.htm>, März 2001
- [81] Medibook Virtueller Fachbereich Medizin, <http://www.httc.de/virtueller-fachbereich-medizin/index.html>,
- [82] Melnik S., Decker S. *A Layered Approach to Information Modeling and Interoperability on the Web*, Proceedings of the ECDL'00 Workshop on the Semantic Web
- [83] Miao Y, *Design and Implementation of a Collaborative Virtual Problem-Based Learning Environment*, Dissertation Technische Universität Darmstadt, 2001
- [84] Miller J. *Rating Services and Rating Systems (and their Machine Readable Descriptions)*, Technical Report; <http://www.w3.org/TR/REC-PICS-services>, 1996
- [85] Miller J. *PICS Label Distribution Label Syntax and Communication Protocols*, Technical Report, <http://www.w3.org/TR/REC-PICS-labels-961031>, 1996
- [86] MILAN, *Modellierung einer Informations- und Lernlandschaft*, <http://www.pi.informatik.tu-darmstadt.de/milan/>, 2001
- [87] Mintert S., Menge S. *XML verpuppt - Aufbereitung mit Cocoon und Extensible Server Pages*, c't Magazin, Heft 10, Seite 222-227, 2000
- [88] MIT, *MITOpenCourseware*, <http://web.mit.edu/ocw>
- [89] Moeller C. *Erstellung und Verwendung einer Ontologie im Medi-Book*, Diplomarbeit KOM, TU Darmstadt, 2000
- [90] Moßgraber Jürgen, *Metdaten*, <http://aragon.iitb.fhg.de/moss/publications/Diplomarbeit/Metadaten.html>
- [91] Mühlhäuser M., Borchers J., Falkowski C. und Manske K. *The conference-/ Classroom of the future: An interdisciplinary Approach*, Glasson B., Vogel D. und Nunmaker J. (Hrsg.) Information Systems and Technology in the international Office of the future. Chapman and Hall, Boca Raton, USA, 1996
- [92] Muenzenberger D. *Interaktive Exploration großer Datenbestände mittels Graphenvisualisierung*, Diplomarbeit Nr. KOM-D-0076, KOM, Technische Universität Darmstadt, 1999
- [93] Multibook Projekt Homepage, <http://www.multibook.de>,
- [94] Mumenthey O. *Spezifikation und Implementierung eines Anfrageprotokolls für verteilte Metadatenrepositories*, Diplomarbeit Nr. KOM-D-126, KOM TU Darmstadt
- [95] Nack F., Lindsay A.T. *Everything you wanted to know about MPEG-7: Part 1*, IEEE Multimedia Systems vol 6. Nr 3, Juli-September S. 65-77, 1999
- [96] NetCoach, *Internet-Autorensystem und Lernplattform für elearning*, <http://www.net-coach.de>, 2001
- [97] Neuhold E., Aberer K. *Vorlesungsskript Multimediale Datenban-*

- ken, 1998
- [98] Natrifical, The Brain, <http://www.natrifical.com>, 2001
- [99] Neuss C. *Hawk - Experimental Tool for Information Visualization*, <http://www.objectweaver.de/software/hawk>
- [100] Neuss C., *Ein objektorientiertes Framework zur begriffsbasierten Informationsvisualisierung*, in W. Seesink *Bildung ans Netz Implementierung Neuer Technologien in Bildungseinrichtungen*, Schriftenreihe der Landesinitiative Hessen-media, S. 215-244, 2000
- [101] Nikolai R., Koschel A. Kramer R. *Metadatenaktualisierung am Beispiel des Umweltdatenkatalogs UDK*, <http://w3g.gkss.de/akudb/vilm/bt/nikolai.html>, 1997
- [102] O'Neill E.T., Lavoie B., McClain P. *Web Characteriazation Project An Analysis of Metadata Usage on the Web*, http://www.oclc.org/research/publications/arr/1998/oneill_etal/metadata.htm
- [103] OntoKnowledge, *Welcome to OIL*, <http://www.ontoknowledge.org/oil>, 2001
- [104] OR-World, <http://dsor.uni-paderborn.de>, 2001
- [105] Powell M.J.D. *An efficient method for finding the minimum of a function of several variables without calculating derivatives*, Computer Journal Vol 7. Seite 155ff., 1964
- [106] Rath H.H. *Knowledge Management: Mit Topic Maps intelligente Informationsnetze aufbauen*, iX, Heft 12 Seite 149-155
- [107] Resnick P. *Filtering Information on the Internet*, <http://www.sciam.com/0397issue/0397resnick.html>
- [108] Roecker C. *Untersuchung zur progammgesützten Generierung von Metadaten für Lerninhalte*, Diplomarbeit KOM TU Darmstadt, 2000
- [109] Saddik A.E., Seeberg C. Steinacker A. Reichenberger K. Fischer S. Steinmetz R. *A Component-based Construction Kit for Algorithmic Visualizations*, Proceedings of the Integrated Design & Process Technology IDPT99, ISBN 1090-9389
- [110] Saddik A.E. et.al, *Metadata for Smart Multimedia Learning Objects*, PTagungsband der 4.ten Australian Computing Education Conference, ACM Press New York, USA, Dezember 2000
- [111] Saddik A.E. *Interactive Multimedia Learning: Shared Reusable Visualization-based Modules*, Dissertation Technische Universität Darmstadt, 2001
- [112] Schremmer C. Kuhmünch C. Wons H. *3Simulations in Interactive Distance Learning*, rd International Conference on New Learning Technologies (NLT) Fribourg Switzerland, 2001, <http://www.eif.ch/telecom/netties/frames.html>
- [113] Schulmeister R. *Virtuelle Universität Virtuelles Lernen*, Oldenbourg Verlag München ISBN 3-486-25742-0, 2001
- [114] Schulmeister R., *Grundlagen hypermedialer Lernsysteme*, Olden-

- bourg Verlag München, 1997
- [115] Scragg G. *Semantic Nets as Memory Models*, Computational Semantics, Fundamental Studies in Computer Science 4 north holland 1976
- [116] Seeberg C. *Modularität und Kohärenz: Probleme bei der Entwicklung elektronischer Lehr- und Lernsysteme*, Werber Seesink (Ed): Bildung ans Netz: Schriftenreihe der Landesinitiative Hessen-medie Band 23, Wiesbaden Dezember 2000
- [117] Seeberg C., Steinacker A., Steinmetz R. *Coherence in Modularly Composed Adaptive Learning Documents*, Proceedings Proceedings of the AH2000, pages 375-379. Springer Verlag, August 2000. ISBN 3-540-67910-3.
- [118] Seeberg C., Steinacker A., Steinmetz R. *Wie unabhängige Lerneinheiten ein individuelles Ganzes ergeben: Datenmodelle für adaptive Lernsysteme*, in Deiters W. Lienemann C. (Hrsg.) Informationslogistik - Informationsversorgung Just in Time, Symposium Publishing Verlag, erscheint 2001
- [119] C. Seeberg, *Modulare Wissensbasen zur Erzeugung adaptiver und kohärenter Lerndokumente*, Dissertation Technische Universität Darmstadt, 2001
- [120] Seeberg S., Steinacker A. Reichenberger K. et.al. *From the User's Needs to Adaptive Documents*, Proceedings of the Integrated Design & Process Technology IDPT'99 Kusadasi Turkey (published in the Proceedings of IDPDT'2000 Texas USA)
- [121] Seeberg C. Steinacker A. Reichenberger K. et.al. *Individual Table of Contents in Web-based Learning Systems*, Proceedings of the 10th ACM Conference on Hypertext and Hypermedia, pages 167-168, February 1999
- [122] Sowa J.F. *Ontology - Definition and Scope*, <http://www.bestweb.net/sowa/ontology/index.htm>, Mai 2000
- [123] Sowa J.F. *Knowledge Representation*, Brooks/Cole, ISBN 0534-94965-7, 2000
- [124] Staab S., Maedche A., Handschuh S. *Creating Metadata for the Semantic Web - An Annotation Environment and the Human Factor*, http://http://www.aifb.uni-karlsruhe.de/WBS/ama/publications/paper_semman.pdf
- [125] Steinmetz R., Seeberg C., Steinacker A. *Coherence in the Learning System k-med*, Proceedings of the GLDV-Spring Meeting, <http://www.uni-giessen.de/fb09/ascl/gldv2001>
- [126] Steinacker A., Seeberg C., Reichenberger K., Fischer S. Steinmetz R. *Dynamically Generated Table of Contents as Guided Tours in Adaptive Hypermedia Systems*, Proceedings of the EdMedia & EdTelecom S.167-175, ISBN 1-58113-064-3, Juni 1999
- [127] Steinacker A., Ghavam A., Steinmetz R. *Metadata Standards for Web-Based Resources*, IEEE Multimedia Systems, no1. Januaray-

- March 2001, pp.70-76
- [128] Steinacker A., Seeberg C., Fischer S. Steinmetz R. *Multibook: Metadata for Web-Based Learning Systems*, 2nd International Conference on New Learning Technologies, Bern Switzerland, Seite 16-24, August 1999
- [129] Steinmann U. *Hyperwave als Entwicklungsumgebung für einen Multimedia-Katalog*, 11. te Fachtagung Kommunikation in Verteilten Systemen (KiVS'99), R. Steinmetz (Hrsg.) Springer Verlag Berlin ISBN-3-540-65597-2, Juni 1999
- [130] R. Steinmetz, *Multimedia-Technologie*, Springer-Verlag Heidelberg, 2000
- [131] Steinmetz R., Seeberg C., Steinacker A., *Coherence in the Learning System k-Med*, Proceedings der GLDV-Frühjahrstagung, pages 17-27, March 2001
- [132] Strasser A. *Generierung domänenspezifischer Wissensrepräsentationssysteme und Transformation von Wissensbasen mit einer Anwendung in der Rechtsinformatik*, Band 8 aus Dissertationen zur Künstlichen Intelligenz, infix 1992
- [133] Teachware on Demand, <http://www.teachware-on-demand.de>,
- [134] Tietze D. *A Framework for Developing Component-based Co-operative Applications*, Dissertation Technische Universität Darmstadt, 2001
- [135] Tolksdorf R. *Die Sprache des Web: HTML 4.3*. Auflage d.punkt Verlag Heidelberg, 1997
- [136] Tschira Klaus, *Workshop Standardisierung von Lehr/Lernsoftware*, Juni 1999, <http://www.kts.villa-bosch.de/deutsch/aktivitas/lernsoft.html>
- [137] Tsichritzis D. *Reengineering the University*, Communications of the ACM, Volume 42. No.6 S.93-100, Juni 1999
- [138] T-Systems, *Global Learning*, <http://www.global-learning.de>
- [139] T Systems, *Corporate Learning*, <http://www.corporate-learning.de>
- [140] ULI, *Universitärer Lehrverbund Informatik*, <http://www.gmd.de/NMB/Bereich%20Hochschulen/ldf%20Projekte/08NM074.htm>
- [141] VIROR, *Virtuelle Hochschule Oberreihn*, <http://www.viror.de>
- [142] W3C, *XML Schema Part 0: Primer W3C Proposed Recommendation*, <http://www.w3.org/TR/xmlschema-0/>, 30.03.2001
- [143] Wade V. et. al. *GESTALT - Getting Educational Systems Talking Across Leading-Edge Technologies*, WorkPackage 5 Object Specification, <http://www.fdgroupp.com/gestalt/news.html>
- [144] Wason T.D., Wiley D. *Structured Metadata Spaces*, <Metadata and Organizing Educational Resources on the Internet, J.Greenberg (Hrsg.) 2001, S.263-278, Haworth NY
- [145] Weber H. *Informationslogistik und Wissensmanagement*, <http://>

- www.informationslogistik.org/publikationen/index.html
- [146] Weibel S. et al, *Dublin Cor Metadata for Ressource Discovery*, RFC 2413 <http://www.ietf.org/rfc/rfc2413.txt>, 1998
- [147] Wessner M. *Software für e-Learning: Kooperative Umgebungen und Werkzeuge*, in Schulmeister R. (Hrsg.) *Virtuelle Universität Virtuelles Lernen*, Oldenbourg Verlag 2001, ISBN 3-486-25742-0, 2001
- [148] Wessner M. Pfister H. *Kooperatives Lehren und Lernen*, In: Schwabe/Unland/Streitz (Eds.) *CSCW Kompendium - Lehr- und Handbuch für das Kooperative Arbeiten*, Springer Verlag 2001
- [149] Wille R. *Begriffliche Wissensverarbeitung: Theorie und Praxis*, Informatik Spektrum 6, 2000 S. 357-369, Springer Verlag Heidelberg, 2000
- [150] Gene Witmer, *Dictionary of philosophy of mind - ontology*, <http://www.artsci.wustl.edu/philos/MindDict/ontology.html>, 2001