



Wiederverwendung von digitalen Lernobjekten in einem auf Aggregation basierenden Autorenprozess

Vom Fachbereich Informatik
der Technischen Universität Darmstadt
genehmigte Dissertation
zur Erlangung des akademischen Grades eines
Doktor-Ingenieurs (Dr.-Ing.)

von

Dipl.-Ing. Stefan Hörmann
geboren in Hoya an der Weser

Erstreferent: Prof. Dr.-Ing. Ralf Steinmetz
Korreferent: Prof. Dr.-Ing. Jörg M. Haake

Tag der Einreichung: 30.09.2005
Tag der Disputation: 02.12.2005

Darmstadt 2006
Hochschulkenziffer D17

Kurzfassung

Die rechnergestützte Lehre gewinnt in der universitären Ausbildung sowie in der außeruniversitären Aus- und Weiterbildung zunehmend an Bedeutung. Im Vergleich zu Büchern bieten rechnergestützte Lehrangebote die Möglichkeit, über die Integration multimedialer Elemente die Interaktivität zwischen Lerner und Lernmedium zu erhöhen. Jedoch ist die Erstellung von hochwertigen Selbstlernmaterialien für die rechnergestützte Lehre, welche die Potenziale des Einsatzes multimedialer Elemente nutzen, mit hohen Kosten verbunden.

Bestehende Ansätze versprechen eine Verbesserung der Kosten-Nutzen-Relation der rechnergestützten Lehre durch die Wiederverwendung bereits existierender Lernmaterialien. Hierbei wird insbesondere die Wiederverwendung kompletter, bestehender und zuvor verfügbar gemachter Lernmaterialien vorgeschlagen. Die mangelnde Anpassungsmöglichkeit von Lernmaterialien bei der Wiederverwendung, sowie der zusätzliche Aufwand, der für die Indizierung und Bereitstellung von Lernmaterialien entsteht, lässt allerdings bisher deren Wiederverwendung oft als nicht rentabel erscheinen, so dass eine Wiederverwendung in diesem Sinne nur selten stattfindet.

Ziel der Arbeit ist die Konzeption und Entwicklung eines Gesamtsystems für die Wiederverwendung von Lernmaterialien, das innovative Ansätze zur Lösung allgemeiner Defizite der Wiederverwendung von Lernmaterialien beinhaltet. Als Ansatz wird hierfür die Wiederverwendung von Lernmaterialien im Autorenprozess vorgesehen. Im Vergleich zu den meisten bestehenden Arbeiten bietet dieser Ansatz die Möglichkeit der Anpassung von Lernmaterialien vor deren Verwendung. Außerdem kann in einem Autorenprozess die partielle Wiederverwendung von Lernmaterialien unterstützt werden. Innerhalb dieses Autorenprozesses können neue Lernmaterialien basierend auf wiederverwendeten und neu erstellten Komponenten zusammengestellt werden. Ein solcher Autorenprozess wird als ein auf Aggregation basierender Autorenprozess bezeichnet.

Zur Verbesserung der Wiederverwendung von Lernmaterialien wird zusätzlich die Integration von Lernobjektarchiven in den Autorenprozess empfohlen. Gegenstand der Untersuchung ist einerseits die Wiederverwendung von Lernmaterialien im Autorenprozess und andererseits das Verfügbarmachen von Lernmaterialien, die im Autorenprozess neu erstellt werden. Das Ziel hierbei entwickelter Lösungsansätze ist es, die Wiederverwendung von Lernmaterialien im Autorenprozess zu unterstüt-

zen und das Verfügbarmachen von Lernmaterialien deutlich zu vereinfachen.

Kernpunkt der vorliegenden Arbeit ist die Entwicklung eines abstrakten Prozessmodells für einen auf Aggregation von Lernmaterialien basierenden Autorenprozess. Dieses Prozessmodell dient als Grundlage für die Entwicklung spezieller EditorKomponenten für Lernmaterialien verschiedener Ausprägungen. Es beinhaltet sowohl die Wiederverwendung bereits bestehender als auch die Erstellung neuer Lernmaterialien. Darüber hinaus macht das abstrakte Modell des Autorenprozesses keine Einschränkungen bezüglich der Quellen aus denen Lernmaterialien für deren Wiederverwendung stammen können. Diese konzeptionelle Integrationsleistung im Prozessmodell ist neuartig.

Grundlage für den entwickelten, neuartigen Autorenprozess ist die Verfügbarkeit von bestehenden Lernmaterialien. Insbesondere für die partielle Wiederverwendung von Lernmaterialien ist deren Verfügbarkeit in modularisierter Form erforderlich. Aus diesem Grund ist im Rahmen dieser Arbeit ein innovatives, generisches Dokumentmodell entwickelt worden, das eine modularisierte Speicherung von Lernmaterialien erlaubt. Es bietet somit die Basis für die partielle Wiederverwendung von Lernmaterialien im Autorenprozess.

Metadaten spielen eine Schlüsselrolle bei der Wiederverwendung von Lernmaterialien, da sie in der Regel für die Indizierung und das Finden wiederverwendbarer Lernmaterialien herangezogen werden. Deshalb wurde in der vorliegenden Arbeit ein neues Verfahren für die erleichterte, semiautomatische Generierung von Metadaten entwickelt. Dieses Verfahren stellt eine Rekombination verschiedener Methoden der Metadatengenerierung dar, das hinsichtlich der Zuverlässigkeit der generierten Metadaten und der Reduzierung des Arbeitsaufwandes bei der lernstandardkonformen Erstellung von Metadaten im Autorenprozess optimiert wurde.

Im Rahmen dieser Arbeit wurde eine Autorenumgebung entwickelt, die für die Erstellung von Lernmaterialien für die rechnergestützte Lehre eingesetzt werden kann. Neben der Autorenumgebung stellt das ResourceCenter Funktionen eines Lernobjektarchivs bereit, mit dessen Hilfe Lernmaterialien für die Wiederverwendung verfügbar gemacht werden können. Das ResourceCenter wird produktiv im Rahmen eines interdisziplinären Projekts für die Erstellung von hochqualitativen Kursen für die rechnergestützte Lehre im Studienbereich der Humanmedizin eingesetzt. Auf Basis dieser Nutzung konnten die im ResourceCenter umgesetzten Konzepte der vorliegenden Arbeit erfolgreich evaluiert werden.

Reusing Digital Learning Objects Based on Authoring by Aggregation

Web-based training is receiving increasing interest in the areas of university education and advanced vocational training. In comparison with books, web-based training offers the possibility of increasing the interactivity between learner and learning object by integrating multi-media elements. However, the production of high-quality web-based training courses, which use the potentials of multi-media elements is comparatively expensive.

Common approaches promise an improvement of the cost-value ratio of web-based training by reusing already existing learning objects. Particularly the reuse of available learning objects as a whole is suggested. This reuse has not seemed to be profitable so far, because of the lacking adaptability of reused learning objects as well as the additional effort for indexing and supplying learning objects. Thus the reuse of learning objects is rarely practiced.

Aim of this thesis is the conception and development of a system for reusing learning objects. It contains innovative approaches for solving general shortcomings in the area of reusing learning objects. As an approach, reusing learning objects in the authoring process is proposed. In comparison to most existing work this approach offers the possibility of customizing learning objects before reusing them. Furthermore, learning objects can partially be reused in the authoring process. In the course of this authoring process new learning objects can be arranged on the basis of reused parts of already existing learning objects. This authoring process is called "authoring by aggregation".

In order to improve reusing learning objects, it is additionally recommended to integrate a repository for learning objects in the authoring process. One aspect of this thesis is being able to reuse and adapt existing learning objects - a second aspect is the necessity to ensure that newly created learning objects are made available for reuse.

The development of an abstract model of an authoring process is of central interest in this thesis. This authoring process should allow the creation of learning objects based on authoring by aggregation. It serves as a basis for the development

of more specific editor components for learning objects with different characteristics. The developed model contains the reuse of existing as well as the creation of new learning objects. Furthermore, the abstract model of the authoring process does not make any restrictions regarding the source from which reused learning objects originate from. The conceptual integration of this model is new.

Basis for the authoring process is the availability of already existing learning objects. The availability of modularized learning objects is particularly required for partially reusing learning objects. For this reason an innovative, generic document model, which permits a modularized storage of learning objects has been developed in this thesis.

Metadata play a major role in reusing learning objects, since they are often utilized for indexing and retrieving. Therefore, in this thesis a new method for semi-automatic metadata creation was developed. The proposed approach combines different, already existing methods for metadata creation. It was optimized to ensure the reliability of created metadata and to reduce the effort of metadata creation in the authoring process.

In the context of this thesis an authoring environment was developed which additionally provides basic functions of a learning object repository. This authoring environment is used productively in an interdisciplinary e-learning project to create high-quality web-based training for graduate courses in human medicine at Hessian universities. Based on this the concepts which are both developed in this thesis and realized in the authoring environment could be evaluated successfully.

Inhaltsverzeichnis

Vorwort	1
1 Einführung	3
1.1 Motivation	3
1.2 Eigener Ansatz und Zielsetzung	4
1.3 Abgrenzung der Arbeit	5
1.4 Aufbau der Arbeit	6
2 Wiederverwendung von Lernobjekten	9
2.1 Wiederverwendung	9
2.2 Lernobjekte	12
2.3 Modularisierung und Aggregation	13
2.4 Aggregation als Basis für Autorenprozesse	15
2.5 Ziel und Problemstellung	17
2.6 Zusammenfassung	19
3 Verwandte und grundlegende Arbeiten	21
3.1 Modularisierung von Lernobjekten	21
3.2 Datenmodelle für Lernobjekte	26
3.3 Autorenwerkzeuge	29
3.4 Metadaten und Metadatengenerierung	31
3.5 Lernobjektarchive	34
3.6 Zusammenfassung	36
4 Grundlagen eines auf Aggregation basierenden Autorenprozesses	39
4.1 Anforderungsanalyse	39
4.1.1 Voraussetzungen für die Wiederverwendung	40
4.1.2 Anforderungen an den Autorenprozess	40
4.1.3 Wiederverwendung von Lernobjekten im Autorenprozess	45
4.2 Dokumentmodell für Kurse	46
4.2.1 Abgrenzung zu verwandten Dokumentmodellen	46
4.2.2 Knoten im Dokumentmodell	47
4.2.3 Aggregation von Lernobjekten im Dokumentmodell	48

4.2.4	Klassifizierung von Dokumentknoten	51
4.2.5	Beschreibung aggregierter Lernobjekte	54
4.3	Analyse des Autorenprozesses	55
4.3.1	Abstraktes Prozessmodell	56
4.3.2	Erweitertes Prozessmodell	58
4.4	Zusammenfassung	62
5	Metadatengenerierung	65
5.1	Methoden zur Erstellung von Metadaten	65
5.1.1	Manuelle Erfassung von Metadaten	65
5.1.2	Metadatenschablonen	66
5.1.3	Nutzung von Abhängigkeiten im Datenschema	68
5.1.4	Nutzung von Kontextinformationen	69
5.1.5	Nutzung bereits bestehender Metadaten	71
5.1.6	Gewinnung von Metadaten durch Inhaltsanalyse	72
5.1.7	Systembedingte Metadaten	74
5.1.8	Ableiten von Metadatensätzen	75
5.1.9	Bewertung	76
5.2	Generierung von Metadaten für Lernobjekte	78
5.2.1	Metadatenschema	78
5.2.2	Obligatorische Datenfelder	79
5.2.3	Semiautomatische Metadatengenerierung	81
5.3	Integration der Metadatengenerierung in den Autorenprozess	86
5.4	Zusammenfassung	88
6	Versionierung modularisierter Lernobjekte	91
6.1	Anforderungen	91
6.2	Versionierung im Allgemeinen	94
6.3	Annahmen für das Versionierungskonzept	95
6.4	Versionierung einzelner Lernobjekte	96
6.4.1	Zustände von Lernobjekten	96
6.4.2	Versionierung von Lernobjekten	98
6.4.3	Bezeichnete Versionen	101
6.4.4	Sichtbarkeit von Versionen im Lernobjektarchiv	102
6.4.5	Modifizierte Kopien von Lernobjekten	103
6.5	Versionierung von Dokumentknoten	105
6.5.1	Propagierung von Zuständen	105
6.5.2	Propagierung neuer Versionen	108
6.5.3	Propagierung benannter Versionen	110
6.6	Zusammenfassung	110

7	Implementierung	111
7.1	Systemarchitektur	111
7.1.1	Anwendungsfälle	111
7.1.2	Einfaches Einzelplatzsystem	113
7.1.3	Erweitertes Einzelplatzsystem	114
7.1.4	Mehrplatzsysteme	115
7.1.5	Bewertung	120
7.2	Lernobjektarchiv	120
7.2.1	Metadatenschema	120
7.2.2	Funktionen des Lernobjektarchivs	124
7.2.2.1	Ablegen von Lernobjekten	124
7.2.2.2	Abrufen von Lernobjekten	129
7.2.2.3	Verwalten von Lernobjekten	132
7.2.2.4	Versionierung von Lernobjekten	133
7.3	Autorenumgebung	135
7.3.1	Datenmodell	135
7.3.2	Editorkomponenten der Autorenumgebung	139
7.3.2.1	Kurseditor	139
7.3.2.2	Lektioneditor	142
7.3.2.3	Blockeditor	145
7.3.3	Einfügen von Lernobjekten in der Autorenumgebung	147
7.3.3.1	Lernobjekt aus der Autorenumgebung einfügen	147
7.3.3.2	Lernobjekt aus dem Lernobjektarchiv einfügen	148
7.3.3.3	Lernobjekt von einem lokalen Laufwerk einfügen	149
7.3.4	Zusammenspiel der Editorkomponenten	150
7.4	Zusammenfassung	155
8	Evaluation	157
8.1	Kontext der Evaluation	157
8.2	Erstellung von Metadaten	158
8.3	Verwendbarkeit der Autorenumgebung	161
8.4	Wiederverwendung im ResourceCenter	163
8.5	Zusammenfassung	165
9	Zusammenfassung und Ausblick	167
9.1	Zusammenfassung	167
9.2	Ausblick	169
	Literaturverzeichnis	171
	Online Referenzen	185

Abkürzungsverzeichnis	187
Anhang	189
A XML-Schema des Datenmodells	191
B Publikationen des Verfassers	195
C Lebenslauf des Verfassers	199

Abbildungsverzeichnis

2.1	Wiederverwendung bei der Erstellung neuer Lernmaterialien	11
2.2	Aggregation eines neuen Lernobjekts (LO)	15
2.3	Autorensystem zur Aggregation von Lernobjekten	18
3.1	Wiederverwendung im Aggregationsniveauspektrum	23
3.2	Hierarchie für modulare Lernobjekte	24
3.3	Funktionen eines Lernobjektarchivs	35
4.1	Aggregation auf Basis von Platzhaltern in der Autorenumgebung	42
4.2	Zusammensetzung eines Dokumentknotens	48
4.3	Visualisierung eines Knotens im Dokumentmodell	48
4.4	Rekursive Aggregation von Objekten im Dokumentmodell	49
4.5	Visualisierung eines Knotens im Dokumentmodell	49
4.6	Verkettung von Knoten des Dokumentmodells	50
4.7	Lernobjektklassen und ihre Aggregation im Dokumentmodell	53
4.8	Abstraktes Prozessmodell	57
4.9	Erweiterte Modellierung des Autorenprozesses	59
5.1	Einfache, formularbasierte Metadateneditoren	66
5.2	Erstellen von Metadatensätzen mit Metadatenschablonen	67
5.3	Kontextinformationen bei der Erstellung von Metadaten nutzen	70
5.4	Bestehende Metadaten bei der Erstellung von Metadaten nutzen	72
5.5	Bestimmung von Metadaten durch Inhaltsanalyse	73
5.6	Metadatenerstellung auf Basis systembedingter Metadaten	74
5.7	Semiautomatische Generierung von Metadaten	85
5.8	Metadatengenerierung im Autorenprozess	88
6.1	Zustände eines Lernobjekts	97
6.2	Zustandsmarker zum Speichern des Zustandes von Lernobjekten	98
6.3	Versionen von Lernobjekten	99
6.4	Verkettung verschiedener Versionen eines Lernobjekts	99
6.5	Bezeichnete Version eines Lernobjekts	102
6.6	Sichtbarkeit von Versionen im Lernobjektarchiv	103

6.7	Ableiten von Lernobjekten	104
6.8	Beispiel eines abgeleiteten Lernobjekts	105
6.9	Propagierung von Zustandsänderungen im Kurs	107
6.10	Änderung von Lernobjekten im Kurs	109
6.11	Propagierung neuer Versionen im Kurs	109
7.1	Anwendungsfälle für die Autorenumgebung	112
7.2	Systemkomponenten eines einfachen Einzelplatzsystems	113
7.3	Systemkomponenten des erweiterten Einzelplatzsystems	114
7.4	Zentrale Autorenumgebung mit zentralem Lernobjektarchiv	116
7.5	Autorenumgebung mit zentralem Lernobjektarchiv	118
7.6	Autorenumgebung mit P2P-basierter Verteilung der Lernobjekte	119
7.7	Ablegen von Lernobjekten im Lernobjektarchiv	125
7.8	Aufladeformular für das Ablegen neuer Lernobjekte	125
7.9	Beschreibung präsentationsspezifischer Metadaten	126
7.10	Vereinfachte Eingabemaske des Metadaten-Wizards	127
7.11	Metadateneditor zur Bearbeitung der generierten Metadaten	128
7.12	Zusammenfassung des neu abgelegten Lernobjekts	129
7.13	Abrufen verfügbarer Lernobjekte	130
7.14	Suchformular zur Suche verfügbarer Lernobjekte	131
7.15	Ergebnisliste gefundener, verfügbarer Lernobjekte	131
7.16	Lernobjekte im ResourceCenter verwalten	132
7.17	Player für WBTs	133
7.18	Basisfunktionen des ResourceCenters als Lernobjektarchiv	134
7.19	Aufbau von Kursen im ResourceCenter	136
7.20	XML-Schema für Kurse im ResourceCenter	137
7.21	Aufbau von Lektionen im ResourceCenter	137
7.22	XML-Schema für Lektionen im ResourceCenter	138
7.23	Autorenprozess des Kurseditors	140
7.24	Kurseditor des ResourceCenters	141
7.25	Autorenprozess des Lektioneneditors	143
7.26	Lektioneneditor des ResourceCenters	144
7.27	Autorenprozess des Blockeditors	145
7.28	In den Lektioneneditor eingebetteter Blockeditor	146
7.29	Lernobjekt aus der Autorenumgebung einfügen	147
7.30	Lernobjekt aus dem Lernobjektarchiv einfügen	148
7.31	Lernobjekt von einem lokalen Laufwerk einfügen	150
7.32	Erstellen eines neuen Kurses	151
7.33	Erstellen einer neuen Lektion	153
7.34	Speichern der Lektion und des Kurses	154
8.1	Wiederverwendung von Lernobjekten	165

Tabellenverzeichnis

5.1	Bewertung Metadatenerstellung	76
5.2	Obligatorische Datenfelder	79
5.3	Generierung obligatorischer Datenfelder	82
5.4	Priorisierung der Methoden für die Metadatenerstellung	84
7.1	Administrative Erweiterungen der Lernobjekt Metadaten	121
7.2	Präsentationsspezifische Erweiterungen der Lernobjekt Metadaten .	123
8.1	Mit dem ResourceCenter erstellte Metadatenätze	160
8.2	Erstellte und gespeicherte Lernobjekte nach Typ	162
8.3	Wiederverwendete Lernobjekte	164

Vorwort

Mit der vorliegenden Dissertationsschrift beschreibe ich grundlegende Ansätze zur effizienten Wiederverwendung von digitalen Lernmaterialien mit der Absicht neue Kurse für die rechnergestützte Lehre in Autorenumgebungen zu erstellen. Damit leiste ich einen Beitrag zu einem Thema, das in der Literatur zwar häufig angesprochen wird und wurde, dem jedoch trotz seiner Bedeutung bislang allgemein relativ wenig Aufmerksamkeit in der Forschung entgegengebracht wurde.

Die Erstellung meiner Arbeit wurde im Wesentlichen durch mein Umfeld ermöglicht und gefördert. Ich bin meinen Eltern und Großeltern für die geistige und materielle Unterstützung, die mir die Schulbildung und das Studium ermöglicht hat, zutiefst dankbar. Ebenso möchte ich mich bei Herrn Prof. Dr.-Ing. Ralf Steinmetz für die Betreuung meiner Doktorarbeit herzlich bedanken. Mein Dank gilt auch dem gesamten Lehrstuhl „Multimedia Kommunikation“, meiner Forschungsgruppe „Multimedia Semantics“ und später „Media Creation, Management & Usage in E-Learning“ und insbesondere meinem geduldigen Zimmerkollegen Dr.-Ing. Andreas Faatz.

Meinen besonderen Dank möchte ich all denen aussprechen, die die vorliegende Arbeit auf Fehler untersucht und mir Hinweise zur Verbesserung der Arbeit gemacht haben: Dr.-Ing. Christoph Rensing, Prof. Dr.-Ing. Jörg M. Haake, Dr.-Ing. Manuel Görtz, Dr.-Ing. Cornelia Seeberg, Dipl. Psych. Susanne Köbler, Dipl. Päd. Julia Sonnberger, Assessor Jan Hansen, Dipl.-Ing. Lasse Lehman. Für die Bearbeitung verschiedener Problemstellungen möchte ich mich an dieser Stelle auch bei meinen Studenten, insbesondere bei Sonja Bergsträcker, Markus Müller und Johannes Pötsch, bedanken.

Ganz besonders möchte ich mich bei meiner Frau Janine für die emotionale und geistige Unterstützung während der Arbeit bedanken. Bedanken möchte ich mich auch bei meinen beiden Söhnen Felix und Wenzel, die mir immer eine willkommene Ablenkung von meiner Arbeit waren. Vielen Dank für die Geduld und das Verständnis, dass ich für die Fertigstellung meiner Arbeit viele Stunden abwesend war.

Stefan Hörmann, September 2005

1 Einführung

1.1 Motivation

Die rechnergestützte Lehre gewinnt in der universitären Ausbildung sowie in der außeruniversitären Aus- und Weiterbildung zunehmend an Bedeutung. Die Vorteile der rechnergestützten Lehre liegen vor allem in der Selbstbestimmung von Lernort, Lernzeit und Lerntempo, sowie der möglichen Individualisierung der Lernmaterialien an die Voraussetzungen der Lerner [Eul01]. Im Vergleich zu Büchern bieten rechnergestützte Lehrangebote die Möglichkeit, über die Integration multimedialer Elemente die Interaktivität zwischen Lerner und Lernmedium zu erhöhen. Zum anderen sind sie in der Lage aufgrund der Multimodalität der Medien verschiedene Sinnesorgane anzusprechen und über die Verwendung multicodaler Darstellungen verschiedene und anschauliche Präsentationen von Lerninhalten zu integrieren [NHHM⁺04].

Die rechnergestützte Lehre galt außerdem lange Zeit als kostensparend. Diese Annahme hat sich mit Ausnahme von Lehrangeboten für eine große Lernerzahl nur in wenigen Fällen bestätigt. Die Erstellung von hochwertigen Selbstlernmaterialien für die rechnergestützte Lehre, welche die Potenziale des Einsatzes multimedialer Elemente nutzen, ist mit hohem Aufwand verbunden [Ker01].

Durch die Wiederverwendung bestehender Lernmaterialien soll die Kosten-Nutzen-Relation für die Erstellung rechnergestützter Lernmaterialien aufgrund der mehrfachen Nutzung verbessert werden. Dabei steht bisher insbesondere die Wiederverwendung bestehender Lernmaterialien seitens der Lehrenden im Fokus der Forschung. Bei dieser Art der Wiederverwendung stellen Lehrende den Lernern eine oder mehrere komplette Lerneinheiten zur Verfügung, die nicht für die Wiederverwendung an den neuen Lernkontext angepasst werden.

Um eine Wiederverwendung von Lernmaterialien zu ermöglichen, ist es erforderlich diese für die Suche zu indizieren und Anderen an einem gemeinsam zugreifbaren Ort zur Verfügung zu stellen [DFC⁺01, Pol03]. Das ist insbesondere dann notwendig, wenn das Angebot verfügbarer Lernmaterialien nicht mehr überschaubar ist und sich gleichzeitig eine jeweils große Menge von Anbietern und Nachfragern gegenübersteht. Für diese Problemstellung wurden in den vergangenen Jahren so genannte Lernobjektarchive entwickelt und Metadatenbeschreibungen zur Indizierung der Lernobjekte standardisiert [LOM02].

Die mangelnde Anpassbarkeit von Lernmaterialien bei der Wiederverwendung

sowie der zusätzliche Aufwand, der für die Indizierung und Bereitstellung von Lernmaterialien entsteht, lässt allerdings bisher deren Wiederverwendung oft als nicht rentabel erscheinen. Eine Wiederverwendung in diesem Sinne findet nur selten statt.

1.2 Eigener Ansatz und Zielsetzung

Der Ausgangspunkt der vorliegenden Arbeit ist die unzureichende Nutzung des Potenzials zur Reduzierung der Erstellungskosten von Lernmaterialien durch die Wiederverwendung.

Der dieser Arbeit zugrunde liegende Ansatz sieht statt der Wiederverwendung nicht angepasster Lernobjekte in der Lehre die Wiederverwendung von Lernmaterialien während der Erstellung neuer Lernmaterialien vor. Dazu ist es notwendig, dass eine partielle Wiederverwendung von Lernmaterialien, wie beispielsweise einer Einleitung eines Themas oder eines Anwendungsbeispiels, und deren Anpassung an den neuen Lernkontext ermöglicht wird. Die neue Lerneinheit wird dann aus bestehenden Fragmenten, die im Rahmen des Autorenprozesses anpassbar sind, und neu entwickelten Fragmenten zusammengesetzt. Dieser Prozess wird in Übereinstimmung mit Duval und Hodgins [DH03] als ein auf Aggregation basierender Autorenprozess (Authoring by Aggregation) bezeichnet.

Weiterhin besteht der Ansatz darin, die Auszeichnung und Bereitstellung von Lernmaterialien sowohl zu vereinfachen als auch so in den Autorenprozess einzubinden, so dass den Autoren diese Aufgaben durch weitestgehende Automatisierung abgenommen werden können. Dank dieser Erleichterung soll die Anzahl der zur Wiederverwendung bereitstehenden Lernmaterialien deutlich erhöht werden.

Das Ziel der vorliegenden Dissertationsschrift ist es nun, einen Autorenprozess zu entwickeln und zu unterstützen, mit dessen Hilfe die partielle Wiederverwendung bereits bestehender digitaler Lernobjekte zur Erstellung neuer Lernobjekte ermöglicht wird. In diesem Autorenprozess sollen bereits bestehende Lernobjekte entweder ohne oder mit Anpassung wiederverwendet und zusammen mit neu erstellten Fragmenten zu neuen Lernobjekten kombiniert werden können. Dabei wird in dem Autorenprozess auch die Wiederverwendung von Lernobjekten anderer Autoren unterstützt. Hierfür wird die Modellannahme getroffen, dass sämtliche verarbeiteten Inhalte einer auf digitale Inhalte bezogenen Lizenz nach dem Muster der GNU General Public License (GPL) der Free Software Foundation [11]¹ unterliegen. Damit klammern wir bei der Argumentation juristische Implikationen der Wiederverwendung aus.

Dabei soll in dieser Arbeit nicht allein eine Analyse und Optimierung des Autorenprozesses erfolgen, sondern es sollen für die Wiederverwendung von Lernob-

¹In der vorliegenden Dissertationsschrift werden zitierte Web-Seiten im Gegensatz zu allen übrigen Zitaten mit einer fortlaufenden Numerierung referenziert. Die Auflistung der referenzierten Web-Seiten erfolgt getrennt im Abschnitt „Online Referenzen“.

jekten verschiedene Konzepte, die den ermittelten Autorenprozess unterstützten, entwickelt und letztendlich durch Implementierung überprüft werden. Das betrifft insbesondere die Betrachtung der Speicherung der Lernobjekte in modularisierter Form, die eine Grundlage für die Wiederverwendung von Bestandteilen der Lernobjekte darstellt. Zusätzlich dazu soll eine weitestgehend automatisierte Erfassung von Metadaten schon während der Erstellung der Lernobjekte untersucht werden. Im Mittelpunkt stehen dabei gleichermaßen Lernobjekte, die mit Hilfe der Autorenumgebung erstellt werden, als auch Lernobjekte, die mit externen Werkzeugen erstellt werden und in die Autorenumgebung eingebracht werden. Dabei soll untersucht werden, in welchem Umfang die Erstellung von Metadatensätzen, die für die Indizierung der Lernobjekte verwendet werden, automatisiert werden kann.

1.3 Abgrenzung der Arbeit

In der vorliegenden Dissertationsschrift liegt der Fokus auf der Wiederverwendung von Lernobjekten und deren Fragmenten in einem auf Aggregation basierenden Autorenprozess, so dass die Arbeit weder einen Beitrag zu einem Autorenwerkzeug für die verteilte, synchrone beziehungsweise kooperative Bearbeitung von Objekten durch mehrere Autoren [Sad01] im allgemeinen liefert, noch werden rechnergestützte, synchrone und kollaborative Lernszenarien, wie sie in [Haa02] beschrieben werden, unterstützt.

Im Vergleich zu [Lie01] liegt der Schwerpunkt dieser Arbeit auf der Konzeption eines auf Aggregation basierenden Autorenprozesses, mit dessen Hilfe Lernobjekte für selbstgesteuertes Lernen entwickelt werden können. Dabei handelt es sich auch nicht um die Produktion von Multimodulmodulen im Sinne von Macromedia Director [17] und SumTotal ToolBook [28] sondern das Erstellen von strukturierten Kursen (standardkonformen WBTs) unter Einbeziehung von multimedialen Elementen.

Die Wiederverwendung bereits bestehender Lernobjekte erfolgt rein manuell im Autorenprozess ohne diesen durch eine Ergänzung des Autorenprozesses durch automatische Generierung von WBTs zu ergänzen, die nach [SSR⁺99] auf Basis von rhetorisch didaktischen Relationen [See02] in Verbindung mit einer fachspezifischen Ontologie, die sich nach [Faa04] semiautomatisch erstellen lassen, realisiert werden könnte. Zur Herstellung der Kohärenz der auf Basis wiederverwendeter Lernobjekte zusammengestellter Kurse werden keine automatisch generierten Zwischentexte, wie in [See02] beschrieben, eingesetzt. Stattdessen werden Wege untersucht, die Kohärenz der zu einem Kurs zusammengestellten Lernobjekte durch manuelle Anpassung in einer Autorenumgebung herzustellen.

Die Adaption der Kurse in adaptiven Systemen, wie sie in [SN04a] beschrieben werden, auf Basis einer Benutzermodellierung [Kob85] unter Verwendung rhetorisch didaktischer Relationen [HS02a], ist ebenfalls kein Forschungsgegenstand dieser Ar-

beit. Ebenso erfolgt keine explizite didaktische Auszeichnung der Lernobjekte, wie sie beispielsweise in [Med03] beschrieben wird, so dass die im Rahmen dieser Arbeit entwickelten Konzepte auch für Anwendungen geeignet sind, die keinen didaktischen Hintergrund haben.

1.4 Aufbau der Arbeit

In Kapitel 2 erfolgt eine vertiefte Motivation der Arbeit und eine Einführung in die der Arbeit zugrunde liegende Terminologie. Hierbei werden grundlegende, für das Verständnis der vorliegenden Arbeit notwendige, Begriffe definiert. Anschließend werden Problem- und Zielstellung der Arbeit detailliert beschrieben.

In Kapitel 3 erfolgt eine Einordnung der Arbeit in das wissenschaftliche Umfeld. Hierfür werden verwandte Ansätze und ähnliche Autorensysteme mit Möglichkeit der Wiederverwendung von Lernobjekten verglichen. Dabei wird insbesondere deren Eignung zur Realisierung eines auf Aggregation basierenden Autorenprozesses, der in Kapitel 2 definiert wird, untersucht.

An Hand einer systematischen Analyse werden allgemeine Anforderungen zur Wiederverwendung von Lernobjekten und Anforderungen an einen auf Aggregation basierenden Autorenprozess in Kapitel 4 untersucht. Insbesondere wird dabei die Wiederverwendung von Lernobjekten im Autorenprozess berücksichtigt. Ausgehend von dieser Untersuchung wird ein generisches Dokumentmodell für die Speicherung von Lernobjekten in modularisierter Form entwickelt. Dieses Dokumentmodell bildet die Grundlage für die Entwicklung eines abstrakten Prozessmodells zur Beschreibung eines auf Aggregation basierenden Autorenprozesses.

Metadaten spielen bei der Wiederverwendung von Lernobjekten eine Schlüsselrolle, weil sie für die Indizierung von Lernobjekten herangezogen werden. Aus diesem Grund wird in Kapitel 5 ein Verfahren für die Generierung von Metadaten entwickelt, mit dessen Hilfe weitgehend automatisch Metadaten für neu erstellte Lernobjekte generiert werden können, so dass letztere direkt nach ihrer Fertigstellung wiederverwendet werden können.

Die Wiederverwendung von Lernobjekten, die sich in einer ständigen Weiterentwicklung und Bearbeitung befinden können, macht eine Versionierung der Lernobjekte notwendig. In Kapitel 6 wird deswegen die Versionierung modularisierter Lernobjekte beschrieben. Dabei werden insbesondere die Aggregationsbeziehungen zwischen Fragmenten von Lernobjekten berücksichtigt.

Kapitel 7 beschreibt das ResourceCenter, die Implementierung der in der Arbeit entwickelten Konzepte für die Wiederverwendung von Lernobjekten in einer auf Aggregation basierenden Autorenumgebung. Hier wird die von dem abstrakten Prozessmodell abgeleitete Autorenumgebung und das daran angebundene Lernobjektarchiv vorgestellt. Beide bilden eine gemeinsame Basis für die Wiederverwendung von Lernobjekten im Autorenprozess. In das ResourceCenter eingebunden ist

der Metadaten-Wizard, die Implementierung des in Kapitel 5 beschriebenen Verfahrens zur semiautomatischen Erstellung von Metadaten.

Auf Basis von Daten, die im produktiven Einsatz der Implementierung in einem Forschungsprojekt gesammelt worden sind, erfolgt in Kapitel 8 die Evaluation der im Rahmen der Arbeit entwickelten und umgesetzten Konzepte. Die Evaluation beinhaltet die Betrachtungen zur Nutzung der Autorenumgebung für die Erstellung und Wiederverwendung von Lernobjekten, sowie zur Arbeitserleichterung, die durch die Verwendung des Metadaten-Wizards erzielt wurde.

Kapitel 9 fasst die in der vorliegenden Arbeit gewonnenen Ergebnisse zusammen und gibt im Ausblick Anknüpfungspunkte für weitere Arbeiten auf dem Gebiet der „Wiederverwendung von digitalen Lernobjekten in einem auf Aggregation basierenden Autorenprozess“.

2 Wiederverwendung von Lernobjekten

In diesem Kapitel erfolgt eine vertiefte Motivation der Arbeit und eine Einführung in die der Arbeit zugrunde liegende Terminologie. Hierfür werden der Begriff der Wiederverwendung selbst und andere grundlegende Begriffe, die im Rahmen der Arbeit verwendet werden, definiert. Anschließend wird das Ziel und die Problemstellung der vorliegenden Arbeit detailliert beschrieben.

2.1 Wiederverwendung

Die Erstellung und Verfügbarkeit qualitativ hochwertiger Lernmaterialien ist eine wichtige Voraussetzung dafür, dass sich die rechnergestützte Lehre auf Basis von Materialien für das selbstgesteuerte Lernen in der Breite durchsetzt. In der Regel ist die Erstellung der Materialien jedoch ein sehr aufwändiger und kostenintensiver Prozess, in dem Autoren, Dozenten, Designer und gegebenenfalls Pädagogen zusammenarbeiten müssen [Urb03]. Durch Wiederverwendung einzelner Materialien, beziehungsweise durch die mehrfache Verwendung vollständiger Lernmaterialien, sollen die Produktionskosten der Lernmaterialien gesenkt beziehungsweise der Return of Investment erhöht werden.

Zwei Arten von Wiederverwendung von Lernmaterialien können unterschieden werden. Die gebräuchliche Weise ist die mehrfache Verwendung identischer Lernmaterialien für verschiedene Lehrveranstaltungen oder Lerner. Die Produktionskosten der Lernmaterialien lassen sich dann verteilen auf die Mehrfachverwendung der Lernmaterialien in mehreren Lehrveranstaltung oder auf die Anzahl der Teilnehmer beziehungsweise der Lernenden. Bei dieser Art von Wiederverwendung werden die Materialien unverändert als komplette Einheit wiederverwendet. Die andere Möglichkeit besteht in der Wiederverwendung von Lernmaterialien bei der Produktion neuer Lernmaterialien. Hierbei werden für die Erstellung neuer Materialien Teile bereits bestehender Lernmaterialien wiederverwendet. Hierdurch können die Produktionskosten je nach Anteil wiederverwendeter Lernmaterialien teilweise erheblich reduziert werden.

In der vorliegenden Arbeit wird die Wiederverwendung von Lernmaterialien bei der Produktion oder genauer bei der Erstellung von Kursen zum selbstgesteuerten Lernen in der rechnergestützten Lehre betrachtet. Die Kurse werden im Folgenden

Web Based Training (WBT) genannt. Die Anwendung der Wiederverwendung von Materialien lässt sich weiterhin in zwei Teilbereiche gliedern: die Wiederverwendung der Materialien in unveränderter und in für die Wiederverwendung angepasster Form. Beide Arten der Wiederverwendung werden in der vorliegenden Arbeit betrachtet.

Aus diesem Zusammenhang ergibt sich für die vorliegende Arbeit folgende Definition für den Begriff Wiederverwendung:

Definition 2.1 (Wiederverwendung) *Wiederverwendung ist die mehrfache Verwendung (von Teilen) bereits bestehender Materialien bei der Erstellung neuer Lernmaterialien. Materialien können sowohl unverändert als auch in angepasster Form wiederverwendet werden.*

Durch die Wiederverwendung sollen Aufwand und Kosten für die Produktion neuer WBTs reduziert werden. Bei der Erstellung neuer Lernmaterialien können durch Wiederverwendung Ressourcen eingespart werden. Diese können für die Produktion neuer Teile der neu zu erstellenden Lernmaterialien genutzt werden [Ruy01], so dass bei gleichem Mitteleinsatz durch die Wiederverwendung mehr Ressourcen in die Entwicklung neuer Teile investiert werden können.

Ein Praxisbeispiel für die Wiederverwendung bei der Erstellung von Lernmaterialien findet sich in der Weiterbildung von Ärzte- und Pflegeberufen. Sowohl Geburtshelfer, Kinderkrankenpfleger, als auch Kinderärzte müssen sich in regelmäßigen Abständen weiterbilden. Aufgrund der Schnittmenge der Themen für die Weiterbildung von Hebammen, Kinderkrankenschwestern und Kinderärzten, ist in der Regel die Wiederverwendung vieler Lernmaterialien bei der Produktion neuer Lernmaterialien möglich. Hierdurch können der Aufwand und dementsprechend die Kosten für die Produktion von WBTs für die Weiterbildung von Angehörigen dieser Berufsgruppen reduziert werden. [Ruy01]

Die Wiederverwendung bestehender Lernmaterialien bei der Erstellung neuer Materialien ist jedoch nicht trivial. Mit Hilfe herkömmlicher Autorenwerkzeuge können oft nur monolithische Lernmaterialien erstellt werden, die entweder nur ganz oder gar nicht genutzt werden können [AM02]. Bei der Verwendung dieser Autorenwerkzeuge werden neu zu erstellende Materialien deshalb oft von Grund auf neu entwickelt. Bestehende Materialien können in diesen Autorensystemen in der Regel nur über Kopieren von wiederverwendbaren Materialien und Einfügen in neu zu erstellende Materialien wiederverwendet werden. Suchmechanismen zum Auffinden für die Wiederverwendung geeigneter Materialien werden in der Regel nicht angeboten.

Zur Verbesserung der Wiederverwendung bereits bestehender Materialien bei der Erstellung neuer Materialien ist eine weitergehende Unterstützung der Wiederverwendung in den Autorensystemen erforderlich. Diese wird durch die Anbindung eines Archivs an das Autorensystem erreicht, über das bestehende Materialien für

die Wiederverwendung abgerufen werden können. Durch die Ergänzung des Autorensystems um Suchfunktionen können Materialien gefunden werden, die für die Wiederverwendung geeignet sind. Für eine Steigerung der Effizienz bei der Wiederverwendung bereits bestehender Materialien sollten Autorensysteme die Wiederverwendung bereits bestehender Materialien sowohl im Originalzustand, als auch in angepasster Form unterstützen. Damit Materialien sowohl komplett, als auch teilweise wiederverwendet werden können, ist die Speicherung der Materialien in einem Datenformat notwendig, das die partielle Wiederverwendung von Materialien unterstützt. Alle Lernmaterialien, die mit dem Autorensystem erstellt werden, können in diesem Datenformat im Archiv abgelegt und so für die Wiederverwendung verfügbar gemacht werden.

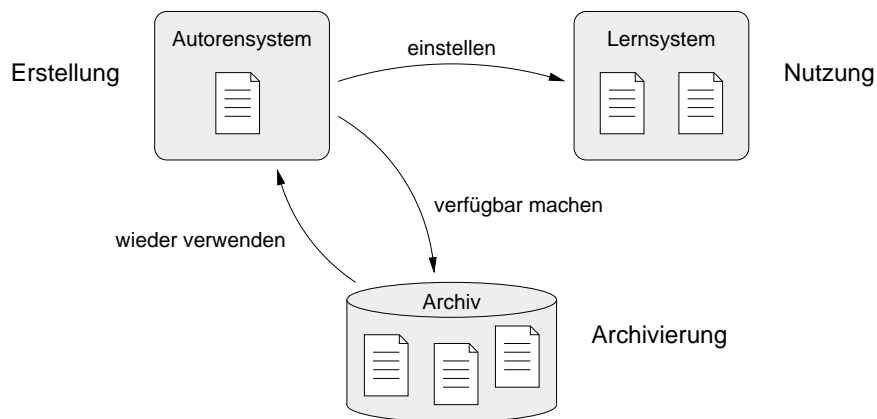


Abbildung 2.1: Wiederverwendung bei der Erstellung neuer Lernmaterialien

Abbildung 2.1 zeigt die Wiederverwendung bereits bestehender Materialien bei der Erstellung neuer WBTs. Der Weg der erstellten WBTs vom Autorensystem hin zum Lernsystem bleibt unverändert erhalten. Hinzu kommt das Archiv zur Archivierung einmal erstellter Materialien. In dem Archiv werden alle Lernobjekte gespeichert, die mit dem Autorensystem erstellt wurden. Die Lernobjekte stehen danach für die Wiederverwendung im Autorensystem bereit. Mit Hilfe von Suchwerkzeugen können für die Wiederverwendung geeignete Materialien im Archiv gesucht werden. Bereits bestehende Materialien können dann beim Erstellen neuer Materialien durch Abrufen aus dem Archiv wiederverwendet werden. Hierfür ist jedoch eine organisatorische sowie solide technische Unterstützung des Wiederverwendungsprozesses notwendig [AH04].

In den folgenden Abschnitten werden verschiedene Begriffe definiert, die in direktem Zusammenhang mit der Wiederverwendung von Lernmaterialien stehen. Sie bilden die Grundlage für die Definition eines Autorenprozesses für die Wiederverwendung von Lernmaterialien.

2.2 Lernobjekte

Die Wiederverwendung von Lernmaterialien ist Inhalt einer Reihe internationaler Forschungsinitiativen. Das gemeinsame Ziel dieser Arbeiten ist es, die Wiederverwendung von Lernmaterialien in Form von Lernobjekten zu verbessern. Da sich bisher keine allgemein verwendete Definition eines Lernobjekts herauskristallisiert hat, wird im Folgenden anhand verschiedener existierender Definitionen der Begriff Lernobjekt eingegrenzt.

Das Konzept der Lernobjekte ist zunächst in den Arbeiten zum Learning Object Metadata (LOM) Standard [LOM02] sehr allgemein definiert worden. Im IEEE Standard werden Lernobjekte als „... *any entity, digital or non-digital, that may be used for learning, education or training*“ beschrieben [LOM02]. Diese Definition fasst den Begriff Lernobjekt jedoch sehr weit. Nach dieser Definition sind beispielsweise auch alle Gegenstände Lernobjekte, die sich als Anschauungsobjekt für das Lernen eignen. So ist eine Dampfmaschine nach dieser Definition ein geeignetes Lernobjekt für die Physik. Diese Definition ist für die vorliegende Arbeit nicht geeignet, weil nur digitale Lernobjekte elektronisch in einem Autorensystem wiederverwendet werden können.

Polsani [Pol03] definiert Lernobjekte mit Bezug zur Wiederverwendung wie folgt: „*A Learning Object is an independent and self-standing unit of learning content that is predisposed to reuse in multiple instructional contexts.*“ Damit definiert er Lernobjekte als unabhängige Einheiten von Lernmaterial, die für sich (allein) stehen können und für die Wiederverwendung in mehreren instruktionalen Kontexten bestimmt sind. Diese Definition macht keine Aussage darüber, ob Lernobjekte digitale oder nicht digitale Elemente sind, legt jedoch fest, dass Lernobjekte unabhängige Einheiten darstellen, die für sich allein stehen können und für die Wiederverwendung bestimmt sind. Diese Erweiterung der Definition schränkt die Menge der Lernobjekte auf solche ein, die unverändert wiederverwendet werden können. Da in der vorliegenden Arbeit die Wiederverwendung von Lernobjekten auch in angepasster Form vorgesehen ist, würde diese Definition die Menge der Lernobjekte unnötig einschränken.

Im Gegensatz zu der Definition von Lernobjekten nach Polsani definiert Wiley in [Wil02] Lernobjekte wie folgt: „*A learning object is any digital resource that can be reused to support learning.*“ Diese Definition schränkt Lernobjekte auf beliebige digitale Ressourcen ein, die für die Unterstützung rechnergestützter Lehre wiederverwendet werden können. Die Definition macht keine Aussagen darüber, ob Lernobjekte allein stehen können müssen und ob sie ein Lernziel haben müssen. Damit bleibt die Definition bezüglich der Typen von Lernobjekten offen, schränkt sie aber auf digitale Ressourcen ein. Allerdings schließt diese Definition Lernmaterialien aus, die nicht wiederverwendet werden können.

In Anlehnung an die Definition von Wiley [Wil02], werden Lernobjekte für die vorliegende Arbeit wie folgt definiert:

Definition 2.2 (Lernobjekt) *Ein Lernobjekt ist eine beliebige digitale Ressource, die für die Erstellung von Lernmaterialien verwendet werden kann.*

Nach dieser Definition sind analog zu Wiley Lernobjekte als beliebige digitale Ressourcen definiert. Ein Lernobjekt muss weder ein Lernziel besitzen, noch für sich allein stehen können. Lernziele für Lernobjekte und alleinstehende Lernobjekte sind jedoch nicht ausgeschlossen. Die Wiederverwendbarkeit von Lernobjekten ist eine wünschenswerte Eigenschaft, die jedoch nicht als Kriterium für Lernobjekte herangezogen werden kann. Aus diesem Grund werden im Unterschied zu der Definition von Wiley auch digitale Ressourcen als Lernobjekte zugelassen, die nicht wiederverwendet werden können. Im Vergleich zu Polsani müssen Lernobjekte nach dieser Definition weder ein Lernziel besitzen, noch für sich alleine stehen können. Hierdurch wird die Menge der Lernobjekte auf Rohdaten, wie beispielsweise Texte und Bilder erweitert. Diese Erweiterung ist wichtig, da in der vorliegenden Arbeit auch die Wiederverwendung beliebiger Lernobjekte, also auch von Rohdaten untersucht werden soll.

2.3 Modularisierung und Aggregation

Verbreitete Autorenwerkzeuge erlauben häufig nur das Erstellen monolithischer Lernobjekte. Diese können jedoch in der Regel nur für den ursprünglichen Zweck oder gar nicht wiederverwendet werden [AM02]. Diese Lernobjekte sind nicht für die Wiederverwendung nach Definition 2.1 geeignet, weil sie sich nicht partiell für die Erstellung neuer Lernobjekte wiederverwenden lassen. Für die Wiederverwendung zum Zweck der Erstellung neuer Lernobjekte auf Basis von Teilen monolithischer Lernobjekte sind diese zunächst in Einzelteile zu zerlegen. Die Zerlegung von Lernobjekten in Einzelteile wird Modularisierung genannt.

Definition 2.3 (Modularisierung) *Modularisierung ist die Zerlegung eines Lernobjekts in Subkomponenten.*

Lernobjekte können sowohl aus inhaltlichen als auch aus technischen Gründen nicht in beliebig kleine Subkomponenten modularisiert werden. Aus diesem Grund muss zwischen Lernobjekten, die modularisiert werden können, und atomaren Lernobjekten, die nicht modularisiert werden können, unterschieden werden. Atomare Lernobjekte brechen die Rekursion, die sich durch die Modularisierung von Lernobjekten ergibt, ab.

Definition 2.4 (Atomares Lernobjekt) *Atomare Lernobjekte können nicht durch Modularisierung in weitere Subkomponenten zerlegt werden.*

Das Ergebnis der Modularisierung eines Lernobjekts sind eine Reihe von Subkomponenten, aus deren Zusammenführung das modularisierte Lernobjekt besteht. Diese Subkomponenten werden Lernobjektfragmente genannt. Beispiele für Lernobjektfragmente sind Lektionen, Abschnitte und, analog zu [CH01], Textstücke und einzelne Bilder. Lernobjektfragmente stellen nach Definition 2.2 ebenfalls Lernobjekte dar, die selbst wieder in Lernobjektfragmente modularisiert werden können, sofern sie kein atomares Lernobjekt darstellen.

Definition 2.5 (Lernobjektfragment) *Die Subkomponenten, die durch die Modularisierung von Lernobjekten entstehen, werden Lernobjektfragmente genannt.*

Das Ziel der Modularisierung von Lernobjekten ist die Wiederverwendung einzelner Lernobjektfragmente. Neue Lernobjekte lassen sich durch die Kombination bestehender modularisierter Lernobjekte und neu erstellter Lernobjektfragmente erstellen. Hierfür werden die Lernobjektfragmente des neuen Lernobjekts zunächst geordnet beziehungsweise arrangiert, sie werden zusammengestellt. Anschließend werden sie in dem vom Autor gewählten Arrangement zu einem neuen Lernobjekt verknüpft. Das ist die Aggregation von Lernobjekten.

Definition 2.6 (Aggregation) *Die Aggregation von Lernobjekten ist das Zusammenstellen verschiedener Lernobjektfragmente und deren Verknüpfung zu einem neuen Lernobjekt.*

Das Ergebnis der Aggregation von Lernobjekten sind stets neue Lernobjekte, die sich aus der Gesamtheit der einzelnen Lernobjektfragmente zusammensetzen. Durch die Summation der einzelnen Lernobjektfragmente sind die aus einer Aggregation hervorgehenden Lernobjekte komplexer.

Zur Beurteilung der Größe von Lernobjekten lassen sich verschiedene Metriken, wie beispielsweise die Größe in Bytes oder die durchschnittliche Lerndauer in Minuten, heranziehen.

Ein wichtiges Maß für die Beurteilung der Größe eines Lernobjekts ist jedoch die Komplexität gemessen an der Schachtelungstiefe der Aggregation des Lernobjektes. Die Definition einer diskreten Skala von Komplexitäten für aggregierte Lernobjekte kann als Klassifikationsschema für Lernobjekte genutzt werden. Die diskretisierte Komplexität von Lernobjekten gemessen an der Schachtelungstiefe der Aggregation wird Aggregationsniveau genannt. Beispiele für Aggregationsniveaus sind: Kurs, Kursmodul, Absatz und Rohmaterialien.

Definition 2.7 (Aggregationsniveau) *Das Aggregationsniveau ist die diskretisierte Komplexität von Lernobjekten gemessen an der Schachtelungstiefe der Aggregation.*

Zur Veranschaulichung der Modularisierung und der Aggregation von Lernobjekten wurden verschiedene Metaphern entwickelt. Beispielsweise wurden Lernmodule häufig mit Lego-Bausteinen verglichen [MAI00]. Nach dieser Metapher lassen sich neue Lernobjekte (Bauwerke) durch das einfache Zusammenstecken von Lernmodulen (Lego-Bausteinen) erstellen. Das Problem dieser Metapher allerdings ist, dass sich Lernmodule im Gegensatz zu Lego-Bausteinen nicht mit jedem beliebigen anderen Lernmodul beziehungsweise Lego-Baustein verknüpfen lassen. In [Wil02] wird deswegen die Modularisierung mit der Metapher der Atome und Moleküle verglichen. Die wesentlichen Eigenschaften dieser Metapher sind, dass sich Lernmodule nicht beliebig kombinieren lassen und, dass Wissen und Übung dazu gehört, um Lernmodule zu neuen Lernobjekten zu kombinieren.

2.4 Aggregation als Basis für Autorenprozesse

Bei einem auf Aggregation basierenden Autorenprozess handelt es sich um einen Prozess, in welchem Lernobjekte durch die Aggregation von Lernobjektfragmenten erzeugt werden. Dabei werden für die Erstellung neuer Lernobjekte sowohl bereits bestehende, als auch neu erstellte Lernobjektfragmente im Autorenprozess aggregiert [BA04]. Ein auf Aggregation basierender Autorenprozess stellt demzufolge eine Form der Erstellung von Lernobjekten dar, bei der das Wiederverwenden einmal erstellter Lernobjekte zum grundlegenden Paradigma wird.

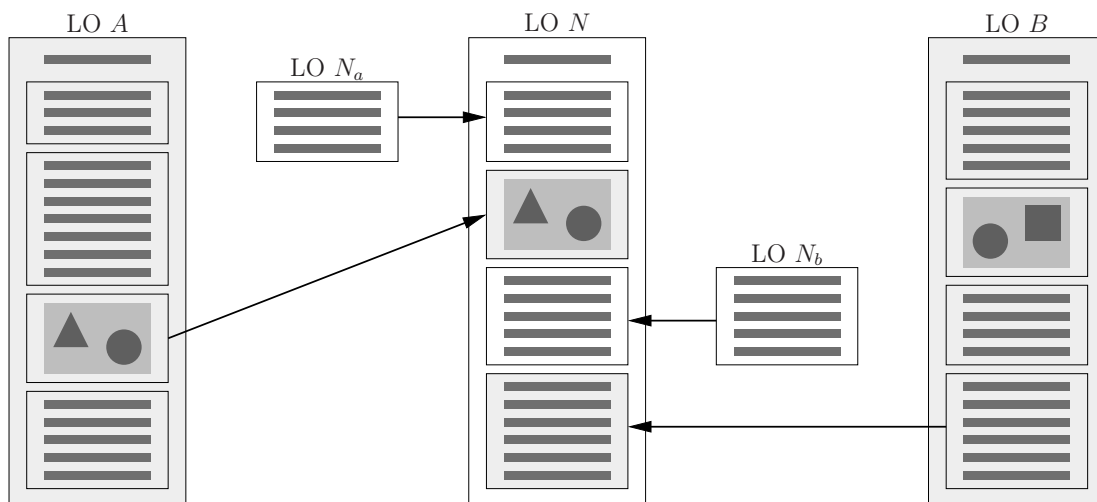


Abbildung 2.2: Aggregation eines neuen Lernobjekts (LO)

Abbildung 2.2 skizziert die Erstellung des Lernobjekts in einem auf Aggregation basierenden Autorenprozess. Bereits bestehende Lernobjekte sind grau, neu erstellte Lernobjekte dagegen sind weiß hinterlegt. In dem Autorenprozess wird

das Lernobjekt N durch eine Zusammenstellung verschiedener Lernobjektfragmente aggregiert. Lernobjekt N setzt sich aus vier Fragmenten zusammen. Das erste Lernobjektfragment N_a ist ein Absatz, der eigens für die Erstellung von Lernobjekt N neu erzeugt wird. Im Anschluss an den Text wird das Bild aus Lernobjekt A wiederverwendet. Danach folgen zwei weitere Absätze. Der nächste Absatz, Lernobjekt N_b , wird ebenfalls wie der erste Absatz eigens für die Erstellung von Lernobjekt N neu geschrieben. Für den dritten Absatz wird der letzte Absatz von Lernobjekt B wiederverwendet. Wie das Beispiel zeigt, ermöglicht ein auf Aggregation basierender Autorenprozess die Verwendung von Lernobjekten aus verschiedenen Quellen [AB04, Mas03].

Analog zu diesem Beispiel lassen sich bei der exemplarischen Produktion von WBTs für die Weiterbildung von Kinderärzten, Kinderkrankenschwestern und Hebammen Lernobjekte wiederverwenden. Für die Wiederverwendung kommen Lernobjekte in Frage, deren Inhalte thematisch zu mindestens zwei der zu erstellenden WBTs passen. Dabei kann die Wiederverwendung von Lernobjekten schon bei deren Erstellung geplant werden.

Für die vorliegende Arbeit ergibt sich folgende Definition für einen auf Aggregation basierenden Autorenprozess:

Definition 2.8 (Aggregation als Basis von Autorenprozessen) *Ein auf Aggregation basierender Autorenprozess ist ein Autorenprozess, der die Erstellung von Lernobjekten auf Basis von Aggregation (im Sinne der Definition 2.6) unterstützt. Das Erstellen neuer Lernobjekte basiert dabei auf der Aggregation von Lernobjektfragmenten. Die Aggregation von Lernobjekten ist nicht auf die Wiederverwendung, das heißt auf die Aggregation bereits verwendeter Lernobjekte, beschränkt, sondern erlaubt auch die Aggregation bisher noch nicht verwendeter Lernobjekte. Die Wiederverwendung bereits bestehender Lernobjektfragmente sieht sowohl die unveränderte als auch die angepasste Verwendung von Lernobjektfragmenten vor.*

In der Definition bleiben die Aggregationsniveaus der erstellten Lernobjekte und Lernobjektfragmente unbestimmt. Der Grund hierfür ist, dass ein auf Aggregation basierender Autorenprozess auf allen Aggregationsniveaus einer modularen Dokumentstruktur angewendet werden kann. Der Zusammenhang zwischen den Aggregationsniveaus des Lernobjekts und den aggregierten Lernobjektfragmenten ergibt sich durch die Aggregation selbst. Dabei umfasst das Ergebnis einer Aggregation die Summe aller Lernobjektfragmente und ist in der Regel vom Aggregationsniveau um eine Stufe höher als das der einzelnen Lernobjekte.

Üblicherweise liegen Lernobjekte nicht untergliedert in Fragmenten vor. Für die Wiederverwendung von Fragmenten eines Lernobjekts in einem auf Aggregation basierenden Autorenprozess ist deswegen eine vorherige Modularisierung der Lernobjekte erforderlich, mit dem Ziel, die benötigten Lernobjektfragmente zu erstellen. Dieser erforderliche Arbeitsschritt stellt einen Nachteil für die Verarbeitung von

Lernobjekten, die mit herkömmlichen Autorenwerkzeugen erstellt worden sind, in einem auf Aggregation basierenden Autorenprozess dar.

Im Gegensatz hierzu wird in der vorliegenden Arbeit ein auf Aggregation basierender Autorenprozess beschrieben, bei dem die Fragmente neu erstellter Lernobjekte nicht zu monolithischen Lernobjekten verschmolzen werden. Für die partielle Wiederverwendung von Lernobjekten bleiben so die Lernobjektfragmente erhalten, ohne dass die entsprechenden Lernobjekte erneut modularisiert werden müssen.

Das heißt, in einem auf Aggregation basierenden Autorenprozess ist die Modularisierung von Lernobjekten für die Wiederverwendung von Lernobjektfragmenten nicht erforderlich, wenn die Lernobjektfragmente nach der Aggregation erhalten bleiben.

Die sukzessive Aggregation von Lernobjekten in einem auf Aggregation basierenden Autorenprozess lässt komplett modularisierte Lernobjekte mit hohen Aggregationsniveaus, wie beispielsweise Kurse, entstehen. Diese modularisierten Lernobjekte lassen sich in Verbindung mit einem auf Aggregation basierenden Autorenprozess in Teilen wiederverwenden oder durch den Austausch einzelner Teile für andere Verwendungszwecke anpassen. So kann durch den Einsatz von Autorenprozessen, die auf Aggregation basieren, eine enorme Arbeitserleichterung und Kostenersparnis erzielt werden.

2.5 Ziel und Problemstellung

Die Wiederverwendung bereits bestehender Lernobjekte in einem auf Aggregation basierenden Autorenprozess kann als ein zukunftsweisendes Verfahren für das Erstellen neuer Lernobjekte angesehen werden. So soll der häufig sehr große Aufwand für die Erstellung von Lernobjekten reduziert werden [AML00]. Aus diesem Grund werden in der vorliegenden Arbeit verschiedene technische Aspekte für die Umsetzung eines auf Aggregation basierenden Autorenprozesses untersucht. Ziel dabei ist die Konzeption eines Autorenprozesses, mit dessen Hilfe die Erstellung von neuen Lernobjekten basierend auf bestehenden Lernobjekten und neu erstellten Lernobjektfragmenten erfolgen kann. Hierfür werden in der vorliegenden Arbeit verschiedene Probleme als Voraussetzung eines auf Aggregation basierenden Autorenprozesses untersucht und gelöst. Zur Überprüfung der entwickelten Konzepte sollen sie anschließend in einem System zusammengeführt und in ihrer Nutzung evaluiert werden.

Abbildung 2.3 zeigt ein Systemszenario, welches die Wiederverwendung von Lernobjekten in einem auf Aggregation basierenden Autorenprozess ermöglicht. Die Akteure in diesem Anwendungsszenario sind Autoren, die bestehende Lernobjekte, auch von anderen Autoren, wiederverwenden möchten. Ziel der Autoren ist das Erstellen von Lernobjekten, die für das selbstgesteuerte Lernen geeignet sind. Diese erstellen sie mit Hilfe eines Autorensystems, das einen auf Aggregation basierenden

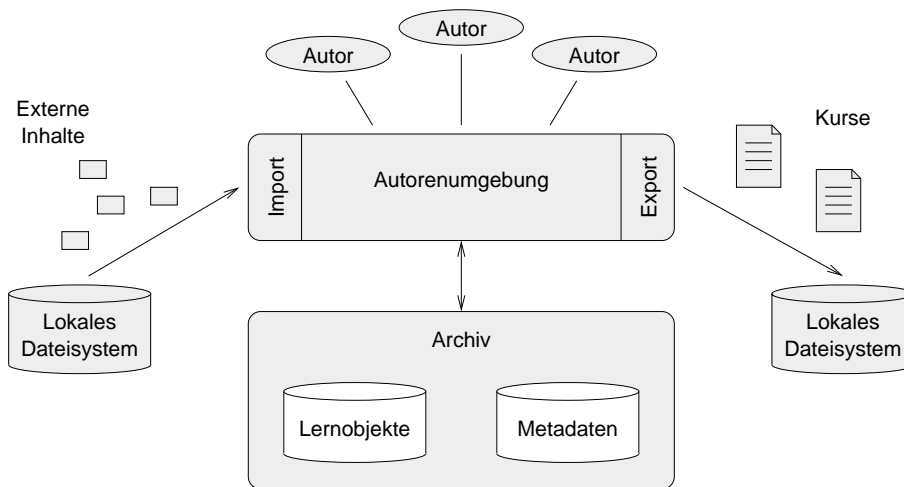


Abbildung 2.3: Autorensystem zur Aggregation von Lernobjekten in Autorengruppen

Autorenprozess unterstützt. Das Autorensystem ist mit Import und Exportfunktionen ausgestattet, mit deren Hilfe es möglich ist, bereits bestehende Lernobjekte oder Lernobjekte, die mit lokalen Werkzeugen erstellt worden sind, in das Autorensystem zu importieren, beziehungsweise das Exportieren erstellter Lernobjekte, wie Kurse zu ermöglichen. Quelle beziehungsweise Ziel der Import- und Exportfunktion ist jeweils das lokale Dateisystem¹ der Autoren. Bereits bestehende Lernobjekte werden allen Autoren über ein Archiv, das mit dem Autorensystem verbunden ist, für die Wiederverwendung verfügbar gemacht. Lernobjekte, die mit dem Autorensystem erstellt worden oder lediglich in das Autorensystem importiert worden sind, werden ebenfalls über das an das Autorensystem angeschlossene Archiv für die Wiederverwendung verfügbar gemacht. Zur Wahrung der Privatsphäre werden hierbei Mechanismen erforderlich mit deren Hilfe der Zugriff auf Lernobjekte eingeschränkt werden kann. In dem Archiv werden die modularisierten Lernobjekte zusammen mit ihren Metadaten gespeichert. Metadaten sind Informationen über die Lernobjekte, mit deren Hilfe gespeicherte Lernobjekte im Archiv gefunden und verwaltet werden können.

Für die Realisierung eines auf Aggregation basierenden Autorensystems sind zunächst auf Aggregation basierende Autorenprozesse selbst zu untersuchen. Grundlage für Autorenprozesse, die auf Aggregation basieren, ist die Verfügbarkeit modularisierter Lernobjekte beziehungsweise von Lernobjektfragmenten. Aus dieser Anforderung heraus ergibt sich für die vorliegende Arbeit die Notwendigkeit ei-

¹Alternativ zu dem lokalen Dateisystem, können auch beliebige Dateisysteme, die auf Computernetzwerken basieren, genutzt werden. Im Folgenden wird aufgrund der besseren Lesbarkeit stellvertretend für alle denkbaren Dateisysteme, auf die von den Arbeitsplatzrechnern der Autoren zugegriffen werden kann, nur das lokale Dateisystem genannt.

nes Dokumentformats, das geeignet ist modularisierte Lernobjekte zu speichern, die sowohl in einem auf Aggregation basierenden Autorenprozess wiederverwendet werden sollen, als auch in einem auf Aggregation basierenden Autorenprozess erstellt worden sind. Hierbei stehen die zur modularen Speicherung notwendigen Eigenschaften von Dokumentfragmenten aggregierter Lernobjekte im besonderen Interesse der Untersuchung.

Auf Aggregation basierende Autorenprozesse haben die Wiederverwendung von Lernobjekten als Paradigma. Für die Wiederverwendung von Lernobjekten müssen diese jedoch zuvor vorbereitet werden. Mit Hilfe der Vorbereitung zur Wiederverwendung neu erstellter Lernobjekte sollen diese dem Autorenprozess für die Wiederverwendung bereitgestellt werden. Dazu gehört neben der Modularisierung der Lernobjekte und dem Verfügbarmachen der modularisierten Lernobjektfragmente, die Beschreibung der Lernobjektfragmente mit Metadaten, um die Lernobjektfragmente im Archiv wieder finden zu können. Die Vorbereitung der Wiederverwendung von Lernobjekten in einem auf Aggregation basierenden Autorenprozess bezieht sich dabei sowohl auf Lernobjektfragmente, als auch auf Lernobjekte, die aus den Lernobjektfragmenten zusammengestellt werden und selbst wiederverwendet werden können. Das Erstellen der Metadaten ist jedoch ein sehr aufwändiger Prozess. Ziel der Betrachtung in der vorliegenden Arbeit ist deswegen, die Erstellung von Metadaten weitestgehend zu automatisieren.

Ein weiterer wichtiger Aspekt für die Realisierung eines auf Aggregation basierenden Autorensystems ist die Versionierung der modularisierten Lernobjekte. Diese ist erforderlich, weil Lernobjekte im Autorenprozess weiterentwickelt und auch in angepasster Form wiederverwendet werden können. Damit sich die Änderungen eines Lernobjekts für die Verwendung an einer Stelle nicht auf alle anderen Stellen im Gesamtsystem übertragen, ist eine Versionierung der Lernobjekte notwendig.

2.6 Zusammenfassung

In diesem Kapitel wurden die Wiederverwendung von Lernobjekten bei der Erstellung neuer Lernobjekte und die Wiederverwendung von Lernobjekten durch mehrfachen Einsatz in Lehrveranstaltung unterschieden.

Der Fokus dieser Arbeit liegt in der Wiederverwendung von Lernobjekten für die Erstellung neuer Lernobjekte, weshalb Wiederverwendung entsprechend definiert wurde. Die Wiederverwendung umfasst hierbei sowohl die Verwendung bestehender Lernobjekte in unveränderter, als auch in modifizierter Form. So sollen mittels der Wiederverwendung von Lernobjekten die Kosten für die Produktion neuer Lernobjekte reduziert werden.

Die Wiederverwendung von Lernobjekten ist jedoch nicht trivial. Verbreitete Autorenwerkzeuge behindern die Wiederverwendung häufig durch die Verwendung monolithischer Datenformate zur Speicherung der Lernobjekte und bieten selten

Suchfunktionen, mittels der bereits bestehende Lernobjekte wiedergefunden werden können. Für die Verbesserung der Wiederverwendung in Autorensystemen wurde deshalb eine Anbindung an ein Archiv gefordert, das dazu genutzt werden soll, bereits bestehende und im Rahmen des Autorenprozesses neu erstellte Lernobjekte, die wiederverwendet werden können, abzulegen und zu finden. Zur Speicherung der Lernobjekte im Archiv wurde ein Datenformat gefordert, das auch eine partielle Wiederverwendung der Lernobjekte erlaubt. Steht ein solches Archiv und ein geeignetes Datenformat zur Speicherung modularisierter Lernobjekte in dem Autorsystem bereit, lässt sich ein auf Aggregation basierender Autorenprozess realisieren. Hierbei werden bei der Erstellung von Lernobjekten Lernobjektfragmente aggregiert.

Ziel der vorliegenden Arbeit ist die Entwicklung von Konzepten zur Realisierung eines auf Aggregation basierenden Autorenprozesses. Hierbei wird insbesondere die Wiederverwendung existierender Lernobjekte zusammen mit neu erstellten Lernobjektfragmenten untersucht. In diesem Zusammenhang ist ein geeignetes Dokumentformat zu definieren, mittels dessen Lernobjekte modular gespeichert werden können, so dass sie partiell in einem auf Aggregation basierenden Autorenprozess wiederverwendet werden können (vgl. Kapitel 4). Mittels einer weitestgehend automatischen Erstellung von Metadaten für neu erstellte Lernobjekte sollen diese der Wiederverwendung für die Autoren leichter zugeführt werden können (vgl. Kapitel 5). Die Versionierung von Lernobjekten, die im Archiv gespeichert sind, schließlich soll das Verwenden und Wiederverwenden von Lernobjekten in verschiedenen Versionen ermöglichen (vgl. Kapitel 6).

Im nächsten Kapitel werden zu den genannten Themen verwandte Arbeiten vorgestellt, um die Arbeit genauer einzuordnen und offene Problembereiche genauer zu definieren.

3 Verwandte und grundlegende Arbeiten

Das folgende Kapitel gibt einen Überblick über verwandte Arbeiten zur Wiederverwendung von Lernobjekten und deren Fragmenten im Autorenprozess. Da bisher nur wenige Systeme die in Abschnitt 2.5 beschriebenen Eigenschaften besitzen, werden in diesem Kapitel verwandte Arbeiten zur Wiederverwendung von Lernobjekten und deren Fragmenten aufgegriffen. Hierfür werden zunächst verschiedene Ansätze für die Modularisierung von Lernobjekten betrachtet, die die Basis für die Wiederverwendung von Lernobjektfragmenten darstellt. Nachdem verschiedene Prinzipien für die Modularisierung von Lernobjekten und Wiederverwendung von Lernobjektfragmenten diskutiert werden, werden verschiedene Modelle für modularisierte Lernobjekte verglichen. Anschließend erfolgt ein Vergleich verschiedener Datenmodelle für die Speicherung modularisierter Lernobjekte. Dabei werden die Ergebnisse aus der Betrachtung der Modularisierung berücksichtigt. Danach werden bestehende Autorenwerkzeuge und Ansätze zur Aggregation von Lernobjekten unter Berücksichtigung geeigneter Dokumentformate verglichen. Besonders wichtig bei diesem Vergleich sind jedoch der Grad der Unterstützung der Wiederverwendung von Lernobjekten und deren Fragmenten im Autorenprozess. Da das Wiederverwenden von Lernobjekten in der Regel auf aussagekräftigen Metadatenätzen beruht, deren Erstellung in der Regel jedoch viel Aufwand erfordert, werden anschließend verwandte Arbeiten für eine weitestgehend automatisierte Erstellung von Metadatenätzen untersucht. Zum Schluss der Betrachtung verwandter Arbeiten werden Lernobjektarchive, mit deren Hilfe bestehende und wiederverwendbare Lernobjekte beziehungsweise Lernobjektfragmente im Autorenprozess verfügbar gemacht werden sollen, auf ihre Eignung zur Unterstützung eines auf Aggregation basierenden Autorenprozesses untersucht.

3.1 Modularisierung von Lernobjekten

Für die Wiederverwendung von Lernobjekten in einem auf Aggregation basierenden Autorenprozess werden modularisierte Lernobjekte benötigt. In diesem Abschnitt werden verschiedene existierende Konzepte und Verfahren für die Modularisierung bestehender Lernobjekte diskutiert.

Die Modularisierung von Lernobjekten ist kein trivialer Vorgang. Problematisch

hierbei ist insbesondere die geeignete Zerlegung der Lernobjekte in Einzelteile und das Finden der passenden Größe eines Moduls. Um Lernobjekte in geeignete Lernobjektfragmente modularisieren zu können, werden von Boyle in [Boy02] drei Prinzipien identifiziert: Das erste Prinzip der Kohäsion verlangt, dass jedes Lernobjekt nur einen Lerngegenstand mit einem Lernziel zum Inhalt haben soll. Das zweite Prinzip der Entkopplung fordert, dass ein Lernobjekte nur so wenig wie möglich Querverweise zu anderen Lernobjekten aufweisen darf. Das dritte und letzte Prinzip betrachtet Lernobjekte als alleinstehende Einheiten, deren Inhalte bei Wiederverwendung in einem anderen Lernkontext noch vollständig zu verstehen sind.

Diese drei inhaltsbezogenen Prinzipien können für die Modularisierung von Lernobjekten verwendet werden, wenn sie als Maß für die Größe von Lernobjektfragmenten angewendet werden. Das ist jedoch nur in begrenztem Umfang möglich, weil mit sinkendem Aggregationsniveau der Lernobjekte schnell die Grenze erreicht wird, ab der Boyles Prinzipien nicht mehr als Eigenschaften von Lernobjektfragmenten angewendet werden können. Dann helfen oft nur noch technische Eigenschaften für die Modularisierung von Lernobjekten. Ein Beispiel hierfür sind multimediale Lernobjekte, die nach Steinmetz und Nahrstedt [SN02] aus statischen und kontinuierlichen Medien bestehen. Die verschiedenen Medientypen von multimedialen Lernobjekten eignen sich für eine Modularisierung auf technischer Basis an Hand der per Definition [SN02] vorhandenen Wechseln verwendeter Medien.

Das Aggregationsniveau (siehe Definition 2.6) eines Lernobjekts hat einen bedeutenden Einfluss auf dessen Wiederverwendbarkeit. Lernobjekte mit kleinerem Aggregationsniveau können leichter wiederverwendet werden, da sie im Vergleich zu Lernobjekten mit hohem Aggregationsniveau weniger an einen bestimmten Kontext gebunden sind. Nachteilig bei der Modularisierung von Lernobjekten in Lernobjektfragmente mit sehr kleinem Aggregationsniveau ist aber der hohe Aufwand für die Organisation und Verwaltung vieler kleiner Lernobjekte [AH04]. Bei der Wahl der Aggregationsstufe eines Lernobjekts muss also ein Kompromiss zwischen potenzieller Wiederverwendbarkeit, Lernkontext und Organisationsaufwand von Lernobjekten gefunden werden [Wil02].

South und Monson visualisieren eine abstrakte Form dieses Kompromisses in [SM02] mit Hilfe eines Granularitäts- und Aggregationsspektrums. Abbildung 3.1 zeigt das dazu äquivalente Aggregationsniveauspektrum. Das Spektrum reicht von Lernobjekten mit hohem Aggregationsniveau, wie ganzen Kursen, bis hinunter zu Rohmaterialien, die Lernobjekte mit niedrigem Aggregationsniveau entsprechen. Innerhalb dieses Spektrums führen South und Monson die Lernschwelle ein, ab der Lernobjekte mit steigendem Aggregationsniveau instruktional nutzbar sind. Zusätzlich führen sie die Kontextschwelle ein, ab der sich Lernobjekte mit sinkendem Aggregationsniveau aufgrund des schwachen bis fehlenden Bezugs zu einem bestimmten Kontext besonders leicht durch Multimediaentwickler wiederverwenden lassen. Kontextschwelle und Lernschwelle spannen modellhaft einen Aggregationsbeziehungsweise Granularitätsbereich auf, innerhalb dessen Lernmodule sowohl von

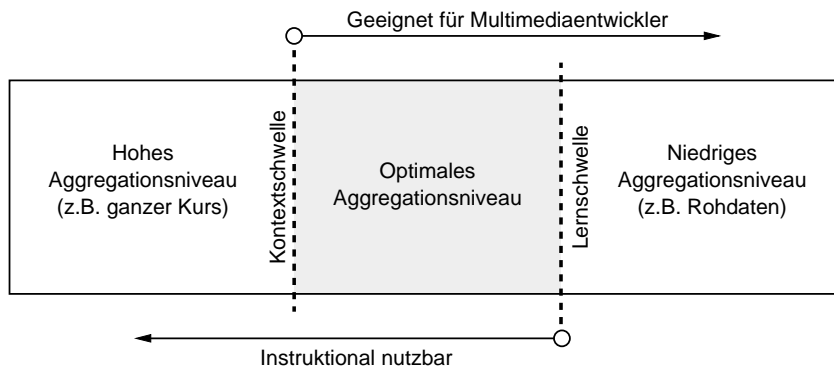


Abbildung 3.1: Wiederverwendung im Aggregationsniveauspektrum (Quelle: South und Monson [SM02])

Multimediaentwicklern als auch von Dozenten wiederverwendet werden können.

Nach South und Monson haben Lernobjekte für die Wiederverwendung das ideale Aggregationsniveau, wenn es zwischen der Kontextschwelle und der Lernschwelle liegt. Lernobjekte mit diesem Aggregationsniveau sind sowohl instruktional, als auch durch Multimediaentwickler wiederverwendbar. Das Aggregations- und Granularitätsspektrum von South und Monson zeigt aber auch, dass sämtliche Aggregationsniveaus von Lernobjekten für die Wiederverwendung geeignet sind. Lernobjekte mit hohem Aggregationsniveau links von der Kontextschwelle lassen sich dabei bevorzugt instruktional, Lernobjekte mit kleinen Aggregationsniveaus rechts von der Lernschwelle bevorzugt von Multimediaentwicklern wiederverwenden.

Damit eine partielle Wiederverwendung von Lernobjekten mit hohen Aggregationsniveaus auch durch Multimediaentwickler ermöglicht werden kann, ist eine geeignete Modularisierung der Lernobjekte erforderlich, ohne dass dadurch der Gesamtzusammenhang des Lernobjekts verloren geht. Die Verfügbarkeit von Lernobjekten mit einem breiten Spektrum von Aggregationsniveaus, die sowohl von Dozenten als auch von Multimediaentwicklern wiederverwendet werden können, ist ein wichtiges Kriterium für die Wiederverwendung modularisierter Lernobjekte. Für die Modularisierung von Lernobjekten gibt es jedoch kein einheitliches Konzept. Die wichtigsten Modularisierungskonzepte zur Klassifikation der Aggregationsniveaus werden im Folgenden vorgestellt.

Das Modularisierungskonzept des Unternehmens Cisco System Inc. sieht hierfür zwei Aggregationsniveaus für Lernobjekte vor. Es wird zwischen *Reusable Information Objects* (RIO) und *Reusable Learning Objects* (RLO) unterschieden [Cis03]. Elemente vom Typ RIO stellen bei diesem Schema Lern-, Übungs- und Testeinheiten dar. RLOs setzen sich aus mehreren RIOs, einer Einleitung und Zusammenfassung zusammen und bilden eine abgeschlossene Lerneinheit ähnlich einer abgeschlossenen Unterrichtsstunde. In [Cis03] wird die Aggregation von Modulen und Kursen angedeutet, jedoch nicht weiter beschrieben.

Im Standard für *Learning Object Metadata* [LOM02], der von dem *Learning Technology Standards Committee* [12] zur Beschreibung von Lernobjekten entwickelt wurde, werden für die Klassifikation von Lernobjekten insgesamt vier Aggregationsniveaus eingeführt. Lernobjekte wie beispielsweise Texte, Bilder oder andere Fragmente, werden nach dem Standard dem kleinsten Aggregationsniveau 1 zugeordnet. Lernobjekte mit dem Aggregationsniveau 2 setzen sich aus Lernobjekten mit dem Aggregationsniveau 1 zusammen. Sie sind mit Lektionen vergleichbar. Aus einer Reihe von Lektionen setzen sich Kurse zusammen. Lernobjekte, die mit Kursen verglichen werden können, haben das Aggregationsniveau 3. Lernobjekte, die sich aus einer Sammlung von Kursen zusammensetzen, werden dem höchsten Aggregationsniveau 4 zugeordnet. Mit diesen vier Aggregationsniveaus wird im LOM-Standard ein Klassifikationsschema beschrieben, das geeignet ist, Lernobjekte mit verschiedenen Aggregationsniveaus zu klassifizieren. Hierfür wird jedoch ein Modulkonzept benötigt, mit dessen Hilfe Lernobjekte mit verschiedenen Aggregationsniveaus instanziiert werden können.

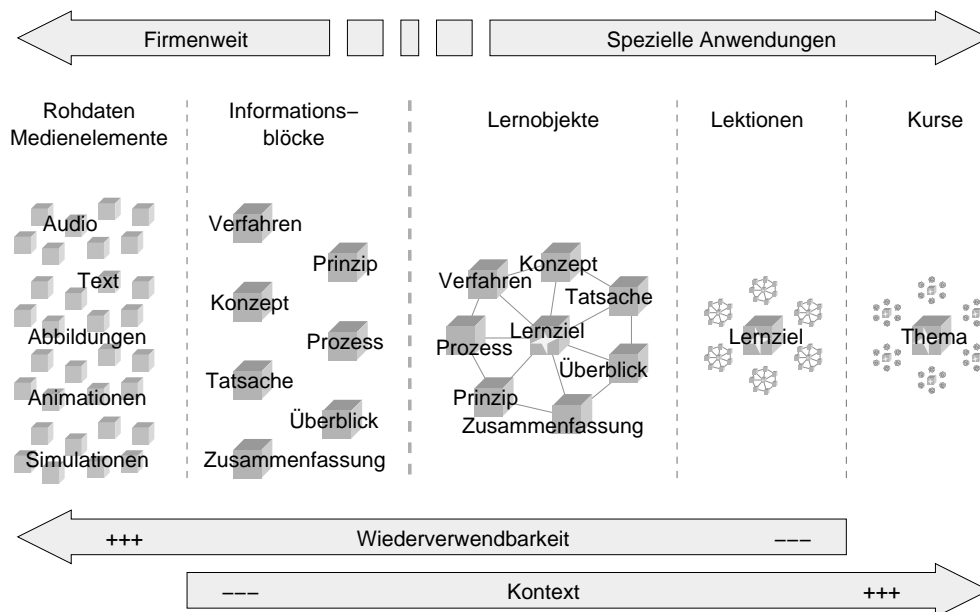


Abbildung 3.2: Hierarchie für modulare Lernobjekte (Quelle: Hodgins [Hod02])

Hodgins schlägt in [Hod02] ein grundlegendes Schema zur Klassifikation von Lernobjekten vor. Abbildung 3.2 zeigt die verschiedenen Aggregationsniveaus, die dieses Modell definiert. Rohdaten oder einfache Medienelemente haben das kleinste Aggregationsniveau in diesem Modell. Diese Elemente befinden sich ausschließlich auf der Datenebene. Beispiele hierfür sind einzelne Sätze, Absätze, Bilder oder Animationen. Informationsblöcke sind eine Menge von Medienelementen, die beispielsweise auf dem generalisierten Informationsblockmodell von Horn [Hor98] basieren können.

Horn beschreibt mit dem Informationsblockmodell ein Schema zur Strukturierung von Textdokumenten. Angewendet auf das Modell von Duval und Hodgins lassen sich danach Medienelemente und Informationsblöcke zu RIOs kombinieren, weil sie kein Lernziel besitzen. Ihre Wiederverwendung erfolgt in der Regel firmenweit, das heißt innerhalb einer geschlossenen Organisation. Objekte mit höheren Aggregationsniveaus sind Lernobjekte, Lektionen und Kurse. Sie basieren auf der sukzessiven Aggregation nächst kleinerer Objekte und sind den RLOs zuzuordnen, weil sie ein Lernziel beziehungsweise Thema beinhalten. Aufgrund der durch die Lernziele und Themen aufgebauten Zusammenhänge, können Objekte dieser Aggregationsniveaus in der Regel nur in speziellen Anwendungen wiederverwendet werden. Mit steigendem Zusammenhang sinkt die Wiederverwendbarkeit der Objekte des hierarchischen Objektmodells von Duval und Hodgins. Das Aggregationsniveau mit dem Namen „Lernobjekt“ darf hierbei nicht mit Lernobjekten nach Definition 2.2 verwechselt werden.

Für die partielle Wiederverwendung von Lernobjekten werden Modularisierungskonzepte mit geeigneten Aggregationsniveaus und einer durchgängigen Aggregation der Lernobjekte benötigt. Das Modularisierungskonzept von Cisco [Cis03] reicht jedoch nicht für die Klassifikation verschiedener Aggregationsniveaus in modularisierten Kursen aus, weil es nur zwei Aggregationsniveaus definiert. Das Schema zur Klassifikation der Aggregationsniveaus von Lernobjekten des LOM-Standards enthält dagegen vier Aggregationsniveaus. Von diesen sind jedoch nur drei für Kurse und deren Lernobjektfragmente vorgesehen, so dass nicht genügend Aggregationsniveaus zur Verfügung stehen, um beispielsweise Informationsobjekte, eine Aggregation von Medienobjekten nach Duval und Hodgins zu differenzieren. Die Unterscheidung zwischen Informations- und Lernobjekten ist jedoch auch nach [Cis03] wichtig, so dass aus diesem Grund das Klassifizierungsschema des LOM-Standards nicht geeignet ist.

Im Vergleich dazu sieht die Hierarchie modularer Lernobjekte von Duval und Hodgins fünf Aggregationsniveaus für Lernobjekte wie Kurse und deren Lernobjektfragmente vor. Dabei decken die Aggregationsniveaus lückenlos Lernobjekte mit sehr niedrigen bis hin zu hohen Aggregationsniveaus ab. Zwischen allen Aggregationsniveaus werden Aggregationsbeziehungen definiert. Das Konzept ist jedoch sehr abstrakt und besitzt ebenso wie die abstrakte Komponentenarchitektur von Verbert et al. [VD04, VKM⁺04] keine erkennbare Möglichkeit für eine tiefere Schachtelung eines Kursdokuments, die durch Rekursion von Aggregationsbeziehungen erreicht werden kann.

Eine Anwendung für die vorgestellten Modularisierungskonzepte ist die Bestimmung von Fragmenten bei der automatischen Modularisierung von bestehenden Lernobjekten. Die durch das verwendete Modulkonzept definierten Aggregationsniveaus werden dazu verwendet, den einzelnen Fragmenten, bei der Modularisierung ein Aggregationsniveau zuzuordnen und die einzelnen Fragmente durch die im Modulkonzept definierten Aggregationsbeziehungen in Zusammenhang zu bringen.

Ein Beispiel für die Modularisierung bestehender Lernobjekte für die partielle Wiederverwendung wird von Cardinaels und Duval [CD03] beschrieben. Ausgangspunkt sind Präsentationsfolien im Powerpoint-Format der Firma Microsoft. In der Arbeit wird die Aufspaltung der Foliensätze in einzelne Folien und die Erstellung von Metadatensätzen, die die einzelnen Folien beschreiben, gezeigt. Eine andere Anwendung ist das Segmentieren von Videos auf Basis von Schnittdetektion nach Günsel et al. [GFT98].

Zur Speicherung dieser Fragmente reicht jedoch ein Modulkonzept allein nicht aus, vielmehr wird ein modulares Dokumentformat benötigt, mit dessen Hilfe Lernobjekte und deren Fragmente gespeichert werden können. Im nächsten Abschnitt werden verschiedene bestehende Dokumentformate verglichen.

3.2 Datenmodelle für Lernobjekte

Ein Merkmal der rechnergestützten Lehre ist, dass Lernobjekte elektronisch gespeichert und ausgegeben werden. Zur Speicherung der Lernobjekte wird daher ein Datenmodell benötigt, das den Anforderungen des Anwendungszwecks der Lernobjekte in der rechnergestützten Lehre gerecht wird. Für verschiedenartige Lernobjekte und Zielanwendungen wurden eine Vielzahl von Datenmodellen entwickelt. Diese lassen sich nach Art der Anwendungen klassifizieren. Zu den wichtigsten Klassen verfügbarer Dokumentmodelle gehören Containerformate für den Datenaustausch, Datenmodelle für das Cross-Media-Publishing, spezielle Multimediaformate, Formate, die die Adaption der Lernobjekte unterstützen, und solche, die eine pädagogische Auszeichnung erlauben.

Ein großer Teil verfügbarer Dokumentmodelle sind für den Austausch von Lernobjekten konzipiert worden. Sie stellen Containerformate dar, die dazu geeignet sind, Lernobjekte zu beschreiben und zu kapseln, so dass sie zwischen verschiedenen Systemen und Firmen hinweg ausgetauscht werden können. Zu diesen Containerformaten gehört das IMS Content Packaging (IMS CP) [IMS01a] des IMS Global Learning Consortiums (IMS) [14]. Die Spezifikation von IMS CP beschreibt die Kapselung von Lernobjekten. Mit Hilfe einer XML-Beschreibung können die Fragmente in eine Struktur gebracht und referenziert werden. Bezüglich der zu kapselnden Lernobjekte stellt die Spezifikation lediglich die Forderung, dass die einzelnen Fragmente Web-Content sein müssen. Basierend auf IMS CP spezifiziert die Advance Distributed Learning Initiative (ADL) [1] das Sharable Content Object Reference Model (SCORM) [Adv01] unter Verwendung weiterer Standards und Spezifikationen. Ebenso, wie die *Computer Managed Instruction* (CMI) Spezifikation [AIC04] des Aviation Industry CBT Committees (AICC) [3], ist das Ziel der SCORM-Spezifikation ein Containerformat für den Austausch von WBTs zwischen Inhaltsersteller und Anwender zu schaffen. Alle drei Containerformate bieten trotz Aggregationsmodell technisch nicht die Möglichkeit der Aggregation von Lernob-

jektfragmenten mit niedrigen Aggregationsniveaus. Zudem ist die Wiederverwendung der Lernobjekte in Form von Containerformaten durch die Kodierung der Fragmente als Web-Content in der Regel auf die Wiederverwendung in unmodifizierter Form beschränkt, weil deren Anpassung und Erweiterung oft mit hohem Aufwand verbunden ist.

Dieses Problem soll mit den Datenmodellen, die für das Cross-Media-Publishing entwickelt wurden, gemildert werden. Beim Cross-Media-Publishing werden Inhalte getrennt vom Layout gespeichert, so dass die Inhalte ohne manuellen Zusatzaufwand auf verschiedenen Ausgabemedien ausgegeben werden können. Datenmodelle für das Cross-Media-Publishing basieren in der Regel auf den Metasprachen *Standard Generalized Markup Language* (SGML) [ISO86] und *Extensible Markup Language* (XML) [XML04]. Die für das Cross-Media-Publishing typische Überführung der Lernobjekte in andere Formate für ihre Distribution in/auf verschiedenen Medien wird in der Regel auf Basis der *Document Style Semantics and Specification Language* (DSSSL) [ISO96] für SGML beziehungsweise *Extensible Stylesheet Language* (XSL) [XSL01] für XML durchgeführt. *DocBook* [WM05] ist eine komplexe XML-basierte Auszeichnungssprache für strukturierte Bücher, die insbesondere für das Erstellen von Büchern über Computer Hard- und -Software optimiert worden ist und derzeit bei der *Organization for the Advancement of Structured Information Standards* (OASIS) [22] gepflegt wird. Die *Darwin Information Typing Architecture* (DITA) [Pri01] ist eine flexible und erweiterbare XML-Architektur von IBM [6] für das Erstellen und Wiederverwenden technischer Dokumente. Das Datenmodell von DITA unterstützt die Wiederverwendung von Themen durch Referenz in unmodifizierter Form, so dass bei der Korrektur die referenzierten Teile nur ein Mal geändert werden müssen. In [Pri01] wird dieses Konzept skalierbare Wiederverwendung genannt. Außerdem erlaubt DITA die rekursive Aggregation von Themen. Die *Dissertation Markup Language* (DiML) [Dob99][27], die an der Humboldt Universität in Berlin entwickelt worden ist, ist ein Archivformat mit dessen Hilfe elektronische Dissertationen in Universitätsbibliotheken integriert werden sollen. Alle drei Dokumentformate ermöglichen die Distribution über die Wege Print und Online.

Für die rechnergestützte Lehre gibt es weitere Dokumentformate, die auf dem Ansatz der Cross-Media-Publishing-Formate basieren. Aufgrund spezieller pädagogischer Auszeichnungsmöglichkeiten bieten sie jedoch zusätzliche Möglichkeiten zur Adaptierung der Lernobjekte in einem gewissen Rahmen. Hierzu zählt die *Learning Material Markup Language* (LMML) [SFB99], die ein Modell-basiertes XML-Framework zur semantischen Beschreibung von Lernmaterialien darstellt. Es erlaubt eine vielseitige Anpassung und Erweiterung für einzelne Fachgebiete, so dass es bereits verschiedene LMML-Sprachen gibt [Fre02]. Ähnliche Ziele werden mit *TeachML* [Tee02] verfolgt. Konzeptionell ist die Ähnlichkeit von LMML und TeachML so hoch, dass sie mittels des Sprachmodells TLMM [Tee01] zusammengeführt werden können. Die *Multidimensional Learning Objects and Modular Lectures Markup Language* (<ML>³) [LTV03] verfolgt die Wiederverwendung von Lernobjekten

durch gezielte Anpassung der Lernobjekte vor der Nutzung. Die Möglichkeit zur Anpassung der Lernobjekte wird durch die gleichzeitige Modellierung verschiedener Varianten der Lernmaterialien erreicht, so dass verschiedene Ausprägungen der Lernobjekte erzeugt werden können. Verbreitete Ausgabeformate aller vier Verfahren sind Handreichungen auf Papier, WBTs und Folien.

Datenmodelle für die Kodierung von Multimediapräsentationen unterscheiden sich deutlich von den bisher genannten Datenmodellen durch die zusätzliche temporale Beschreibung der Inhalte. Die Synchronized Multimedia Integration Language (SMIL) [SMI05] stellt ein Datenmodell für synchronisierte Multimediapräsentation für das Web dar, bietet jedoch im Vergleich zum ZYX Multimediadokumentmodell (ZYX) [BK99] keine Modularisierung der Inhalte an. ZYX ist mit speziellen Anforderungen an Wiederverwendung, Adaptierbarkeit, Interaktion und präsentationsneutraler Beschreibung für Multimediainhalte entworfen worden. Das ZYX Datenmodell basiert auf Präsentationselementen, die einzelne Knoten der baumförmigen Dokumentstruktur darstellen und die mittels Identifikatoren miteinander verknüpft werden können. Dabei repräsentieren die Präsentationselemente sowohl atomare Elemente, wie Texte, Bilder und Videos, als auch komplexe Kombinationen von Medienelementen. Mit Hilfe von Operatorelementen können temporale Synchronisation, Interaktion, Adaptierbarkeit und räumliches und audiovisuelles Layout des Multimediadokuments beschrieben werden. Dokumentmodelle für die Kodierung von Multimediapräsentationen verfolgen aufgrund der zusätzlichen temporalen Beschreibung einen anderen Ansatz als Dokumentmodelle für die statische Präsentation von Lernobjekten. Das ZYX-Modell bietet ein geeignetes, auf Aggregation basierendes Konzept für die Modularisierung wiederverwendbarer Lernobjekte. Es bietet jedoch keine Konzepte für die Versionierung einzelner Dokumentfragmente, so dass die Wiederverwendung in modifizierter Form in einem auf Aggregation basierenden Autorenprozess nur schwer möglich ist.

Ein dem ZYX-Modell relativ ähnliches Dokumentmodell ist in dem Projekt *Live Long Learning* (L3) als Teil einer organisatorischen und technischen Infrastruktur zur Sicherung lebensbegleitender Weiterbildung entwickelt worden [Ger03]. Als Teil der technischen Infrastruktur ist ein Dokumentmodell für Lernmaterialien entwickelt worden [GA03]. Dieses Dokumentmodell basiert auf einem Aggregationsmodell, das verschiedene Aggregationsniveaus für Lernobjekte beinhaltet. Die technische Infrastruktur und das Dokumentmodell von L3 sind jedoch nicht für die Aggregation von Rohmaterialien ausgelegt.

Das *IMS Learning Design* (IMS LD) [IMS03b] deren Entwicklung auf der Educational Modeling Language (EML) [HKL⁺00] basiert ist eine Beschreibungssprache für didaktische Konzeptionen von Lernobjekten. Sie kann genutzt werden, um zusammen mit LOM Metadaten und IMS Content Packaging Lehrveranstaltungen zu planen und formal zu beschreiben. IMS LD und EML eignen sich jedoch nicht für die Auszeichnung von Lernmaterialien, da sie den Fokus auf die Beschreibung von pädagogischen Szenarien legen.

Für einen auf Aggregation basierenden Autorenprozess ist ein Dokumentformat notwendig, das geeignet ist, Lernobjekte bis hin zu Aggregationsniveaus von Rohmaterialien modularisiert zu speichern. Keines der vorgestellten existierenden Dokumentformate unterstützt jedoch die direkte Adressierbarkeit und die Beschreibung von Lernobjekten auf dem Aggregationsniveau von Rohmaterialien mit Metadaten. Trotzdem besitzen einige der genannten Dokumentformate Eigenschaften, die ein geeignetes Dokumentformat für die Wiederverwendung fragmentierter Lernobjekte beim Erstellen von WBTs haben sollte. Dazu gehören die modularisierte Speicherung von Lernobjektfragmenten nach ZYX, jedoch ohne die Synchronisierungsoperatoren, sowie das Wiederverwenden von Lernobjekten durch Referenzieren nach DITA, um eine skalierbare Wiederverwendung zu erzielen [Pri01]. Außerdem ist die Versionierung von Lernobjektfragmenten erforderlich, die im Fall von L3 für das lokale Bearbeiten von Lernobjekten optimiert ist. Für die Auszeichnung von Textfragmenten lassen sich schließlich in DocBook geeignete Strukturen finden, mit deren Hilfe mittels gängiger Cross-Media-Publishing-Verfahren die Lernobjekte für verschiedene Ausgabemedien erzeugt werden können.

3.3 Autorenwerkzeuge

Aus dem in Kapitel 2 beschriebenen Anwendungsszenario und den Zielen der Arbeit ergeben sich folgende Kriterien für Autorensysteme: Das Autorensystem soll für die Produktion von WBTs eingesetzt werden können und die Aggregation von Lernobjekten ermöglichen. Mit Hilfe dieses auf Aggregation basierenden Autorenprozesses sollen außerdem Lernobjekte aller Aggregationsstufen inklusive Textfragmenten und anderen Rohmaterialien erstellt werden können. Für nicht textbasierte Rohmaterialien können spezialisierte Werkzeuge eingesetzt werden, da deren Nachimplementierung als Teil des Autorensystems zu kostenintensiv ist. Der auf Aggregation basierende Autorenprozess soll sowohl die Wiederverwendung bereits bestehender Lernobjekte in der Originalfassung als auch in modifizierter Fassung ermöglichen. Dabei muss eine Wiederverwendung von Lernobjekten aller Aggregationsniveaus in allen drei Bereichen des durch South und Monson [SM02] aufgespannten Aggregationspektrums, der durch die Kontext- und die Lernschwelle in drei Bereiche geteilt wird, ermöglicht werden.

Die verwandten Autorenwerkzeuge lassen sich in verschiedene Klassen gruppieren. In der ersten Gruppe sollen die Dokumenterstellungssysteme vorgestellt werden, mit deren Hilfe Dokumente nur auf Basis bestehender Inhalte erstellt werden können. Neue Inhalte können bei diesem Ansatz nur nachträglich mit externen Werkzeugen hinzugefügt werden. Zu diesen Systemen ist das *System for COntent REuse* (SCORE) [KAKRM03] zu zählen. Bei SCORE werden zunächst auf Basis einer domänenspezifischen Ontologie Themen ausgesucht. Die Zuordnung der Inhalte erfolgt automatisch nach der Wahl eines didaktischen Templates [AKKRM03].

Die Erstellung einer Wissensbasis, mit deren Hilfe eine automatische Auswahl von Inhalten erfolgt, ist jedoch nach unseren eigenen Erfahrungen, die wir im Projekt k-MED [Szi05] gemacht haben, sehr aufwändig. Ein anderer Ansatz wird deswegen im *ALOCoM-Framework* [VJD⁺05] verfolgt. Bestehende Lernobjekte werden bei dem Import in das System automatisch modularisiert, indexiert und schließlich in das interne Komponentenmodell [VD04, VKM⁺04] überführt. Im Gegensatz zu SCORE können Lernobjektfragmente für die Erstellung neuer Lernobjekte mit Hilfe schlüsselwortbasierter Suchen ausgewählt werden.

Ebenso wie bei ALOCoM können mit dem *CourseEditor* [IGdLR⁺03] bestehende Inhalte zusammengestellt und exportiert werden. Ein anderer Ansatz wird mit dem *Environment for Learning and Teaching* (ELAT) [BRG⁺04] verfolgt. Dabei werden die Inhalte unter Verwendung spezieller Templates mit externen Werkzeugen erstellt und in das Content Management System importiert. Die Wiederverwendung bestehender Lernobjekte wird in ELAT jedoch mangels Suchmöglichkeiten noch nicht unterstützt.

Alle Systeme haben jedoch den Nachteil, dass bestehende Inhalte nur im ursprünglichen Zustand wiederverwendet werden können. Änderungen und Erweiterungen können nur mit Hilfe externer Werkzeuge vorgenommen werden, nachdem die neu erstellten Lernobjekte exportiert worden sind. Damit geänderte und erweiterte Lernobjekte wiederverwendet werden können, müssen sie wieder in diese Autorensysteme importiert werden. Problematisch dabei ist, dass unveränderte Lernobjekte beim Import in das System dupliziert werden, solange das für die Bearbeitung der Lernobjekte verwendete Datenformat nicht das Verfolgen von Änderungen der Lernobjekte, zum Beispiel mittels Identifikatoren der Fragmente, erlaubt.

In [GAP02] wird ein systematischer Ansatz für die Produktion einfacher Kursunterlagen beschrieben. Bestehende Lernobjekte werden bei diesem Ansatz zunächst modularisiert und anschließend arrangiert. Fehlende Fragmente werden in einem iterativen Prozess erstellt und eingefügt. Aufgrund fehlender Strukturierungsmöglichkeiten beziehungsweise Schachtelungstiefen im Autorenprozess ist der beschriebene Ansatz nicht für die Produktion von komplexen WBTs geeignet.

Im deutschen Leitprojekt *Lebenslanges Lernen* (L3) [GA03] wird die Wiederverwendung bestehender Lernobjekte in modifizierter Form unter Verwendung externer Autorenwerkzeuge durch die Versionierung der Lernobjekte unterstützt. Mit Hilfe eines integrierten Editors können Lernpfade zwischen den einzelnen Lernobjektfragmenten auf Basis sachlogischer Relationen und verschiedener didaktischer Strategien erstellt werden. Problematisch bei beiden Ansätzen ist, dass die Wiederverwendung nur von Lernobjekten mit hoher bis mittlerer Granularität unterstützt wird. Der Grund hierfür liegt darin, dass das Aggregationsniveau der kleinsten adressierbaren Lernobjekte größer als das von Rohmaterialien ist.

Eine weitere Gruppe relevanter Arbeiten stellen Lernsysteme mit integrierten Autorensystemen dar. Ilias open source [13] ist eine an der Universität Köln entwickelte, weit verbreitete Lernplattform, die neben den gewöhnlichen Lernplattform-

funktionalitäten eine Autorenumgebung zur Erstellung von WBTs bietet. Dabei ist die Wiederverwendung von Lernobjektfragmenten in Ilias zwar möglich, wird jedoch wiederum dadurch erschwert, dass Metadaten nicht für Objekte aller Aggregationsniveaus erfasst werden.

Zu Werkzeugen mit anderen Zielen ist der Phoenix Editor [YVT⁺04, FMMW05] zu zählen, der im Ariadne Projekt [2] entwickelt worden ist. Mit dessen Hilfe können importierte HTML-Dokumente auf konzeptueller Basis mit pädagogischen Strukturen in Hypertextdokumenten ausgezeichnet und in XML gespeichert werden. Der *Reload Editor* [24] ist ein weit verbreitetes Werkzeug zum Erstellen von IMS Content Packaging Paketen inklusive dazugehöriger *IMS Learning Resource Meta-Data* [IMS01b] Metadatensätze. Der Reload-Editor ist im Prinzip kein Autorenwerkzeug für die Erstellung von Lernmaterialien selbst.

Von allen vorgestellten Systemen werden viele der Kriterien, die zu Beginn des Abschnitts aufgestellt wurden, nicht erfüllt:

Die Wiederverwendung von Lernobjekten in allen verfügbaren Aggregationsniveaus ist bei den meisten vorgestellten Arbeiten nicht gewährleistet. In der Regel liegt dies an der fehlenden Adressierbarkeit von Lernobjekten mit niedrigen Aggregationsstufen, wie die für Rohmaterialien (SCORE, L3, Ariadne, ELAT). Nicht adressierbare Lernobjekte lassen sich einzeln schlecht finden und somit insgesamt nicht befriedigend wiederverwenden. Außerdem skaliert ihre Wiederverwendung nicht, weil mit jeder Kopie zusätzlicher Aufwand für deren Pflege entsteht [Pri01].

Die Auszeichnung der Lernobjekte aller Aggregationsniveaus mit Metadaten ist ebenfalls bei vielen Arbeiten nicht erfüllt (SCORE, L3, ELAT, [GAP02]). Bestehende Lernobjekte werden dann in der Regel über einem Katalog verfügbar gemacht (Ilias). Bei einer größeren Anzahl von Lernobjekten können diese jedoch oft nicht mehr wiedergefunden werden. Aus diesem Grund ist die Auszeichnung der Lernobjekte aller Aggregationsstufen mit Metadaten für das Auffinden und Wiederverwenden bereits bestehender Lernobjekte besonders wichtig.

Aufgrund fehlender Software-Werkzeuge für die Bearbeitung aggregierter Lernobjekte aller Aggregationsstufen können Lernobjekte oft nur in unmodifizierter Form wiederverwendet werden (Ariadne, [GAP02], CourseEditor, ALOCoM). Das trifft vor allem auf die Arbeiten aus der automatischen Dokumentgenerierung zu (SCORE), die zwar über reichhaltige Auszeichnung der Lernobjekte verfügen, um diese zusammenzustellen, jedoch eine Bearbeitung der zusammengestellten Inhalte nur mit externen Werkzeugen ermöglichen. Doch gerade die Wiederverwendung von Lernobjekten in modifizierter Form erweitert die Möglichkeiten der Wiederverwendung von Lernobjekten in entscheidender Weise. Bis auf L3 bietet jedoch kein Autorensystem eine hierfür notwendige Versionierung der Lernobjekte an.

3.4 Metadaten und Metadatengenerierung

Metadaten werden im Allgemeinen als Daten bezeichnet, die Informationen über andere Daten enthalten [LOM02]. Im Kontext der rechnergestützten Lehre werden Lernobjekte mit Metadaten beschrieben. Zu den typischen Metadaten gehören beispielsweise Angaben über Autor, Titel oder Zeitpunkt der Veröffentlichung. So beinhalten Metadaten Grundinformationen, die unterstützend bei der Suche, Beurteilung, Akquise und Verwendung der Lerninhalte helfen sollen. In der Regel sind die erfassten Metadaten so beschaffen, dass sie sowohl von Lernern und Dozenten, als auch von automatisierten Rechenprozessen genutzt werden können [LOM02]. In dem sie relevante Informationen über die Lernobjekte zur Verfügung stellen, ermöglichen Metadaten die Organisation und Suche nach Lernobjekten [Wil02]. Da Metadaten für die Suche nach bestehenden Lernobjekten herangezogen werden, hat die Qualität der Metadaten einen entscheidenden Einfluss auf die Wiederverwendung von Lernobjekten.

Hierfür ist die Erfassung und Speicherung der Metadaten nach einem standardisierten Schema erforderlich, damit eine Vergleichbarkeit von Lernobjekten schon anhand der Metadatenbeschreibungen gewährleistet werden kann. Aus diesem Grund wurden in den vergangenen Jahren für verschiedene Anwendungen Metadatenschemata wie Dublin Core (DC) [DC003] der Dublin Core Metadata Initiative [9], Ariadne Metadata [ND03] der Ariadne Foundation [2], IMS Learning Resource Metadata (LRM) [IMS01b] des IMS Global Learning Consortiums (IMS) [14], IEEE Learning Object Metadata (LOM) [LOM02] des Learning Technology Standards Committees (LTSC) [12] und darauf basierenden Subsets, wie CanCore [4] und SingCore [29] entwickelt. Alle diese Metadatenschemata sind bis auf DC, das für Bibliotheken konzipiert wurde, speziell für die rechnergestützte Lehre entwickelt worden. Das LOM-Schema ist bereits 2002 von der IEEE standardisiert worden. Es ist bis auf wenige, kleine Unterschiede identisch mit LRM, dessen XML-Repräsentation Teil der SCORM-Spezifikation und deshalb weit verbreitet ist. Abbildungsvorschriften für die Konvertierung von Metadaten im LOM- beziehungsweise LRM-Schema in andere Metadatenschemata sind definiert. Für Subsets, wie CanCore oder SingCore, werden sogar gar keine Abbildungsvorschriften benötigt. Aufgrund verfügbarer Abbildungsvorschriften zur Konvertierung von Metadaten, der weiteren Verbreitung des LRM XML-Schemas und der Tatsache, dass das LRM XML-Schema Teil der SCORM-Spezifikation ist, wird in der vorliegenden Arbeit Learning Resource Metadata (LRM) des IMS Global Learning Consortiums als Metadatenschema zur Beschreibung der Lernobjekte verwendet.

Noch vor wenigen Jahren wurde das Erstellen von Metadaten als ein einfacher Prozess angesehen [Dow01]. Durch die Verwendung komplexer Metadatenschemata, wie LOM beziehungsweise LRM, ist die Erstellung von Metadaten jedoch keinesfalls als trivial einzustufen. Aus diesem Grund werden Metadaten sehr oft von professionellen Metadatenautoren aus dem Bibliotheksbereich angefertigt. Sie verfügen

über fachliches Wissen und Erfahrungen beim Katalogisieren oder Indexieren von Lernobjekten. Darüberhinaus sind sie mit dem verwendeten Metadatenchema vertraut und somit in der Lage qualitativ hochwertige Metadaten zu erstellen [Wei00]. Die Erstellung von Metadaten von professionellen Indexierern ist jedoch mit hohen Kosten verbunden. Außerdem kann es durch die Einbeziehung professioneller Metadatenautoren zu Verzögerungen bei der Erstellung von Metadaten kommen. Da Metadatenätze jedoch als Grundlage für das Wiederfinden bestehender Lernobjekte dienen, ist die Existenz der Metadaten ein wichtige Voraussetzung für die Wiederverwendung bestehender Lernobjekte.

Alternativ dazu können Metadaten auch direkt von den Autoren der Lernobjekte erstellt werden. Die Autoren sind jedoch in der Regel nicht mit den Konzepten der Metadaten vertraut. Im Vergleich zu professionellen Metadatenautoren besitzen Autoren zwar das Wissen über die den Lernobjekten zu Grunde liegenden Thematiken, haben jedoch in der Regel keine Kenntnisse über die Erstellung von Metadaten. Es ist deswegen davon auszugehen, dass Metadaten, die von den Autoren der Lernobjekte erstellt werden, qualitativ schlechter als Metadaten sind, die von professionellen Metadatenautoren erstellt worden sind [Wei00].

Die automatische Erstellung zumindest eines Teils der Metadaten verspricht dagegen den Aufwand bei der Metadatengenerierung zu vereinfachen und die Erstellung selbst zu vereinheitlichen. Von der automatischen Erstellung von Metadaten können vor allem die Autoren der Lernobjekte profitieren, die im Vergleich zu professionellen Metadatenautoren in der Regel keine Erfahrung und kein Wissen für die Katalogisierung von Lernobjekten besitzen [GCDRL03].

Am National Institute of Environmental Health Sciences [21] wurde eine Studie zum Vergleich der Qualität von automatisch erstellten Metadaten mit der von manuell von Experten erstellten Metadaten auf Basis eines Information-Retrieval-Experiments durchgeführt [Irv03]. Das Ergebnis dieser Untersuchung ist, dass die professionell erstellten Metadaten nicht signifikant besser sind als die automatisch generierten Metadaten. In eine weiteren Studie des Institutes wurde untersucht, ob Autoren geeignet sind, Metadaten zu erstellen [GPPR01]. Das Ergebnis dieser Untersuchung ist, dass Autoren in der Lage sind akzeptable Metadaten zu erstellen die gleichwertig mit den von Experten erstellten Metadaten sind. Ein kritisches Problem bei der dezentralen Erstellung der Metadaten durch die Autoren der Lernobjekte selbst ist, sie zur Erstellung von Metadaten zu motivieren. Hierfür muss das Erstellen von Metadaten möglichst einfach gestaltet sein und belohnt werden [GPPR01].

Die erste Generation von Werkzeugen für die Erstellung von Metadaten stellen einfache Eingabemasken ohne automatische Hilfe bei der Generierung von Metadaten dar. Die Verwendung dieser Metadateneditoren garantiert technisch fehlerfreie Metadaten, ohne dass Metadatenautoren Kenntnisse einer bestimmten Programmiersprache besitzen müssen [GSM⁺03]. Durch die Verwendung von Metadatenvorlagen können diese Metadateneditoren sehr komfortabel genutzt werden. Zu den

wichtigsten Editoren dieser Kategorie von Metadatenerstellungswerkzeugen gehört der Ariadne Metadateneditor [VCVD⁺01, NDS⁺03], der LOM-Editor [Ste02] aus dem MultiBook-Projekt [20] und der Metadateneditor des Reload Editors [24]. Mittlerweile sind Metadateneditoren dieser Kategorie in vielen Produkten zu finden.

Metadatenerstellungswerkzeuge der zweiten Generation unterstützen Autoren bei der Erstellung von Metadaten. Die Palette der verschiedenen Ansätze ist sehr breit. Sie reicht von Werkzeugen für die Textverarbeitung mittels der textbasierte Lernobjekte zusammengefasst, deren Schlüsselworte bestimmt und auf Basis von Begriffen klassifiziert werden können [16], bis hin zu Werkzeugen für die Bearbeitung bereits bestehender Metadaten von Web-Seiten [7, 19].

In Bezug auf die vorliegende Arbeit sind die Arbeiten von Bourda und Cardinaels sehr interessant: In [CMD05] stellen Cardinaels et al. ein Web-Service für die automatische Generierung von Metadaten für Lernobjekte vor. Mit Hilfe des Web-Services können LOM-konforme Metadatenätze generiert werden. Dabei orientiert sich die Metadatengenerierung an die für Ariadne obligatorisch definierten Datenfelder. Bourda et al. [BDK02] beschreiben ein Werkzeug für die semiautomatische Generierung von Metadaten. Es ermöglicht die Metadatengenerierung auf Basis von Metadatenvorlagen, Benutzerprofil und Beziehungen zwischen Lernobjekten. So wird beispielsweise durch Versionsbeziehungen das Versionsdatenfeld modifiziert oder durch Aggregationsbeziehungen das Aggregationsniveau eines Lernobjekts modifiziert.

Die genannten Ansätze zur automatischen Metadatengenerierung stellen in der Regel Insellösungen dar, die selten direkt mit anderen Komponenten oder Arbeitsvorgängen kombiniert sind. Es fehlt die Integration der automatischen Metadatengenerierung in den Autorenprozess, in dem ähnlich wie von Bourda et al. [BDK02] beschrieben, die Metadatengenerierung in Abhängigkeit zu anderen Lernobjekten erstellt werden können ohne diese explizit angeben zu müssen. Durch die Integration der automatischen Metadatengenerierung ist es möglich diese Metadaten allein aus den Aktionen der Benutzer heraus zu generieren.

3.5 Lernobjektarchiv

Voraussetzung für die Wiederverwendung von Lernobjekte ist deren Verfügbarkeit. Solange Lernobjekte über Suchfunktionen gefunden und bei Bedarf wieder abgerufen werden können, sind diese auch wiederverwendbar. Hierfür werden Speicher-, Such- und Abruffunktionen in einem System benötigt.

In der Literatur werden solche Systeme *Learning Object Repositories* (LOR) genannt. Für den Begriff LOR gibt es im Deutschen keine eindeutige Übersetzung. Im Allgemeinen ist damit die Ablage oder ein Aufbewahrungsort zum Speichern von Lernobjekten gemeint, der zugleich als Fundgrube oder Quelle von Lernobjekten dient. Der Begriff *Digitale Bibliothek* wird in der Literatur als digitale Funktions-

einheit definiert, die die Dienste einer traditionellen Bibliothek mit neuen elektronischen Quellen und Methoden verknüpft [Gap93, Mag03]. Im Gegensatz dazu wird ein digitales Archiv als ein System definiert, das lediglich die physikalische Aufbewahrung elektronischer Dokumente sowie Such- und Zugriffsfunktionen zur Verfügung stellt [Bor99]. Im weiteren Verlauf dieser Arbeit wird deswegen als Übersetzung für LORs der Begriff Lernobjektarchiv verwendet.

Ein Lernobjektarchiv speichert sowohl digitale Lernobjekte, als auch ihre beschreibenden Metadaten. Charakterisierend für LORs aber ist, dass Lernressourcen und ihre beschreibenden Metadaten zusammen in einem System verarbeitet werden, unabhängig davon, ob Lernressourcen und Metadaten physikalisch zusammen oder getrennt in verschiedenen Systemen gespeichert werden [ND02]. Zu den am häufigsten von LORs unterstützten Metadatenpezifikationen gehören LOM, IMS Learning Resource Metadata [IMS01b] und Dublin Core [DC003].

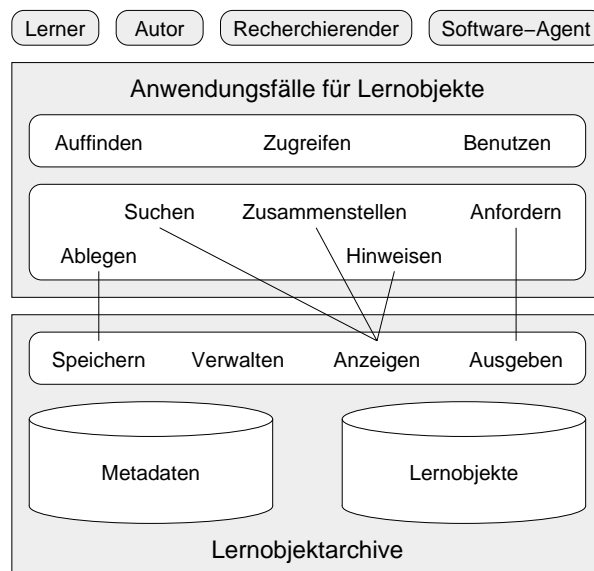


Abbildung 3.3: Funktionen eines Lernobjektarchivs (Quelle [IMS03a])

Nach [IMS03a] muss ein digitales Lernobjektarchiv die in Abbildung 3.3 dargestellten Funktionen bereitstellen. Im unteren Bereich der Abbildung befinden sich die Lernobjektarchive. Basierend auf den Datenbanken für Metadaten und den Lernobjekten bieten sie Grundfunktionen zum Speichern, Verwalten, Anzeigen oder Abrufen von Metadaten und Lernobjekten. Als Anwender, die mit digitalen Lernobjektarchiven arbeiten, werden in [IMS03a] Lerner, Autoren, Recherchierende, sowie Software-Agenten genannt. Zum Wiederverwenden bereits bestehender Lernobjekte sind die Anwendungsfälle Auffinden, Zugreifen und Benutzen vorgesehen. Differenzierter lassen sich die Anwendungsfälle Ablegen, Suchen, Zusammenstellen, Hinweisen und Anfordern auf Basis der Grundfunktionen digitaler Lernobjektar-

chive definieren. Nach Tschritzis [Tsi99] werden in der vorliegenden Arbeit in der Regel Professoren in der Rolle als Autor das Lernobjektarchiv für die Wiederverwendung von Lernobjekten und deren Fragmenten bei der Erstellung neuer Lernobjekte nutzen.

Je nach Funktionsumfang der digitalen Lernobjektarchive, kann mit Hilfe einer erweiterten Suche gezielt nach bestimmten Werten in Datenfeldern der Metadatenätze gesucht werden [ND02]. Klassische Methoden aus dem Information Retrieval, wie das Vektordistanzmodell [BYRN99], können herangezogen werden, um gefundene Treffer mit absteigender Relevanz zu sortieren und um ein zügiges Auffinden der gewünschten Lernobjekte zu ermöglichen. Das Auffinden relevanter Lernobjekte wird zusätzlich durch Techniken der Informationsvisualisierung unterstützt, um den Zugriff auf Lernobjekte zu verbessern [KMT⁺05]. Die Suche in entfernten oder verteilten Lernobjektarchiven wird in [IMS03a] [23] spezifiziert. Beispiele erfolgreicher Lernobjektarchive sind das Ariadne Knowledge Pool System [DFC⁺01], EdNA [10], Merlot [18], SMETE [26], Careo [5] und DSpace [8].

Für die Wiederverwendung von Lernobjekten in einem auf Aggregation basierenden Autorenprozess werden Lernobjektarchive benötigt, die die Ablage und Beschreibung von Lernobjekten in allen Aggregationsniveaus unterstützen. Von den genannten Lernobjektarchiven trifft das nur auf das Ariadne KPS zu. Zusätzlich dazu ist die Versionierung der Lernobjekte erforderlich, damit die Wiederverwendung bereits bestehender Lernobjekte in modifizierter Form erfolgen kann. Da die Metadaten der Lernobjekte durch die Autoren selbst erstellt werden sollen, ist eine möglichst automatische Metadatengenerierung im Lernobjektarchiv erforderlich. Diese Anforderungen werden von keinem der genannten Lernobjektarchive erfüllt [ND02].

3.6 Zusammenfassung

Die Recherche nach verwandten Arbeiten insbesondere nach Autorenwerkzeugen zeigt, dass kein Autorensystem, einen auf Aggregation basierenden Autorenprozess für die Wiederverwendung von Lernobjekten oder deren Fragmenten unterstützt. Von allen vorgestellten Systemen werden viele der Kriterien für die Wiederverwendung von Lernobjekten im Autorenprozess nicht erfüllt. Die meisten betrachteten Systeme unterstützen keine Wiederverwendung von Lernobjekten in allen verfügbaren Aggregationsniveaus. Der Grund hierfür ist die fehlende Adressierbarkeit von Lernobjekten mit niedrigen Aggregationsstufen, wie beispielsweise Medienobjekten. Die Auszeichnung der Lernobjekte aller Aggregationsniveaus mit Metadaten ist ebenfalls bei vielen Systemen nicht erfüllt. Das ist jedoch für das Auffinden und Wiederverwenden bereits bestehender Lernobjekte besonders wichtig. Aufgrund fehlender Software-Werkzeuge für die Bearbeitung aggregierter Lernobjekte aller Aggregationsstufen können Lernobjekte oft nur in unmodifizierter Form

wiederverwendet werden. Doch gerade die Wiederverwendung von Lernobjekten in modifizierter Form erweitert die Möglichkeiten der Wiederverwendung von Lernobjekten in entscheidender Weise. Daher erfolgt in Kapitel 4 die Betrachtung eines abstrakten, auf Aggregation basierenden Autorenprozess.

Für die partielle Wiederverwendung von Lernobjekten ist deren Modularisierung erforderlich. Bei dem Vergleich bestehender Modularisierungskonzepte hat sich die Hierarchie modularer Lernobjekte von Hodgins [Hod02] als geeignetes Konzept herauskristallisiert. Mit fünf Aggregationsniveaus deckt es Lernobjekte mit sehr niedrigen bis hin zu hohen Aggregationsniveaus ab. Es werden jedoch keine rekursiven Aggregationsbeziehungen zwischen den Aggregationsniveaus definiert, so dass eine tiefere Schachtelung eines Kursdokuments nicht durchgeführt werden kann.

Zur Speicherung von Lernobjektfragmenten reicht jedoch ein Modulkonzept allein nicht aus, vielmehr wird ein Dokumentformat benötigt, mit dessen Hilfe Lernobjekte und deren Fragmente modular gespeichert werden können. Für einen auf Aggregation basierenden Autorenprozess ist ein Dokumentformat notwendig, das geeignet ist, Lernobjekte bis hin zu Aggregationsniveaus von Medienobjekten modularisiert zu speichern. Keines der vorgestellten existierenden Dokumentformate unterstützt jedoch die direkte Adressierbarkeit oder die Beschreibung von Lernobjekten auf dem Aggregationsniveau von Medienobjekten mit Metadaten. Trotzdem besitzen einige der genannten Dokumentformate Eigenschaften, die ein geeignetes Dokumentformat für die Wiederverwendung fragmentierter Lernobjekte beim Erstellen von WBTs haben sollten. Dazu gehören die modularisierte Speicherung von Lernobjektfragmenten nach ZYX [BK99], jedoch ohne die Synchronisierungsoperatoren. Die Wiederverwendung von Lernobjekten kann in dem Modell durch Referenzieren ähnlich wie in DITA [Pri01] gelöst werden, um eine skalierbare Wiederverwendung zu erzielen. In Kapitel 4 wird aus diesem Grund ein eigenes Dokumentmodell zur Speicherung modularisierter Lernobjekte entworfen.

Metadatenbeschreibungen von Lernobjekten werden häufig für die Indizierung von Lernobjekten in Datenbanken genutzt. Das Erstellen von Metadaten ist jedoch mit hohem Aufwand verbunden. Damit trotzdem in einem auf Aggregation basierenden Autorenprozess Metadaten für die neu erstellten Lernobjekte und deren Fragmente erstellt werden können, ist die Integration einer weitestgehend automatischen Metadatengenerierung in den Autorenprozess wichtig. Verwandte semiautomatische Metadatenwerkzeuge stellen jedoch in der Regel Insellösungen dar, die bisher noch nicht im Autorenprozess integriert sind. Da die Metadaten parallel zu der Erstellung neuer Lernobjekte generiert werden sollen, wird in Kapitel 5 ein aus verschiedenen Ansätzen kombiniertes Verfahren zur Generierung von Metadaten vorgestellt, das in den auf Aggregation basierten Autorenprozess integriert wird.

Damit die Wiederverwendung bereits bestehender Lernobjekte in modifizierter Form erfolgen kann, ist die Versionierung der Lernobjekte und deren Lernobjektfragmente erforderlich. Nur eines der verwandten Autorensysteme unterstützt die Versionierung von Lernobjekten (L3), diese jedoch nur für Lernobjekte mit middle-

rem bis hohem Aggregationsniveau. Aus diesem Grund wird in Kapitel 6 ein Verfahren für die Versionierung modularisierter Lernobjekte entwickelt, das insbesondere die Aggregationsbeziehungen zwischen den Fragmenten aggregierter Lernobjekte berücksichtigt.

4 Grundlagen eines auf Aggregation basierenden Autorenprozesses

Die Unterstützung der Wiederverwendung von Lernobjekten und deren Fragmenten in einem auf Aggregation basierenden Autorenprozess ist das Ziel der Arbeit. In diesem Kapitel werden die Grundlagen eines auf Aggregation basierenden Autorenprozesses untersucht und ein abstraktes Prozessmodell für einen solchen Autorenprozess vorgestellt. Hierfür wird zunächst eine Analyse der Anforderungen zur Unterstützung der Wiederverwendung von Lernobjekten in einem Autorenprozess durchgeführt. Zunächst werden die grundsätzlichen Voraussetzungen für die Wiederverwendung von Lernobjekten untersucht. Anschließend erfolgt die Analyse der Anforderungen an den Autorenprozess einschließlich der Betrachtung der Wiederverwendung von Lernobjekten im Autorenprozess. Da wie in Abschnitt 3.2 beschrieben kein geeignetes Dokumentmodell für die Speicherung modularisierter beziehungsweise aggregierter Lernobjekte existiert, wird nach der Anforderungsanalyse zunächst ein entsprechendes Dokumentmodell vorgestellt. Zum Schluss erfolgt die Analyse eines auf Aggregation basierenden Autorenprozesses, aus der in zwei Schritten ein abstraktes, erweitertes Prozessmodell hervorgeht.

4.1 Anforderungsanalyse

In diesem Abschnitt erfolgt eine Analyse der Anforderungen an Autorenprozesse, die eine Wiederverwendung und Aggregation von Lernobjekten erlauben. Eingangs werden die grundlegenden Voraussetzungen für die Wiederverwendung von Lernobjekten im Allgemeinen zusammengefasst. In den beiden darauf folgenden Abschnitten werden die Anforderungen bezüglich eines auf Aggregation basierenden Autorenprozesses und der Wiederverwendung von Lernobjekten in dem Autorenprozess untersucht.

4.1.1 Voraussetzungen für die Wiederverwendung von Lernobjekten

In der Literatur bleibt der Fokus gegenwärtiger Arbeiten, die pädagogisch wertvolle Lernmodule als Ergebnis der Aggregation von Inhalten anstreben, häufig auf die Analyse der notwendigen Eigenschaften der zu kombinierenden Lernobjekte beschränkt [CDH⁺04]. Die Aggregation der Lernobjekte wird in diesen Arbeiten nur am Rande betrachtet und nicht detailliert beschrieben. Dies ist leicht verständlich, da man lange Zeit analog zur Lego-Metapher [Wil02] davon ausging, dass die Kombination der Lernobjekte ein trivialer Prozess sei, solange wiederverwendbare Lernobjekte zur Verfügung stehen. Deshalb werden in vielen Arbeiten analog zur Lego-Metapher die notwendigen Eigenschaften dieser Bausteine beschrieben, damit diese wiederverwendbar werden und somit als Grundlage eines auf Aggregation beruhenden Autorenprozesses dienen können, wie in Abschnitt 3.1 beschrieben.

Die grundlegende Voraussetzung für die Aggregation von Lernobjekten in einem Autorenprozess, ist ohne Zweifel deren Existenz. Die bloße Existenz von Lernobjekten reicht jedoch noch nicht aus, um diese in einem Autorenprozess mittels Aggregation zu kombinieren. In [Pol03] werden weitere Basisanforderungen für die Wiederverwendung von Lernobjekten aufgestellt:

Modularisierung Nur wenn eine Modularisierung von Inhalten in diskrete, unabhängige und eigenständige Lernobjekte erfolgt, können sie unter Berücksichtigung inhaltlicher Gesichtspunkte in verschiedenen Kombinationen aggregiert werden, wie in Abschnitt 2.3 beschrieben.

Verfügbarkeit Inhalte, die durch Aggregation kombiniert werden sollen, müssen für die Autoren verfügbar sein. Dabei sind verschiedene Distributionswege möglich. Die Verwendung von Lernobjektarchiven hat sich in den letzten Jahren als Distributionsplattform etabliert.

Auffindbarkeit Für die Verwendung bereits bestehender Lernobjekte ist deren Auffindbarkeit eine grundsätzliche Voraussetzung. Um den Zugriff auf Lernobjekte zu ermöglichen werden deshalb geeignete Suchverfahren benötigt, die auf den Lernobjekten selbst oder deren Metadatenbeschreibungen basieren können.

Im Folgenden werden die über eine reine Wiederverwendung hinausgehenden aggregationspezifischen Anforderungen in Form von Funktionen betrachtet. Diese sind wesentlich komplexer als eine triviale Kombination von digitalen Dokumenten.

4.1.2 Anforderungen an den Autorenprozess

Für einen auf Aggregation basierenden Autorenprozess, in dem Lernobjekte und deren Fragmente zusammen mit neu erstellten Lernobjekten wiederverwendet werden

sollen, ergeben sich Anforderungen, die verschiedene Teilbereiche des Autorenprozesses abdecken.

Aggregation von Lernobjekten Der wesentliche Bestandteil eines auf Aggregation basierenden Autorenprozesses ist ein Verfahren zur Kombination modularisierter Lernobjektfragmente. Hierfür ist es erforderlich, eine möglichst einfache Methode zur Strukturierung der modularisierten Lernobjekte zu finden, mittels der die Zugehörigkeit von Lernobjektfragmenten zu Lernobjekten eines höheren Aggregationsniveaus und die Struktur der einzelnen zu aggregierenden Lernobjektfragmente definiert werden kann. Diese Strukturierung kann durch Operationen wie beispielsweise Verschieben, Sortieren und Zuordnen erfolgen.

Aggregation aller Aggregationsniveaus Eine weitere Anforderung an den Autorenprozess ist die Aggregation von Inhalten über alle drei Bereiche des Granularitäts- und Aggregationsspektrums hinweg, das in [SM02] aufgespannt wird. Damit wird erreicht, dass Inhalte sowohl auf instruktionaler sowie auf Ebene des Mediendesigns wiederverwendet werden können. Hierfür wird ein Aggregationsmodell benötigt mit dessen Hilfe Inhalte über den gesamten Bereich, den das Spektrum abdeckt, aggregiert werden können.

Sichtbarkeit der Lernobjektfragmente Damit die Lernobjektfragmente von den Autoren bequem strukturiert beziehungsweise geordnet werden können, ist es weiterhin erforderlich, dass sie einen Überblick über den Inhalt der einzelnen Lernobjektfragmente besitzen. Dieser kann bei einer Darstellung der Lernobjektfragmente durch Platzhalter schnell verloren gehen. Dabei werden die einzelnen Lernobjekte und deren Fragmente, wie in [HFM⁺01] vorgeschlagen, in der Autorenumgebung nicht direkt angezeigt, sondern durch Platzhalter ersetzt. Abbildung 4.1 zeigt eine Autorenumgebung, in der statt den Lernobjekten Platzhalter angezeigt werden, die sich aus einem Piktogramm und dem Titel des Lernobjekts zusammensetzen. Da sich auf Basis dieser Umsetzung die Inhalte des abgebildeten Kurses höchstens erahnen lassen, ist es wichtig, dass die Lernobjekte während des Autorenprozesses angezeigt werden. Das gilt insbesondere für die Strukturierung von Lernobjekten mit niedrigen Aggregationsniveaus, wie beispielsweise Medienobjekte. Der in [HFM⁺01] beschriebene Ansatz stellt deshalb, auch in Kombination mit einer Vorschaufunktion, keine befriedigende Lösung für die Strukturierung von Lernobjektfragmenten dar.

Um eine Sichtbarkeit der Inhalte während des Autorenprozesses zur Strukturierung der Lernobjektfragmente zu gewährleisten ist eine Darstellung der Lernobjekte in der Autorenumgebung notwendig, die exakt mit der Darstellung bei dem Export der Kurse übereinstimmt. Diese Darstellungsart von

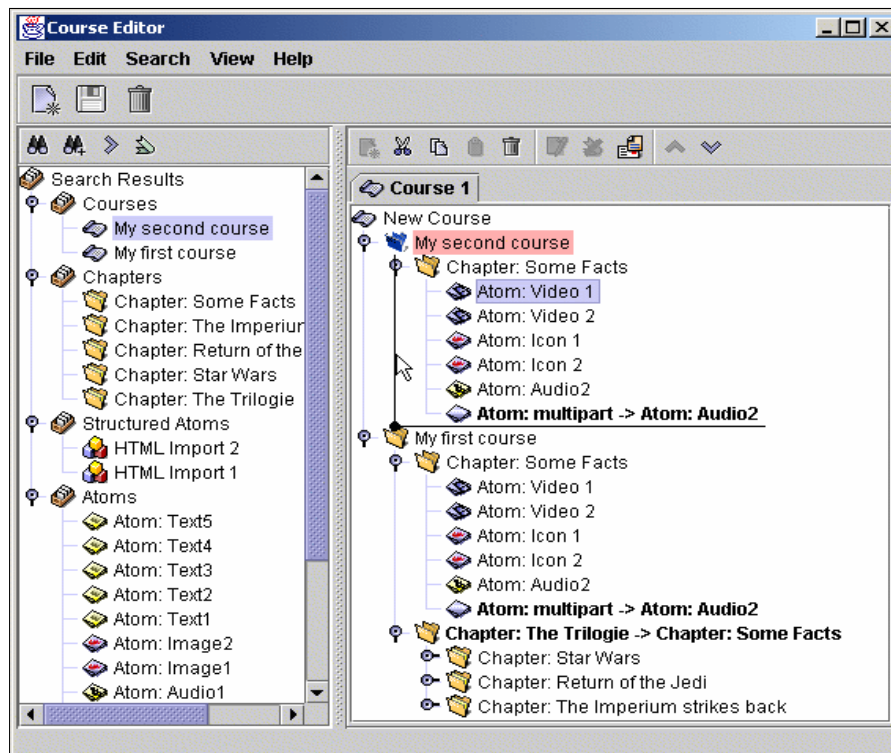


Abbildung 4.1: Visualisierung aggregierter Lernobjekte mit Platzhaltern in einer Autorenumgebung (Quelle: [HFM⁺01])

Inhalten in Editoren wird „What You See Is What You Get“ (WYSIWYG) genannt. Diese Art der Darstellung ist jedoch bei einem auf Aggregation basierenden Autorenprozess aufgrund verschiedener Ursachen nicht immer herzustellen. Grund hierfür ist die schlechte Handhabbarkeit von Lernobjekten mit höheren Aggregationsniveaus bei der Strukturierung von Lernobjektfragmenten in einer WYSIWYG-basierten Darstellung. Der Platz, der für die Darstellung dieser Lernobjekte benötigt wird, behindert dabei ihre Strukturierung. Hinzu kommt, dass bei der Trennung von Inhalt und Layout und der Verwendung mehrerer Layoutvorlagen eine WYSIWYG-Darstellung der Lernobjekte ohnehin nicht zu realisieren ist, weil das Layout ein wichtiger Bestandteil für die Bearbeitung in einer WYSIWYG-Umgebung darstellt. Aus diesen Gründen ist eine WYSIWYG-Umgebung für die Strukturierung von Lernobjekten nicht geeignet. Stattdessen bietet sich hierfür eine rein inhaltsbezogene Darstellung der Lernobjektfragmente ohne endgültiges Layout an.

Anpassbarkeit der Lernobjektfragmente Nach [BA04] und in dieser Arbeit werden in einem auf Aggregation basierenden Autorenprozess sowohl bereits bestehende als auch neu erstellte Inhalte aggregiert. In [DH03] und in dieser

Arbeit kann dabei eine Anpassung bestehender Lernobjekte erfolgen. Für den Autorenprozess ergibt sich daraus die Anforderung, dass Lernobjektfragmente für eine Wiederverwendung angepasst und komplett neu erstellt werden können. Damit sind Funktionen notwendig, mittels derer die Lernobjektfragmente inklusive ihrer Metadatenbeschreibungen bearbeitet beziehungsweise angepasst werden können. In einem Anwendungsszenario, in dem viele verschiedene Autoren Inhalte aggregieren und wiederverwenden sollen, dürfen diese Funktionen nicht auf eigene Lernobjekte beschränkt sein, sondern auch für Lernobjekte fremder Autoren ausführbar sein.

Versionierung der Lernobjekte Für die Wiederverwendung anpassbarer Lernobjekte muss der Autorenprozess eine Versionierung der Lernobjekte und ihrer Fragmente unterstützen. Mit Hilfe dieser in den Autorenprozess integrierten Versionierung der Lernobjekte wird die Modifikation von Lernobjektfragmenten möglich, ohne dass dadurch andere aggregierte Lernobjekte verändert werden, die die veränderten Lernobjektfragmente referenzieren. Ein Beispiel hierfür ist die Verwendung eines Lernobjektfragments eines anderen Autors als Teil eines eigenen Lernobjekts. Mit Hilfe der Versionierung soll die Veränderung eines von einem fremden Autoren wiederverwendeten Lernobjektfragments keine Auswirkung auf das eigene Lernobjekt haben, das das geänderte Lernobjektfragment verwendet.

Eine interessante Anwendung, die auf der Versionierung von Lernobjekten basiert, ist die Handhabung korrigierter Lernobjekte. Besonders interessant ist dabei zu entscheiden, ob Korrekturen in aggregierten Lernobjekten angenommen werden oder nicht. Für beide Fälle gibt es plausible Anwendungen. Zur Unterstützung dieser Anwendungsfälle sind zusätzliche Mechanismen sowohl für die Signalisierung als auch für die automatische Annahme von Korrekturen notwendig.

Recherche nach Lernobjektfragmenten Als eine grundlegende Voraussetzung für eine Wiederverwendung von Lernobjekten wurden in 4.1.1 bereits Funktionen zur Suche und zum Zugriff auf ein Lernobjektarchiv aufgeführt. Wird die Suche in den Autorenprozess integriert, kann sie durch Anpassung der Suche nach geeigneten Lernobjektfragmenten in Abhängigkeit des Kontextes verbessert werden. Hierfür können beispielsweise bei der Suche nach geeigneten Lernobjektfragmenten diejenigen ausgeblendet werden, die aufgrund des Aggregationsniveaus für die Aggregation nicht geeignet sind.

Verschiedene Erstellungsverfahren Ein Autorenprozess, der auf Aggregation basiert, kann das Arbeiten sowohl nach dem Top-Down- als auch dem Bottom-Up-Verfahren gleichermaßen gut unterstützen. Im Top-Down-Verfahren erstellt der Autor zunächst eine Struktur und füllt die einzelnen Strukturele-

mente später mit bestehenden oder neu erstellten Lernobjekten. Im Bottom-Up-Verfahren erfolgt die Erstellung in umgekehrter Reihenfolge. Zunächst werden einzelne Medienobjekte erstellt, die anschließend mit Hilfe der Strukturelemente aggregiert werden. Beide Ansätze zum Erstellen von Inhalten müssen unterstützt werden. Für die Erstellung von Inhalten nach dem Top-Down-Verfahren ist es erforderlich, dass die Autorenumgebung vom höchsten Aggregationsniveau ausgehend das Erstellen und Zusammenfassen von zunächst leeren Strukturelementen erlaubt, die Platzhalter für später zu erstellende beziehungsweise aggregierende Inhalte darstellen. Außerdem ist es erforderlich, dass diese leeren Platzhalterelemente bereits aussagekräftige Bezeichnungen tragen können und bei Bedarf bereits teilweise mit Inhalten gefüllt werden können, um das Erstellen von Inhalten vom Allgemeinen bis hin zu Details zu unterstützen. Bei dem Bottom-Up-Verfahren wird die Erstellung von Inhalten mit Details begonnen. Hierfür ist es erforderlich, dass im Autorenprozess die Erstellung von Lernobjekten kleiner Aggregationsniveaus unterstützt wird, ohne dass diese von vorneherein eine Aggregationsbeziehung zu übergeordneten Lernobjekten besitzen.

Speicherformat für aggregierte Lernobjekte Für das Zielformat zur Speicherung der in einem auf Aggregation basierenden Autorenprozess erstellten Lernobjekte sind zwei verschiedene Ansätze möglich, die für verschiedene Anwendungsfälle optimiert sind. Das wichtigste Zielformat stellt dabei die modularisierte Speicherung der aggregierten Lernobjektfragmente dar. Denn anders als bei verbreiteten Textverarbeitungsanwendungen ist es nicht sinnvoll die aggregierten Lernobjekte nach ihrer Aggregation als ein monolithisches Objekt beziehungsweise Dokument zu speichern, sondern die einzelnen Fragmente auch nach der Aggregation als solche bestehen zu lassen [CD03]. Sie stehen dann sowohl als einzelne Lernobjektfragmente als auch als ein einzelnes aggregiertes Lernobjekt für die weitere Aggregation bereit. Datenformate dieser Art sind in der Regel anwendungsspezifisch und auf die Verarbeitung modularisierter Lernobjekte zugeschnitten. Sie eignen sich dementsprechend nicht für den Austausch von Lernobjekten, etwa zur Übertragung der erstellten Lernobjekte von einer auf Aggregation basierenden Autorenumgebung zu einem Lernmanagementsystem, in dem die Lernobjekte den Lernenden dargestellt werden. Für diese Übertragung der Lernobjekte müssen andere Datenformate verwendet werden, die zwar weiter verbreitet sind, aber in der Regel nicht die Aggregation in einem Autorenprozess unterstützen. Ein weit verbreitetes Datenformat hierfür ist SCORM, das bereits in Abschnitt 3.2 beschrieben wurde.

Für die Unterstützung der Wiederverwendung von Lernobjekten im Autorenprozess ergeben sich weitere Anforderungen, die im folgenden Abschnitt erläutert werden.

4.1.3 Wiederverwendung von Lernobjekten im Autorenprozess

Die in Abschnitt 4.1.1 genannten Voraussetzungen für eine Wiederverwendung von Lernobjekten müssen auch in einem auf Aggregation basierenden Autorenprozess für eine wiederkehrende Wiederverwendung neu erstellter Lernobjekte gewährleistet sein. Das heißt, die im Autorenprozess erstellten oder aggregierten Lernobjekte müssen modularisierbar, verfügbar und auffindbar sein. Dabei soll dies für die Autoren möglichst transparent¹ und mit geringem Aufwand erfolgen, da die Autoren in der Regel nicht bereit sein werden einen Zusatzaufwand zu investieren, um ihre Lernobjekte wiederverwendbar zu machen. Der Grund für diese mangelnde Bereitschaft liegt darin, dass das Ziel der Autoren mit der fertiggestellten Aggregation von Lernobjekten zunächst erfüllt ist: Die Lernobjekte sind zusammengestellt. Weitere Aktivitäten führen zu keiner Verbesserung der Lernobjekte.

Modularisierung der Lernobjekte Die Modularisierung aggregierter Inhalte stellt solange kein Problem dar, solange die Inhalte nach ihrer Zusammenstellung nicht zu einem monolithischen Dokument kombiniert werden. Für den Autorenprozess bedeutet das, dass einerseits das Arbeiten mit modularisierten Inhalten möglichst transparent gestaltet werden muss und andererseits nach der Fertigstellung der Aggregation von Lernobjektfragmenten die neu zusammengestellten Lernobjekte weiterhin als modulare und modularisierbare Inhalte bestehen bleiben, ohne dabei die Aggregation der Lernobjektfragmente unabänderlich durch Kombination in einem Dokument zu manifestieren.

Modularisierte Speicherung der Lernobjekte Für das Dokumentformat, das die Basis der Modularisierung darstellt, ist dementsprechend eine Struktur erforderlich, die eine transparente Modularisierung im Sinne von für den Benutzer unsichtbarer Modularisierung gestattet. Hierfür ist es erforderlich, dass das Dokumentformat besonders Dokumentfragmente unterstützt, die als Container für eigenständige Inhalte verwendet werden können, so dass trotz der transparenten Modularisierung der Inhalte für die Benutzer nachvollziehbar bleibt, nach welchen Regeln die Inhalte modularisiert werden, damit zusammengehörige Inhalte stets zusammenbleiben und so eigenständige Inhalte bilden, so dass diese wiederverwendet werden können. Die Modularisierung der Inhalte kann dabei durch geeignete Ableitung des Autorenprozesses in Abhängigkeit des gewählten Dokumentmodells erfolgen.

¹Transparente Funktionen im Sinne der Informatik werden als nicht direkt erkennbar definiert. In der vorliegenden Arbeit wird der Begriff der transparenten Funktion für solche verwendet, die für Benutzer weitestgehend automatisiert und intuitiv zu verwenden sind, so dass transparente Funktionen oft ohne das Wissen der Benutzer zur Anwendung kommen können.

Metadatengenerierung Das Erstellen von Metadaten stellt einen nicht zu unterschätzenden Aufwand dar. Aus diesem Grund ist es erforderlich den Autoren Hilfestellung bei ihrer Erstellung zu leisten. Hierfür ist die Weiterentwicklung der Werkzeuge zur Erstellung der Metadaten erforderlich, um Autoren durch eine weitestgehend automatische Metadatenerstellung zu entlasten. Damit die Autoren auch wirklich Metadaten erfassen, ist es erforderlich die Funktionen zur Metadatenerstellung in den Autorenprozess zu integrieren. Das erlaubt es weiterhin Kontextinformationen aus dem Autorenprozess für die Erstellung von Metadaten zu verwenden.

Basierend auf den gesammelten Anforderungen erfolgt zunächst die Beschreibung eines Dokumentmodells, das für die Speicherung modularisierter Lernobjekte geeignet ist und die Grundlage für einen auf Aggregation basierenden Autorenprozess darstellt. Danach erfolgt die Modellierung eines abstrakten, auf Aggregation basierenden Autorenprozesses.

4.2 Dokumentmodell für Kurse

In diesem Abschnitt erfolgt die Beschreibung des Dokumentmodells zur Speicherung von Lernobjekten, die in einem auf Aggregation basierenden Autorenprozess erstellt werden. Dabei erfüllt das im Rahmen dieser Arbeit entworfene Dokumentmodell die Anforderungen, die in den Abschnitten 4.1.2 und 4.1.3 aufgestellt wurden. Im ersten Abschnitt erfolgt zunächst eine Abgrenzung zu verwandten Dokumentmodellen. Für die Beschreibung des Dokumentmodells wird dann die Beschaffenheit der einzelnen Elemente, auch Knoten genannt, beschrieben. Danach wird die Aggregation von Lernobjekten im Dokumentmodell eingeführt. Für dieses wird anschließend ein Schema zur Klassifikation der Dokumentknoten beschrieben. Zum Schluss erfolgt eine Betrachtung zur Beschreibung aggregierter Lernobjekte.

4.2.1 Abgrenzung zu verwandten Dokumentmodellen

Zur Speicherung von Lernobjekten, die in einem auf Aggregation basierenden Autorenprozess erstellt werden, wird ein Dokumentmodell benötigt, das geeignet ist, die aggregierten Lernobjekte in modularisierter Form abzulegen. Damit soll erreicht werden, dass die aggregierten Lernobjekte nicht als monolithisches Objekt beziehungsweise Dokument gespeichert werden, so dass deren zusammengestellten Fragmente auch nach der Aggregation als solche bestehen und für eine eventuelle Wiederverwendung verfügbar bleiben.

Keines der verwandten Dokumentmodelle, die in Abschnitt 3.2 besprochen wurden, erfüllt jedoch alle Anforderungen und Voraussetzungen zugleich, die in den

Abschnitten 4.1.2 und 4.1.3 bezüglich geeigneter Dokumentmodelle als Basis genannt wurden. Aus diesem Grund wurde im Rahmen dieser Arbeit ein eigenes Dokumentmodell entwickelt, das die entsprechenden Anforderungen erfüllt.

Das entwickelte Dokumentmodell definiert die Aggregation von Lernobjektfragmenten verschiedener Aggregationsniveaus ohne jedoch eine Beschreibung der Inhalte festzulegen. Das bedeutet das in dem Dokumentmodell die Lernobjekte aller Aggregationsniveaus als „Blackbox“, das heißt als beliebiges binäres Objekt, betrachtet werden. Die Aggregationsbeziehungen zwischen den Lernobjekten werden hierfür unabhängig von deren Inhalt und deren Datenformat in den Metadaten gespeichert. Das erleichtert die Verwaltung der Lernobjekte, weil zum Feststellen der Aggregationsbeziehungen zwischen den Lernobjekten statt der eigentlichen Daten nur die Metadaten der Lernobjekte gelesen und interpretiert werden müssen.

Die Aggregation beziehungsweise Modularisierung ist ähnlich zu dem Ansatz des ZYX-Modells, ohne jedoch die Operatoren zur Beschreibung temporaler Beziehungen der Lernobjektfragmente zu verwenden, da diese für die Beschreibung von WBTs nicht benötigt werden. Außerdem entfällt eine Beschreibung zur expliziten sowie absoluten räumlichen Positionierung der aggregierten Lernobjektfragmente, weil die Positionierung der Lernobjektfragmente automatisch in Abhängigkeit der gewählten Vorlage für die Ausgabe erfolgen soll. Wiederverwendete Lernobjekte beziehungsweise -fragmente werden ähnlich zu DITA referenziert, jedoch mit dem Unterschied, dass Lernobjektfragmente bis hinunter zu Medienobjekten durch Referenz wiederverwendet werden können. Für die Knoten des Dokumentmodells ist eine Versionierung vorgesehen. Ziel der Versionierung ist die Unterstützung der Wiederverwendung aggregierter Lernobjekte in modifizierter Form. Ansätze zur Versionierung und Replizierung von Lernobjekten für ein verteiltes Autorensystem, wie sie im Projekt L3 entwickelt wurden, werden nicht betrachtet. Die Versionierung von Lernobjekten wird in Kapitel 6 behandelt. Konzepte aus dem Cross-Media-Publishing fließen nicht in das Dokumentmodell ein, weil das Dokumentmodell die Lernobjekte als beliebige digitale Objekte betrachtet, wohingegen Ansätze aus dem Cross-Media-Publishing stets ein Datenmodell der Lernobjekte selbst beschreiben.

In den nächsten Abschnitten wird das im Rahmen dieser Arbeit entwickelte Dokumentmodell erläutert, das in [HS02b, HSDK⁺03b] bereits teilweise veröffentlicht wurde.

4.2.2 Knoten im Dokumentmodell

Ein Lernobjekt wird durch einen Knoten im Dokumentmodell repräsentiert. Die Knoten des Dokumentmodells setzen sich aus den Inhalten des Lernobjekts und deren Metadatenbeschreibung zusammen. Abbildung 4.2 zeigt die Zusammensetzung eines Dokumentknotens. Er setzt sich aus jeweils genau einem Objekt der Klasse 'Lernobjekt' und einem Objekt der Klasse 'Metadaten' zusammen. Dabei beinhaltet das Objekt der Klasse Lernobjekt die physikalischen Daten der modularisierten

Lernobjekte, sowie einen Identifikator zur Referenzierung der physikalischen Daten. Das Objekt der Klasse 'Metadaten' enthält eine Metadatenbeschreibung der physikalischen Daten und trägt ebenfalls einen Identifikator, damit der Metadatenatz referenziert werden kann. In der Metadatenbeschreibung werden die physikalischen Daten des Objekts der Klasse 'Lernobjekt' über dessen Identifikator referenziert, so dass auf die physikalischen Daten zugegriffen werden kann, die durch einen Metadatenatz beschrieben werden.

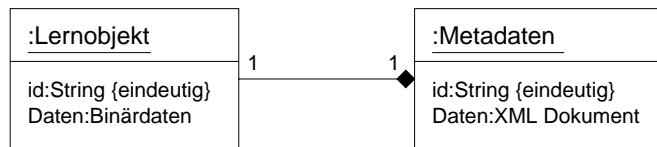


Abbildung 4.2: Zusammensetzung eines Dokumentknotens

Zur Referenzierung eines Dokumentknotens reicht es demnach aus, eine Referenz auf dessen Metadatenbeschreibung zu haben. Über die Metadatenbeschreibung kann dann mit Hilfe der Referenz auf die physikalischen Daten des Dokumentknotens zugegriffen werden.

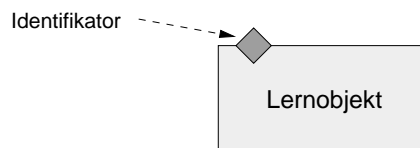


Abbildung 4.3: Visualisierung eines Knotens im Dokumentmodell

Für die Visualisierung von Dokumentknoten wird im Folgenden eine vereinfachte Darstellung gewählt, wie in Abbildung 4.3 gezeigt. Das graue Rechteck umfasst die physikalischen Daten und den korrespondierenden Metadatenatz des Lernobjekts. Die dunkelgrau gefärbte Raute stellt einen Identifikator dar, mit dessen Hilfe Dokumentknoten referenziert werden können. Die Raute symbolisiert gewissermaßen den Identifikator der Metadatenbeschreibung.

4.2.3 Aggregation von Lernobjekten im Dokumentmodell

Die im vorherigen Abschnitt eingeführten Dokumentknoten stellen die Bausteine eines Dokuments dar. Die für den Aufbau eines aggregierten Dokumentes erforderlichen Beziehungen zwischen den einzelnen Dokumentknoten werden in diesem Abschnitt eingeführt.

Zur Aggregation von Inhalten wird die hierarchische Beziehung „HasPart“ und die Umkehrbeziehung „IsPartOf“ verwendet. Mit Hilfe dieser Beziehungen können hierarchische Kursdokumente aufgebaut werden, deren Wurzelknoten und innere

Knoten die Inhalte der Blätter der Hierarchie des Dokumentmodells aggregieren. Hierfür werden alle Teile des Kursdokuments, aus denen sich ein Knoten des Dokuments zusammensetzt mit Hilfe der „HasPart“ Beziehung referenziert.



Abbildung 4.4: Rekursive Aggregation von Objekten der Klasse „Lernobjekt“ im Dokumentmodell

Abbildung 4.4 zeigt die rekursive Aggregation von Knoten eines Dokuments in UML-Notation. Danach können Instanzen der Klasse Dokumentknoten beliebig viele Beziehungen vom Typ „HasPart“ zu anderen Instanzen dieser Klasse haben. Mit Hilfe dieser rekursiven Beziehung lassen sich beliebige gerichtete Graphen aufbauen. Aufgrund der Bedeutung der „HasPart“-Beziehung sind gerichtete Graphen mit Zyklen dieser Beziehungen ungültig.

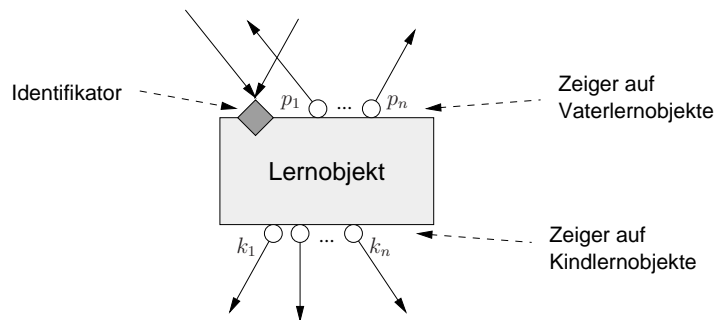


Abbildung 4.5: Visualisierung eines Knotens im Dokumentmodell

Für dieses rekursiv definierte Dokumentmodell wird die Visualisierung aus Abbildung 4.3 erweitert. Wie Abbildung 4.5 zeigt, wird jeder Dokumentknoten mit Zeigern auf die Vater- und Kindknoten im Kursdokument versehen. Hierfür werden die Zeiger p und k eingeführt, die die Vater- beziehungsweise Kindknoten referenzieren. Mit Hilfe der Zeiger k werden alle Dokumentknoten des Kursdokuments referenziert aus denen sich der entsprechende Kindknoten zusammensetzt. Sie sind vom Typ „HasPart“. Die Umkehrbeziehung ist vom Typ „IsPartOf“. Sie werden mit Hilfe der Zeiger p gespeichert. Mit Hilfe der Zeiger p kann ausgehend von einem Lernobjekt, das in Kursen verwendet wird, festgestellt werden, in welchen Kursen es verwendet wird.

Die Zeiger v und k eines Knotens im Kursdokument werden in den Metadaten eines Lernobjekts gespeichert, so dass beispielsweise ohne Analysieren der physikalischen Daten des Lernobjekts ihre Verwendung in Kursen festgestellt werden kann.

LOM bietet für diesen Zweck die Kategorie „Relation“ an. Für jede Beziehung eines Lernobjekts zu einer anderen wird ein Eintrag der Kategorie „Relation“ angelegt. Für jede der Beziehungen wird der Typ und das Ziel zu dem Lernobjekt, welches das Ziel der Referenz ist, gespeichert.

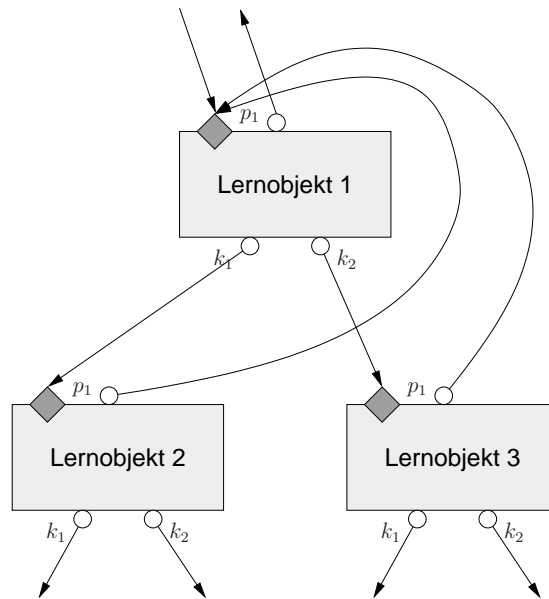


Abbildung 4.6: Verkettung von Knoten des Dokumentmodells

Den Aufbau einer hierarchischen Kursdokumentstruktur mit Hilfe der Zeiger p und k zeigt Abbildung 4.6. Lernobjekt 1 stellt eine Aggregation von Lernobjekt 2 und Lernobjekt 3 dar. Zu diesem Zweck werden die Lernobjekte 2 und 3 unter Verwendung der Zeiger k_1 und k_2 referenziert. Die Lernobjekte 2 und 3 referenzieren jeweils Lernobjekt 1 über deren Zeiger p_1 , weil sie Teil dieses Lernobjekts sind. Lernobjekt 1 ist selbst Teil eines anderen Lernobjekts. Aus diesem Grund referenziert es das Vaterlernobjekt mit Hilfe des Zeigers p_1 . Auch die Lernobjekte 2 und 3 aggregieren weitere Lernobjekte, so dass sie ebenfalls k -Zeiger besitzen.

Durch Verknüpfen von Lernobjekten mit Hilfe der „HasPart“-Beziehungen entstehen Kursdokumente im Lernobjektarchiv. Die Kursdokumente stellen gerichtete Graphen dar, deren Wurzelknoten und innere Knoten Lernobjekte des Lernobjektarchivs aggregieren. Sowohl die Wurzel des Kursdokuments als auch die inneren Knoten des Kursdokuments stellen Lernobjekte dar, die ebenfalls im Lernobjektarchiv abgelegt werden können. Knoten im Kursdokument, die keine untergeordneten Knoten referenzieren, stellen die Blätter der Kursdokumente dar, die Rohdaten enthalten. Sie aggregieren keinen weiteren Inhalte. Aufgrund dessen besitzen sie keine Zeiger k .

4.2.4 Klassifizierung von Dokumentknoten

Im vorherigen Abschnitt ist die Aggregation von Inhalten beschrieben worden, mit dessen Hilfe Kursdokumente auf Basis bestehender im Lernobjektarchiv gespeicherter Lernobjekte aggregiert werden können. Bei der Aggregation von Inhalten unter Verwendung der „HasPart“-Beziehung sind zunächst keine Randbedingungen über die Bedeutung der beteiligten Knoten oder den Grund der Aggregation notwendig. Für die Erstellung von Kursdokumenten basierend auf Aggregation ist es jedoch erforderlich durch ein Klassifikationsschema eine zusätzliche Semantik der Dokumentknoten einzuführen, und die möglichen Fälle, Lernobjekte zu aggregieren, einzuschränken. Schließlich ist es nicht sinnvoll, als Teil eines Abschnitts einen Kurs über eine „HasPart“-Beziehung zu referenzieren.

Das Schema zur Klassifikation von Lernobjekten, die Teil einer Kursstruktur sind und das in diesem Abschnitt eingeführt werden soll, orientiert sich an den folgenden drei Eigenschaften:

Aufbau der Dokumentstruktur Die Anzahl und Eigenschaften der Klassen, denen Knoten der Kursstruktur zugeordnet werden können, ist in hohem Maße abhängig von dem gewünschten Aufbau der Kursdokumente. Ein Beispiel hierfür ist die Gliederung und die Gliederungstiefe eines Kurses. Kurse, für die eine flache bis hin zu wenig oder gar keiner Gliederung definiert ist, benötigen andere Klassen als Kurse für die eine sehr tiefe Gliederung beschrieben wird.

Wiederverwendung Die Definition der Einheiten beziehungsweise Lernobjektfragmente eines Kurses, die wiederverwendet werden sollen, bestimmen ebenfalls stark die Anzahl der Klassen und deren Eigenschaften. Alle Lernobjektfragmente eines Kurses, die für die Wiederverwendung verfügbar gemacht werden sollen, müssen als eigenständige, modularisierte Lernobjekte, das heißt Knoten des Kursmodells, wieder zu finden sein, damit bei der partiellen Wiederverwendung von Lernobjekten genau ein Knoten eines aggregierten Lernobjekts referenziert werden kann. Das ergibt sich aus den allgemeinen Forderungen zur Verbesserung der Wiederverwendung [Pol03]. Demnach können modularisierte Lerninhalte, die jeweils für sich mit Metadaten beschrieben sind, durch leichtes Auffinden besser wiederverwendet werden, wie in 4.1.1 beschrieben wurde.

Ausgabeformat Die gewünschten Ausgabeformate prägen ebenfalls die Entwicklung des Schemas zur Klassifikation der Dokumentknoten eines Kurses. Die Schwierigkeit dabei liegt darin, dass es jeweils für jedes Format eine Abbildung der Kursdokumentknoten geben muss, nach dem die einzelnen Knoten des Kursdokuments den entsprechenden Komponenten des Ausgabeformats zugeordnet werden können. Mit steigender Anzahl von Ausgabeformaten steigt

damit auch zunehmend der Aufwand für die Erstellung des Klassifikationsschemas und der Entwicklung der Abbildungsvorschriften. Das Klassifikationsschema stellt dabei eine abstrakte Menge von Klassen zur Klassifikation von Dokumentknoten dar, die sich für jedes Ausgabeformat abbilden lassen.

In Abschnitt 3.1 wurden verschiedene Klassifikationsschemata für Lernobjekte miteinander verglichen. Das Ergebnis der Untersuchung ist, dass das Schema von Duval und Hodgins am besten für die Klassifikation von Lernobjekten wie Kursen und deren Lernobjektfragmente geeignet ist. Hodgins beschreibt in [Hod02] keine Rekursion bei der Aggregation des vorgeschlagenen Klassifikationsschemas. Deshalb eignet sich das Schema nicht für die Klassifikation der Fragmente von Lernobjekten mit beliebig tiefen Hierarchien. Zusätzlich verschwimmen aufgrund der feinen Modellierung von Informationsblöcken, Lernobjekten und Lektionen, die Grenzen zwischen diesen Lernobjektklassen, so dass die intuitive Verwendung und eine transparente Modularisierung der Lernobjekte nach diesem Schema nicht gewährleistet werden kann. Aus diesen Gründen wird in der vorliegenden Arbeit das Klassifikationsschema von Duval und Hodgins mit den folgenden Modifikationen übernommen. Die Klasse Lernobjekte entfällt. Lernobjekte der Klasse Lektion setzen sich deswegen nicht mehr aus Lernobjekten, sondern aus Informationsobjekten zusammen. Zum Aufbau von beliebig tiefen Hierarchien, wird eine Rekursion in die Aggregation von Lernobjekten der Klasse Lektion aufgenommen. Mit Hilfe dieser Rekursion können Lektionen untergeordnete Lektionen enthalten. Auf Basis dieser Rekursion können Hierarchien von Lektionen gebildet werden.

Dieses Modell erlaubt die Aggregation beziehungsweise Modularisierung von Kursen mit beliebiger Gliederungstiefe. Die Wiederverwendung von Kursen oder deren Fragmenten erfolgt auf Basis der definierten Lernobjektklassen. Durch die Definition der vier Lernobjektklassen können Lernobjektfragmente mit drei Aggregationsniveaus wiederverwendet werden. Aufgrund der Rekursion im Aggregationschema können Lektionen mit verschiedener Komplexität erstellt und wiederverwendet werden.

Als Zielformat zur Ausgabe der Kurse bietet sich SCORM [Adv01] an. Dieses Datenformat hat sich in der rechnergestützten Lehre für den Austausch von WBTs zwischen Inhaltsersteller und den Betreibern von Lernplattformen etabliert. Die Spezifikation von SCORM beschreibt neben Metadaten für Lernobjekte und einer Schnittstelle zum Austausch von Punkteständen zwischen Lerneinheit und Lernplattform ein primitives Modell zur Aggregation von Inhalten. Die Abbildung von aggregierten Kursen in das Aggregationsmodell ist deswegen problemlos möglich. Hierfür ist es erforderlich die Lektionen und deren untergeordnete Lernobjektfragmente in webbasierte Dokumente zu konvertieren. Die Aggregation der Inhalte wird in einem Manifest beschrieben. In [HSGS03] wird die Erstellung SCORM-basierter WBTs ausführlich beschrieben.

Abbildung 4.7 zeigt das resultierende Klassifikationsschema im Aggregationsdia-

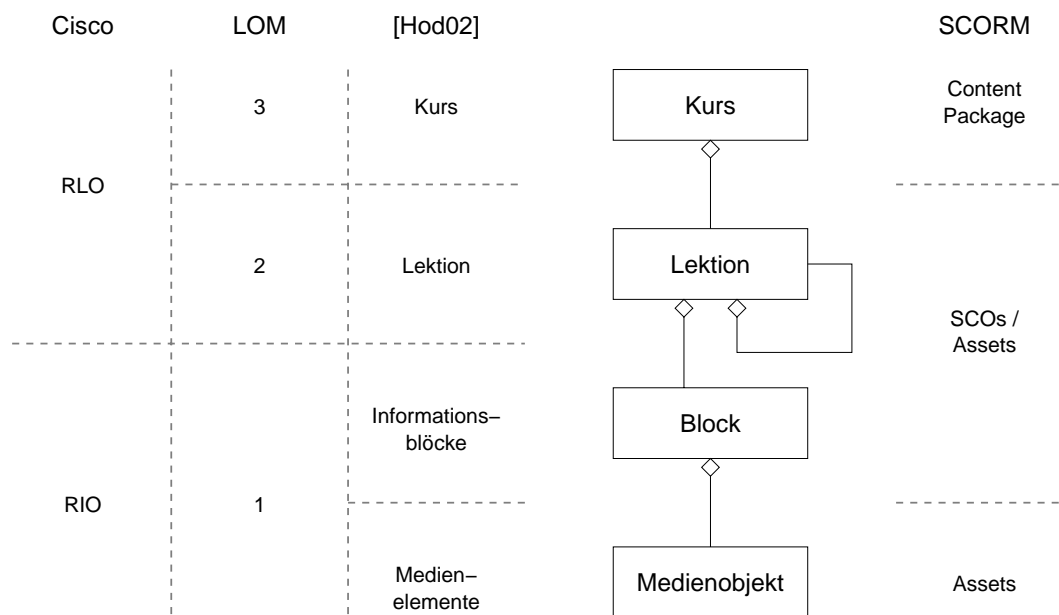


Abbildung 4.7: Lernobjektklassen und ihre Aggregation im Dokumentmodell

gramm. Links neben dem Diagramm befinden sich zum Vergleich die diskutierten Schemata aus Abschnitt 3.1. Da es im Klassifikationsschema die Klasse Lernobjekt nicht mehr gibt, befindet sich auch kein entsprechender Eintrag in der Spalte des Modells von Hodgins [Hod02]. Rechts neben dem Klassifikationsschema befindet sich die Zuordnung der Klassen zu den Komponenten des Aggregationsschemas von SCORM. Kurse entsprechen dabei in SCORM den Content-Packages. Lektionen werden je nach Komplexität entweder als SCO (Sharable Content Object) oder Asset realisiert. In der Regel werden hierfür Lektionen in HTML-Seiten konvertiert, die auch die Informationsblöcke enthalten. Diese referenzieren die Medienobjekte, die Assets darstellen. Von der größten, bis hin zur kleinsten Aggregationsstufe werden folgende Klassen von Dokumentknoten definiert:

Kurs Der Wurzelknoten eines Kursdokuments wird der Klasse Kurs zugeordnet. Sie stellen in der vorliegenden Arbeit die größte Organisationseinheit für aggregierte Inhalte dar. Kurse setzen sich aus einer beliebigen Anzahl von Lektionen zusammen. Als vorübergehender Zustand während der Kurserstellung sind auch Kurse zugelassen, die aus keiner Lektion bestehen. Als oberstes Element der Kursstruktur aggregieren Knoten dieser Klasse alle Inhalte, die zu einem Kurs für rechnergestützte Lehre gehören mit Hilfe der „HasPart“-Beziehung.

Lektion Die nächst kleinere Organisationseinheit ist die der Lektionen. Sie stellen die Bausteine für die Erstellung von Kursen dar. Sie haben ein Lernziel und sind eigenständige Sinneinheiten. Mit ihrer Hilfe können Kurse in

kleinere Einheiten, den Lektionen, gegliedert werden. Lektionen können sich entweder aus beliebig vielen weiteren Lektionen oder beliebig vielen Blöcken zusammensetzen. Werden durch eine Lektion weitere Lektionen aggregiert, so gelten die aggregierten Lektionen aufgrund des Gliederungscharakters als Unterlektionen der Lektion, die sie aggregiert.

Block Die Organisationseinheit Block stellt eine Menge von Medienobjekten dar, die zusammen eine Sinneinheit ergeben, jedoch kein Lernziel haben. Die Aufgabe dieser Organisationseinheit ist es, die Inhalte in Sinneinheiten zu gliedern. Diese Sinneinheiten können bei Bedarf als solche bei der Ausgabe dargestellt und wiederverwendbar gemacht werden. Ein Block setzt sich aus beliebig vielen verschiedenen Medienobjekten zusammen. Die Reihenfolge der aggregierten Medienobjekte wird gespeichert, wird aber aufgrund der verschiedenen Abbildungsvorschriften und Schablonen für das Layout von Inhalten möglicherweise nicht berücksichtigt.

Medienobjekt Medienobjekte stellen die Klasse mit dem kleinsten Aggregationsniveau dar. Die Dokumentknoten der Klasse Medienobjekt aggregieren keine Inhalte, sie stellen lediglich die physikalischen Daten der Medienkomponenten, wie Texte, Bilder, Animationen und Simulationen bereit, die durch die Dokumentknoten der anderen Klassen zu einem Kurs aggregiert werden.

Dieses Schema kann zur Klassifikation der Lernobjekte wie Kurse und deren Fragmente genutzt werden. Dabei entspricht jede Klasse im Klassifikationsschema einem Aggregationsniveau. Das Aggregationsschema legt die Regeln fest, nach denen Lernobjekte aggregiert werden können. Im nächsten Abschnitt wird diskutiert, welche Erweiterungen für das Dokumentmodell benötigt werden, um modularisierte beziehungsweise aggregierte Lernobjekte wie Kurse zu speichern.

4.2.5 Beschreibung aggregierter Lernobjekte

In den vorherigen Abschnitten wurde ein Verfahren zur Aggregation von Inhalten auf Basis von Metadaten nach [HSDK⁺03a] beschrieben. Dieses Verfahren besitzt den Vorteil, dass trotz der Aggregation der Inhalte auf Seite des Lernobjektarchivs keine speziellen Parser zur Bestimmung der Abhängigkeit zwischen den verschiedenen Bestandteilen eines Kurses benötigt werden. Damit auf Basis des in den letzten Abschnitten beschriebenen Dokumentmodells Lernobjekte, wie Kurse aus Lernobjektfragmenten aggregiert und deren Fragmente flexibel wiederverwendet werden können, sind aus folgenden Gründen Erweiterungen für das Dokumentmodell notwendig:

- Die vorgeschlagene Methode zur Speicherung der „HasPart“-Beziehungen besitzt die Eigenschaft, dass die Reihenfolge der Teile, aus denen sich ein innerer Knoten im Kursdokument zusammensetzt, nicht sichergestellt ist. Nach

dem LOM-Standard, der Teil der SCORM Spezifikation ist, ist die Liste der Beziehungen zu anderen Lernobjekten nicht geordnet. Das bedeutet, dass die Speicherung der Aggregationsbeziehung und deren Umkehrbeziehung in LOM allein nicht ausreichend ist, um ein Kursdokument mit festgelegter Reihenfolge der Lernobjektfragmente aufzubauen. Die Speicherung der Aggregationsbeziehungen in LOM bietet aber den Vorteil, dass diese ausgewertet und benutzt werden können, ohne spezielle Datenformate zur Aggregation von Inhalten zunächst analysieren zu müssen.

- Die ausschließliche Speicherung der Attribute eines Lernobjektfragments bringt die Einschränkung mit sich, dass die Attribute des Lernobjektfragments in jeder Aggregation verwendet werden. Eine Differenzierung der Attribute für verschiedene Anwendungsfälle bei der Verwendungen eines Lernobjektfragments ist nicht möglich. Abhilfe schafft die zusätzliche Speicherung verschiedener, für die Darstellung von Lernobjektfragmenten wichtiger Attribute, in der Beschreibung des aggregierten Lernobjekts. Mit Hilfe dieser zusätzlichen Attribute ist eine differenzierte Darstellung aggregierter Lernobjektfragmente möglich. In diesem Fall stellen die Attribute im Metadatensatz Standardwerte dar, die mit Hilfe der zusätzlich in der Beschreibung der Aggregation gespeicherten Attribute überschrieben werden können. Eine Anwendung hiervon ist beispielsweise die Wiederverwendung einer Simulation mit einer von der Standardgröße abweichenden Größe.

Aus diesen Gründen wird eine zusätzliche Beschreibung der aggregierten Lernobjekte benötigt. Ihr Ziel ist es die Aggregation der Lernobjektfragmente zu beschreiben und eventuell benötigte Eigenschaften beziehungsweise Parameter der aggregierten Lernobjektfragmente zu speichern. Die hierfür benötigte Beschreibung ist stark vom Anwendungsfall abhängig. Für das Klassifikationsschema aus dem vorangehenden Abschnitt müssen Lernobjekte der Klassen Kurs, Lektion und Block über eine Beschreibung verfügen. Für die Realisierung dieser Beschreibungen bietet sich XML an. In Abschnitt 7.3.1 wird das XML-Schema zur Beschreibung der Lernobjekte der genannten Klassen dargestellt, wie sie für die Implementierung des ResourceCenters verwendet wird.

4.3 Analyse des Autorenprozesses

In den vorangehenden Abschnitten des Kapitels wurde ein modulares Dokumentmodell zur Speicherung von Lernobjekten wie Kursen vorgestellt, das sich aus aggregierten Lernobjektfragmenten zusammensetzt. Dieses Dokumentmodell bildet die Grundlage für die Aggregation von Lernobjekten in einem auf Aggregation basierenden Autorenprozess. Darauf aufbauend erfolgt in diesem Abschnitt die Analyse

eines solchen Autorenprozesses. Als Zielanwendung des betrachteten Autorenprozesses liegt hierbei die Erstellung von Kursdokumenten zugrunde, die die Eigenschaften des in Abschnitt 4.2 definierten Kursdokumentformats besitzen. Zunächst erfolgt eine abstrakte Betrachtung eines Autorenprozesses zur Aggregation von Inhalten. Diese wird anschließend in einem zweiten Teil erweitert.

4.3.1 Abstraktes Prozessmodell

In diesem Abschnitt wird ein abstraktes Prozessmodell für einen auf Aggregation basierenden Autorenprozess entwickelt. Hierfür wird der in [GAP02] präsentierte Ansatz zur Zusammenstellung von Kurskomponenten auf rein technische Aspekte zur Aggregation von Inhalten reduziert. Dabei wird auf die Implementierung pädagogischer Modelle nicht eingegangen. Folgende vier Schritte werden für das abstrakte Prozessmodell identifiziert:

Initialisierung: Erzeugen eines leeren Lernobjekts für die Aggregation von Lernobjektfragmenten. Dabei kann das Lernobjekt einer Klasse mit beliebigem Aggregationsniveau angehören, mit der Einschränkung, dass Lernobjekte der entsprechenden Klasse aggregiert werden. In diesem Schritt werden zusätzlich erste Metadaten über das neu erstellte Lernobjekt gesammelt. Beispielsweise lassen sich schon jetzt die Werte für das Aggregationsniveau und die Struktur des Lernobjekts transparent festlegen.

Materialbeschaffung: Suche nach geeigneten, bereits bestehenden Lernobjektfragmenten und Produktion von Lernobjektfragmenten, die neu erstellt werden müssen.

Aggregation: Aggregation der im vorherigen Schritt gesammelten Lernobjektfragmente analog zu den im ersten Schritt festgelegten Metadaten, wie beispielsweise der Struktur.

Distribution: Verfügbarmachen des aus Lernobjektfragmenten neu zusammengestellten Lernobjekts. In diesem Schritt werden zusätzlich die neu erstellten Lernobjekte mit Metadaten beschrieben. Zusammen mit den Metadaten, werden die Lernobjekte danach in einem Lernobjektarchiv gespeichert.

Abbildung 4.8 zeigt die einzelnen Aktivitäten der zuvor genannten abstrakten Schritte zur Erstellung eines neuen Lernobjekts mittels Aggregation von Lernobjektfragmenten. Das Aktivitätsdiagramm² zeigt das Erstellen eines neuen Lernobjekts, das sowohl aus bestehenden, als auch aus neu erstellten Lernobjektfragmenten

²In den Aktivitätsdiagrammen dieser Arbeit wird zugunsten der besseren Übersichtlichkeit auf die Behandlung von Fehlerfällen und dem Abbruch von Aktivitäten verzichtet.

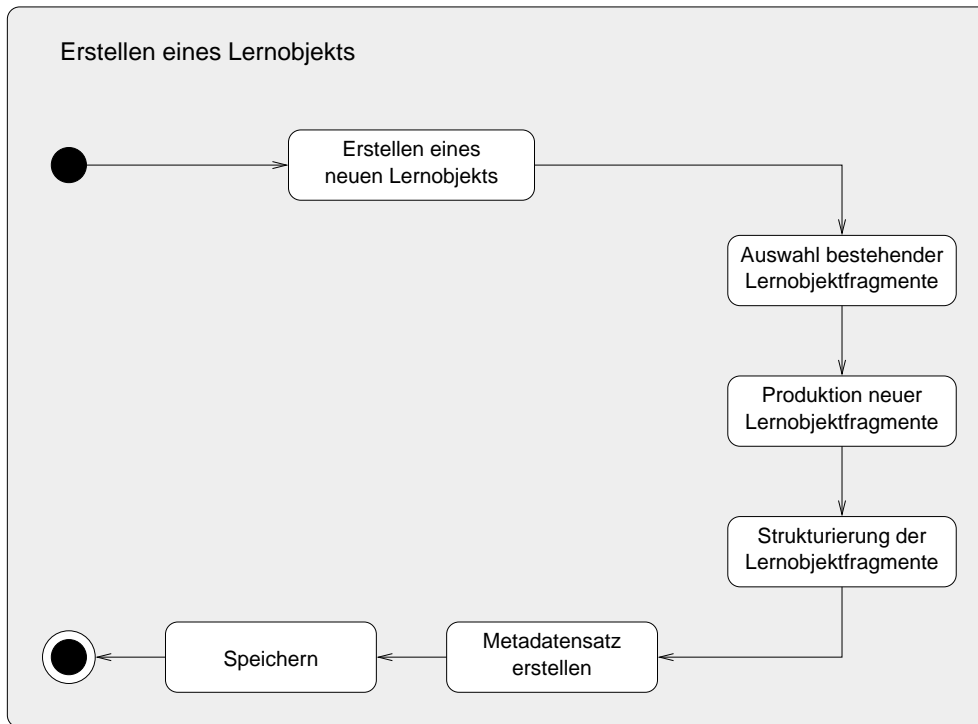


Abbildung 4.8: Abstraktes Prozessmodell

aggregiert wird. Zunächst wird ein leeres Lernobjekt erstellt, das noch keine Lernobjektfragmente aggregiert. Anschließend beginnt die Auswahl bestehender Lernobjektfragmente, die wiederverwendet werden sollen. Hierfür wählen die Autoren geeignete Lernobjektfragmente im Lernobjektarchiv aus. Im nächsten Schritt werden alle fehlenden Lernobjektfragmente produziert. Anschließend werden die ausgewählten und neu erstellten Lernobjektfragmente in die gewünschte Reihenfolge gebracht und so das neue Lernobjekt strukturiert. Danach werden die Metadaten des neu erstellten Lernobjekts erfasst. Mit Hilfe der Metadaten werden die neu erstellten Lernobjekte im Lernobjektarchiv indiziert, so dass sie auf Basis der Metadaten wiedergefunden werden können. Als letzter Schritt wird das neu erstellte Lernobjekt im Lernobjektarchiv gespeichert und so für andere Autoren verfügbar gemacht. Im Unterschied zu [GAP02] ist der in Abbildung 4.8 dargestellte Autorenprozess nicht auf die Produktion von Lernobjekten eines bestimmten Aggregationsniveaus beschränkt. Das heißt, dass nach diesem Algorithmus sämtliche Lernobjekte, die sich aus Lernobjektfragmenten zusammensetzen, erstellt werden können.

Nach [Sed98] wird mit einem rekursiven Algorithmus ein Problem gelöst, indem er eine oder mehrere kleine Instanzen des gleichen Problems löst. Für die Realisierung von rekursiven Problemen enthalten Algorithmen Rekursionen, das heißt, die Algorithmen rufen sich selbst auf. Der in Abbildung 4.8 dargestellte Autorenprozess

für die Erstellung neuer aggregierter Lernobjekte beinhaltet diese Rekursion. Die Aktivität „Produktion neuer Lernobjektfragmente“ beinhaltet die Erstellung neuer Lernobjekte mit niedrigerem Aggregationsniveau, die als Lernobjektfragmente der neu zu erstellenden Lernobjekte aggregiert werden. Das entspricht einer Rekursion im Algorithmus zur Erstellung von Lernobjekten. Die Rekursion im Autorenprozess ist so zu verwirklichen, dass aus einer Programminstanz zur Erstellung eines neuen Lernobjekts eine neue Instanz zur Erstellung eines Lernobjektfragments gestartet wird. Solange die Instanz zur Erstellung des Lernobjektfragments nicht beendet ist, befindet sich die Instanz zur Erstellung des neuen Lernobjekts im Zustand „Produktion neuer Lernobjektfragmente“. Aufgrund der Rekursion im Dokumentmodell, das in Abschnitt 4.2 beschrieben wurde, bietet sich das Bearbeiten von Lernobjekten in einem rekursiven Algorithmus an.

4.3.2 Erweitertes Prozessmodell

In der Realität werden die einzelnen Aktivitäten zur Erstellung eines neuen Lernobjekts in den wenigsten Fällen von den Autoren in der Reihenfolge durchgeführt, in der sie nach Abbildung 4.8 dargestellt sind. Aus diesem Grund sollte der Autorenprozess so gestaltet sein, dass die einzelnen Aktivitäten zur Erstellung und Bearbeitung von Lernobjekten in beliebiger Reihenfolge ausgeführt werden können. Zur Modellierung dieses Sachverhaltes ist eine Schleife für die Bearbeitung der Lernobjekte in das Aktivitätsdiagramm einzufügen, mittels der die Arbeitsschritte für das Auswählen von Lernobjektfragmenten, das Produzieren neuer Lernobjektfragmente und das Strukturieren der Lernobjektfragmente in beliebiger Reihenfolge geschehen können.

Im Folgenden wird deswegen der in Abbildung 4.8 dargestellte Autorenprozess derart erweitert, dass die verschiedenen Bearbeitungsschritte von den Autoren in beliebiger Reihenfolge ausgeführt werden können. Darüber hinaus wird der Autorenprozess so erweitert, dass auch die Bearbeitung bereits bestehender Lernobjekte möglich ist. Verschiedene Teilaktivitäten können auf dem neu zu erstellenden oder bereits bestehenden Lernobjekt durchgeführt werden.

Abbildung 4.9 zeigt das Aktivitätsdiagramm des erweiterten Autorenprozesses. Zu Beginn wird unterschieden, ob ein neues Lernobjekt erstellt oder ein bereits bestehendes Lernobjekt bearbeitet werden soll. Für den Fall, dass ein neues Lernobjekt erstellt werden soll, wird ein leeres Lernobjekt erstellt und anschließend ein Metadatenatz für das neue Lernobjekt erzeugt. Damit ist die Erstellung der Metadaten vor die eigentliche Bearbeitung des Lernobjekts gerückt. Der Grund hierfür liegt an der Verzahnung der Erstellung von Metadaten mit der Erstellung von Lernobjekten. Für den Fall, dass die initial erstellten Metadaten nachträglich bearbeitet werden sollen, können diese auf Wunsch zu einem späteren Zeitpunkt geändert werden. Für den Fall, dass ein bestehendes Lernobjekt bearbeitet werden soll, ist dieses zunächst auf geeignete Weise auszuwählen. Danach gelangt der

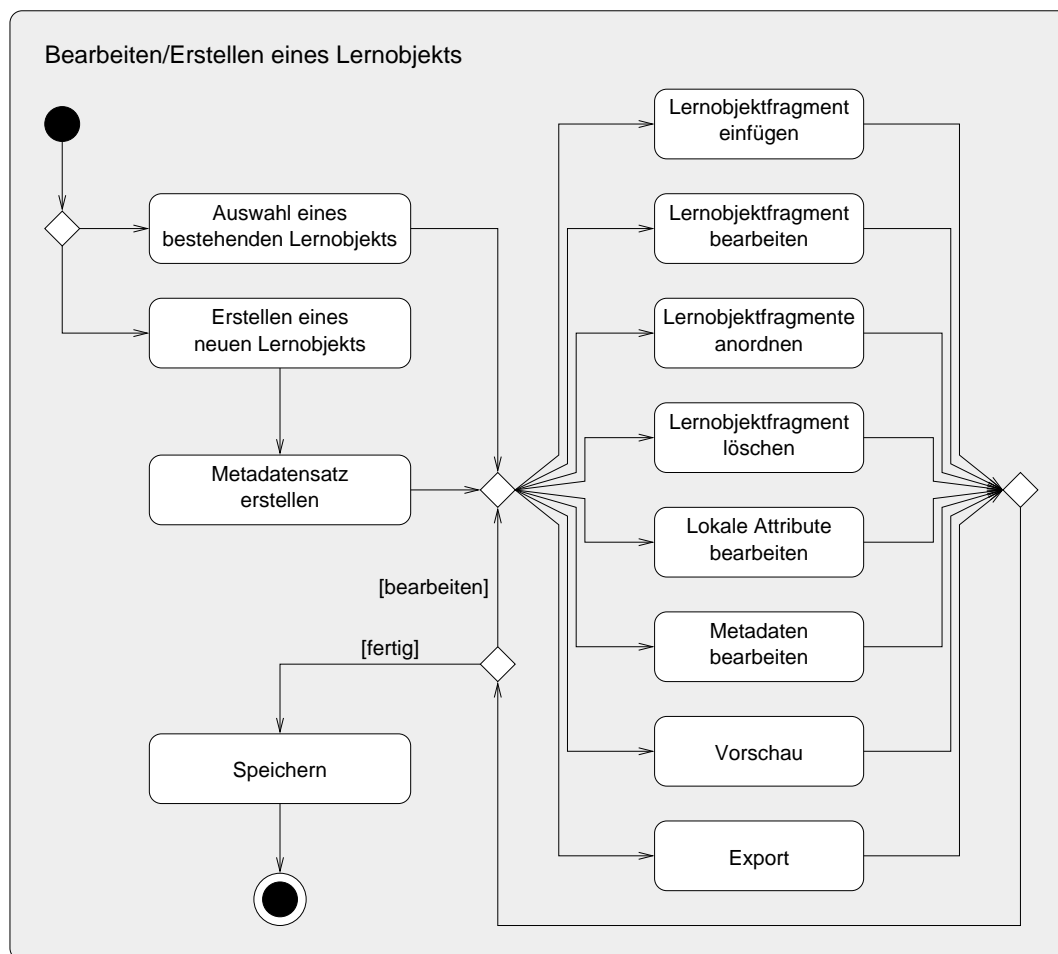


Abbildung 4.9: Erweiterte Modellierung des Autorenprozesses

Autorenprozess nach dem Aktivitätsdiagramm in eine Schleife, in der verschiedene Aktionen auf dem zu erstellenden beziehungsweise zu bearbeitenden Lernobjekt durchgeführt werden können. Für jeden Zyklus dieser Schleife kann der Autor des Lernobjekts selbst wählen, welche Aktivität durchgeführt wird. Sobald die Arbeiten abgeschlossen sind, wird als letzte Aktivität des Autorenprozesses das betreffende Lernobjekt inklusive des dazugehörigen Metadatensatzes gespeichert.

Folgende Aktivitäten können innerhalb des Bearbeitungszyklusses durchgeführt werden:

Lernobjektfragment einfügen Diese Aktivität beschreibt das Einfügen eines Lernobjektfragments in ein Lernobjekt an einer frei wählbaren Stelle. Sie ist in weitere Subaktivitäten zu untergliedern: Im ersten Schritt muss die Position selektiert werden, an die das einzufügende Lernobjektfragment integriert werden soll. Danach ist die Quelle auszuwählen, auf der das einzufügende

Lernobjektfragment gespeichert ist. Für das Einfügen mit lokalen Werkzeugen erstellter Medienkomponenten ist beispielsweise das lokale Dateisystem auszuwählen. Für den Fall, dass ein innerhalb eines Lernobjektarchivs bestehendes Lernobjekt als Lernobjektfragment eingefügt werden soll, wird als Quelle das Lernobjektarchiv gewählt. Nach der Wahl der Quelle muss im zweiten Schritt das Lernobjektfragment ausgewählt werden. Für das lokale Dateisystem bringen hierfür die Betriebssysteme geeignete Auswahlhilfen mit. Für die Auswahl eines bereits existierenden Lernobjekts im Lernobjektarchiv müssen geeignete Auswahlmechanismen, die beispielsweise auf den Metadatenbeschreibungen der gespeicherten Lernobjekte basieren, bereitgestellt werden. Nach der Auswahl eines Lernobjektfragments wird dieses an der im ersten Schritt ausgewählten Stelle in das in Bearbeitung befindliche Lernobjekt eingefügt.

Alternativ zum lokalen Dateisystem und dem Lernobjektarchiv kann auch das Autorensystem wiederum als Quelle eines Lernobjektfragments dienen, mittels dessen Inhalte erstellt und an entsprechender Stelle eingefügt werden können. Das entspricht der zuvor bereits angesprochenen Rekursion des Autorenprozesses. Als Lernobjektfragment wird in diesem Fall ein neues Lernobjekt eines niedrigeren Aggregationsniveaus eingefügt, das zur Bearbeitungszeit des Lernobjekts eigens hierfür erstellt wird. Dazu wird eine weitere Instanz des Autorenprozesses gestartet, mit der das neue Lernobjekt erstellt wird, das danach anschließend als Lernobjektfragment des Lernobjekts eingefügt wird, für das es erstellt wurde. Solange die Instanz zur Erstellung der Lernobjektfragmente nicht die Aktivität „Speichern“ durchlaufen hat, verharret der Autorenprozess des Lernobjekts, das durch ein weiteres Lernobjektfragment ergänzt werden soll, im Zustand „Lernobjektfragment einfügen“.

Lernobjektfragment bearbeiten Mit dieser Aktivität wird die Bearbeitung eines Lernobjektfragments eingeleitet. Ebenso wie das Einfügen eines Lernobjektfragments stellt das Bearbeiten eines Lernobjektfragments eine Rekursion im Autorenprozess dar. Für die Bearbeitung eines Lernobjektfragments wird deswegen eine weitere Instanz des Autorenprozesses gestartet. Nach der Auswahl des zu bearbeitenden Lernobjektfragments gelangt der Autor direkt in die Bearbeitungsschleife des Autorenprozesses, in der die Bearbeitung der Lernobjektfragmente in beliebiger Reihenfolge geschehen kann. Solange die Instanz zur Bearbeitung des Lernobjektfragments aktiv ist, bleibt die Instanz des Autorenprozesses zur Bearbeitung des Lernobjekts, dessen Lernobjektfragment dabei bearbeitet wird, in der Aktivität „Lernobjektfragment bearbeiten“ stehen.

Lernobjektfragmente anordnen Durch diese Aktivität wird die Reihenfolge der Lernobjektfragmente geändert, aus denen sich das Lernobjekt zusammen-

setzt. Das Ändern der Reihenfolge von Lernobjektfragmenten eines aggregierten Lernobjekts kann auf verschiedenen Wegen geschehen. Im einfachsten Fall werden zwei benachbarte Lernobjektfragmente miteinander in ihrer Reihenfolge vertauscht. Hierfür genügt es ein Lernobjektfragment auszuwählen und zu bestimmen, ob dieses in der Reihenfolge weiter nach vorne oder weiter nach hinten verschoben werden soll. Die Anordnung zweier nicht benachbarter Lernobjektfragmente eines Lernobjekts kann beispielsweise über eine Zwischenablage durch Kombination der Aktivitäten „Lernobjektfragment löschen“ und „Lernobjektfragment einfügen“ erreicht werden. Dabei wird die Referenz auf ein Lernobjektfragment beim Löschen in die Zwischenablage kopiert und anschließend beim Einfügen die entsprechende Referenz aus der Zwischenablage wieder hervorgeholt.

Lernobjektfragment löschen Ziel dieser Aktivität ist es ein beliebiges Lernobjektfragment innerhalb eines Lernobjekts zu löschen. Hierfür muss dieses Lernobjektfragment zunächst ausgewählt werden. Löschen heißt in diesem Zusammenhang, dass das Lernobjektfragment nicht mehr Teil des Lernobjekts ist. Hierfür wird die Referenz auf das entsprechende Lernobjektfragment in dem Lernobjekt gelöscht, so dass dieses nicht mehr Teil des Lernobjekts ist. Das Lernobjektfragment ist jedoch noch im Lernobjektarchiv verfügbar.

Lokale Attribute bearbeiten Mit Hilfe dieser Aktivität können die Werte lokaler Attribute bearbeitet werden. Sie werden benötigt, um Lernobjektfragmente, die als Teil eines Lernobjekts referenziert werden, für genau diese eine Verwendung zu parametrisieren, ohne dabei auf globale Werte aus den Metadaten zurückgreifen zu müssen. Ein Beispiel für die Verwendung von lokalen Attributen ist die Parametrisierung von Simulationen. Simulationen können häufig durch Parametrisierung in bestimmte Zustände versetzt werden, um bestimmte Szenarien durchspielen zu können. In der Regel wird die Simulation für jedes Szenario einzeln referenziert. Mit Hilfe der lokalen Attribute kann die Simulation jeweils in einen bestimmten Zustand gebracht werden.

Metadaten bearbeiten Diese Aktivität bietet die Möglichkeit die Metadaten des aktuell in Bearbeitung befindlichen Lernobjekts zu bearbeiten. Das ist beispielsweise dann erforderlich, wenn die automatisch erstellten Metadaten, die zusammen mit dem leeren Lernobjekt für die Aggregation von Lernobjektfragmenten erstellt worden sind, bearbeitet werden sollen. Diese Aktivität kann aber auch dann genutzt werden, wenn der Metadatensatz aufgrund der durchgeführten Arbeiten an dem Lernobjekt erweitert oder geändert werden soll.

Vorschau Diese Aktivität ermöglicht die Vorschau eines Lernobjekts, das sich in Bearbeitung befindet. Sie wird benötigt, da es aufgrund der Rekursion der

Prozessinstanzen der Autorenumgebung oft nicht möglich ist, alle Inhalte des in Bearbeitung befindlichen Lernobjekts auf einen Blick zu sehen. Ein weiterer Grund für die Notwendigkeit der Vorschaufunktion ist, dass in der Regel Lernobjekte während der Erfassung möglichst ohne Formatierung erstellt werden, die direkt das Erscheinungsbild prägen, um eine bessere Wiederverwendung der Lernobjekte zu ermöglichen. Inhalt und Layout werden so getrennt. Durch die Verwendung einer Layoutvorlage kann die endgültige Fassung der Lernobjekte mit Hilfe der Vorschaufunktion angezeigt werden.

Export Mit Hilfe dieser Aktivität können aggregierte Inhalte exportiert werden. Bei dieser Aktivität kommen die Layoutvorlagen zum Einsatz, die bereits bei der Aktivität „Vorschau“ erwähnt wurden. Dabei werden die aggregierten Inhalte in ein anderes Format überführt, das von anderen Systemen weiterverarbeitet werden kann. Mit Hilfe der Exportfunktion lassen sich beispielsweise aggregierte Kurse als SCORM-konformes Kurspaket exportieren.

Durch Ableitung spezialisierter Klassen zur weiteren Differenzierung von Kursdokumentknoten wird statt dieser abstrakten Sicht auf den Autorenprozess ein auf die definierten Dokumentknoten abgeleiteter Autorenprozess benötigt. Bezogen auf einzelne Klassen dieses Dokumentmodells besitzen die abgeleiteten Autorenprozesse nur dann Rekursionen, wenn Rekursionen durch das Dokumentmodell für diese Klasse definiert sind. Außerdem werden nicht alle Aktivitäten in den abgeleiteten Autorenprozessen für die Bearbeitung von Instanzen der verschiedenen Klassen des Dokumentmodells benötigt. Die abgeleiteten Autorenprozesse zur Bearbeitung der aggregierten Lernobjekte des Dokumentmodells werden in Abschnitt 7.3.2 beschrieben.

4.4 Zusammenfassung

Auf Basis der zu Beginn des Kapitels durchgeführten Analyse der Anforderungen an einen auf Aggregation basierenden Autorenprozess wurde als Grundlage für einen solchen Autorenprozess ein generisches Dokumentmodell vorgestellt. Dieses Dokumentmodell stellt die modulare Speicherung der Lernobjekte und deren Lernobjektfragmente sicher, so dass die Lernobjektfragmente ohne vorherige Modularisierung der Lernobjekte wiederverwendet werden können. Für die Modularisierung der Lernobjekte werden vier Aggregationsniveaus inklusive einem Aggregationsschema zur Aggregation von Lernobjekten definiert. Die Aggregation von Lernobjektfragmenten beruht dabei auf deren Referenzierung. Im Unterschied zu anderen Ansätzen erlaubt dieses generische Dokumentmodell die Referenzierung von Lernobjekten bis hinunter zu den kleinsten Aggregationsniveaus. Innerhalb des Dokumentmodells werden die Lernobjekte und deren Fragmente als „Black-box“ betrachtet, so dass die Aggregation von Lernobjekten unabhängig von dem

gewählten Datenformaten angewendet werden kann. Das erleichtert die Verwaltung der Lernobjekte, weil zum Feststellen der Aggregationsbeziehungen nicht die Daten der Lernobjekte interpretiert, sondern lediglich die Metadaten herangezogen werden müssen.

Auf Basis dieses Dokumentmodells und den grundlegenden Anforderungen an Autorenprozesse, die auf Aggregation basieren, wurde ein abstraktes Prozessmodell für die Aggregation von Lernobjekten entwickelt. Das Prozessmodell dient als Konzept für die spätere Entwicklung von EditorKomponenten im ResourceCenter (vgl. Kapitel 7) für Lernobjekte der jeweiligen Aggregationsniveaus. Im Autorenprozess wird hierfür die Rekursion von Teilkomponenten ermöglicht, die von dem abstrakten Prozessmodell abgeleitet sind, was dem rekursiv definierten Dokumentmodell entspricht. Der Autorenprozess unterstützt sowohl die Wiederverwendung bereits bestehender als auch die Verwendung neu erstellter Lernobjekte. Das abstrakte Modell des Autorenprozesses macht dabei keine Einschränkungen bezüglich der Quellen, aus denen Lernobjekte zur Wiederverwendung stammen können. Dabei werden sowohl Lernobjektarchive als auch lokale Dateisysteme berücksichtigt.

5 Metadatengenerierung

In dem folgenden Kapitel wird ein Verfahren für eine weitestgehend automatisierte Metadatengenerierung beschrieben. Dieses Verfahren erweitert und kombiniert die in Kapitel 3.4 beschriebenen Ansätze. Es zeichnet sich dadurch aus, dass es in den Autorenprozess, der in Kapitel 4 präsentiert wurde, für die Generierung von Metadaten begleitend zur Erstellung von Lernobjekten integriert werden kann. Durch die weitestgehende Automatisierung soll der Aufwand für die Metadatengenerierung deutlich reduziert werden. Hierfür werden zunächst verschiedene Methoden zur Erstellung von Metadaten vorgestellt und Funktionen beziehungsweise Einsatzgebiete diskutiert. Anschließend erfolgt eine Bewertung der Verfahren hinsichtlich der Zuverlässigkeit der generierten Metadaten und der Arbeitserleichterung die bei den Autoren durch deren Anwendung zu erwarten ist. Danach wird auf Basis dieser Bewertung ein semiautomatisches Verfahren zur Generierung von Metadaten vorgestellt. Dabei werden alle obligatorischen Datenfelder des gewählten Metadatenschemas berücksichtigt. Zum Schluss wird die Integration des Verfahrens in den auf Aggregation basierenden Autorenprozess beschrieben.

5.1 Methoden zur Erstellung von Metadaten

In diesem Abschnitt werden verschiedene Verfahren für die Erfassung und Erstellung von Metadaten beschrieben und miteinander verglichen. Unter anderem wird hierfür die manuelle Metadatenerstellung und die Nutzung von Metadaten-schablonen, Abhängigkeiten im Metadaten-schema, Kontextinformationen und bereits bestehende Metadaten diskutiert. Dabei wird auch die Gewinnung von Metadaten durch Inhaltsanalyse, die Nutzung systembedingter Metadaten und das Ableiten von Metadaten-sätzen berücksichtigt. Anschließend werden die verschiedenen Verfahren hinsichtlich der Zuverlässigkeit der Metadaten, dem technischen Aufwand für die Realisierung des Verfahrens und dem Nutzen, den das entsprechende Verfahren für die Autoren darstellt, verglichen.

5.1.1 Manuelle Erfassung von Metadaten

Metadateneditoren stellen Werkzeuge zum Erstellen und Bearbeiten von Metadaten-sätzen dar. Der Begriff Metadateneditor umfasst dabei eine breite Spanne verschiedener Ansätze für das Erstellen und Bearbeiten von Metadaten. In diesem

Abschnitt werden Metadateneditoren betrachtet, die ausschließlich der manuellen Erfassung von Metadaten dienen.

Das zentrale Bedienelement eines solchen Metadateneditors ist eine Eingabemaske, mittels der die einzelnen Metadatenfelder bearbeitet werden können. Dieses Eingabefeld ist in der Regel einem Formular nachempfunden. Das Formular ist im Allgemeinen an das jeweils verwendete beziehungsweise unterstützte Metadaten-schema angelehnt. Dabei werden die Strukturierung und Anordnung der Datenfelder in der Regel an das verwendete Metadaten-schema angepasst. Je nach Anwendung und Implementierung des Metadateneditors können alle, ausgewählte oder nur eine Mindestanzahl von Datenfeldern für die Bearbeitung im formularbasierten Eingabefeld angezeigt werden [KESS01].

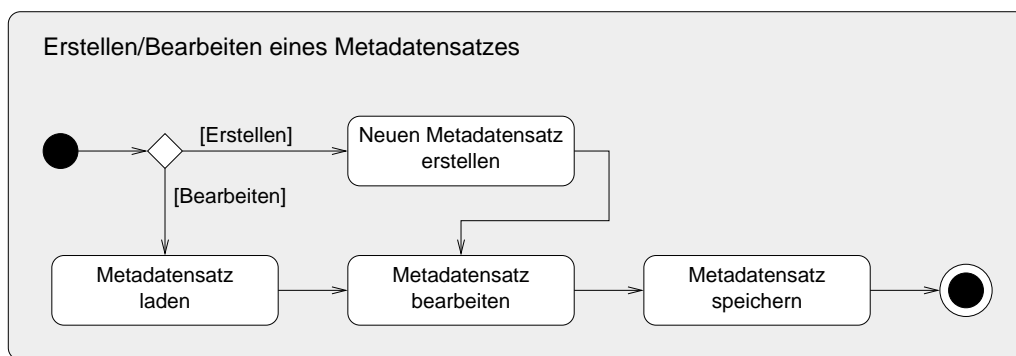


Abbildung 5.1: Einfache, formularbasierte Metadateneditoren

Abbildung 5.1 zeigt das Aktivitätsdiagramm zum Erstellen und Bearbeiten von Metadatenansätzen mit einem einfachen, formularbasierten Metadateneditor. Für den Fall, dass ein neuer Metadatenansatz erstellt werden soll, wird ein neuer leerer Datensatz erzeugt. Andernfalls wird ein bestehender Metadatenansatz für dessen Bearbeitung geladen. Anschließend wird der neu erstellte, leere Metadatenansatz beziehungsweise der bestehende, geladene in dem formularbasierten Eingabeelement angezeigt. Mittels des Eingabeelementes kann der entsprechende Datensatz jetzt bearbeitet werden. Ist die Bearbeitung abgeschlossen, wird der Metadatenansatz abgespeichert. Je nach Anwendung wird der Metadatenansatz im Dateisystem oder einer Datenbank abgelegt.

Einige bestehende Metadateneditoren besitzen verschiedene Hilfen, mittels der die Erstellung der Metadaten erleichtert werden kann. Im Folgenden werden verschiedene Ansätze zur Erleichterung der Metadatenerstellung vorgestellt.

5.1.2 Metadaten-schablonen

Bei einfachen formularbasierten Metadateneditoren ist es eine gängige Praxis, beim Erstellen von neuen Metadatenansätzen auf leere Datensätze aufzusetzen. Hierbei zei-

gen die Metadateneditoren nach dem Erzeugen des leeren Datensatzes eine ebenso leere Ansicht des Metadatensatzes in der formularbasierten Eingabemaske, die vom Metadatenautoren auszufüllen ist. Das erfordert einen hohen Aufwand seitens der Metadatenautoren.

In der Praxis zeigt sich jedoch schnell, dass die Metadatensätze von Lernobjekten einer bestimmten Klasse teilweise dieselben Werte für verschiedene Metadatenfelder besitzen. Professionelle Metadatenautoren arbeiten deswegen mit Schablonen beziehungsweise Templates. Unter Verwendung von Schablonen können Datenfelder, deren Werte für eine bestimmte Klasse von Lernobjekten bereits feststehen, mit Werten aus der Metadatenschablone übernommen werden [CMD05]. Die restlichen Metadaten werden manuell ergänzt.

Metadatenschablonen eignen sich auch um Datenfelder mit Werten zu belegen, die mit großer Wahrscheinlichkeit zutreffen. Die entsprechenden Datenfelder werden hierdurch mit Vorgabewerten gefüllt, die bei Bedarf angepasst werden müssen. Die Gefahr bei der Verwendung von Vorgabewerten liegt darin, dass die Anpassung der Vorgabewerte vergessen wird. Durch die Verwendung von Vorgabewerten können bei unachtsamer Verwendung falsche Metadaten erstellt werden. Aus diesem Grund sollte die Arbeit mit Vorgabewerten geschulten Metadatenautoren vorbehalten bleiben.

Ein Beispiel für eine sinnvolle Anwendung von Metadatenschablonen ist die Erstellung von Metadaten zu einer Menge von Lernobjekten eines Fachautoren durch einen ausgebildeten Metadatenautor. Dieser teilt die einzelnen Lernobjekte in Gruppen ein, für die ähnliche Metadatensätze zu erstellen sind. Für jede dieser Gruppen erstellt er eine Schablone, die sich weitestgehend aus der Schnittmenge aller Metadatensätze der Gruppe zusammensetzt. Die Schablonen dienen als Grundlage bei der Erstellung der Metadaten einer Gruppe von Lernobjekten.

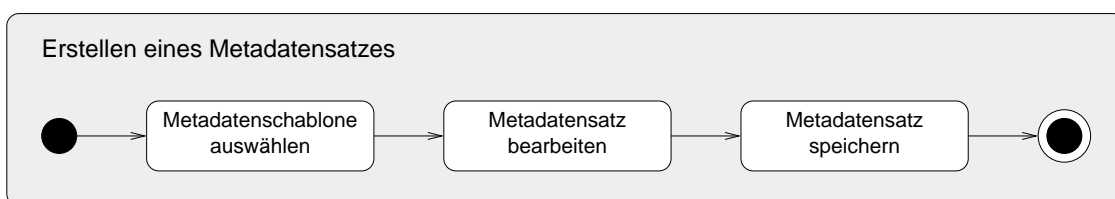


Abbildung 5.2: Erstellen von Metadatensätzen mit Metadatenschablonen

Abbildung 5.2 zeigt die Integration von Schablonen in den Prozess zur Erstellung von Metadatensätzen. Im ersten Schritt der Metadatenerstellung wird nach dem Aktivitätsdiagramm eine passende Schablone ausgewählt. Die Schablone entspricht einem vorgefertigten Metadatensatz, der im zweiten Schritt mit Hilfe des Metadateneditors weiterbearbeitet werden kann. Ist die Bearbeitung des Metadatensatzes abgeschlossen, wird der Metadatensatz zur weiteren Verwendung gespeichert. Das Aktivitätsdiagramm sieht nicht das Erstellen von leeren Metadatensätzen für den

Fall vor, in dem keine bestehende Schablone verwendet werden kann. Der Grund hierfür ist, dass ein leerer Metadatensatz als spezielle Form einer Schablone betrachtet werden kann, nämlich eine leere Schablone. Diese leere Schablone sollte als Standardschablone in die Auswahlliste vorhandener Schablonen aufgenommen werden, damit man darauf zugreifen kann, wenn keine der anderen Schablonen angewendet werden kann.

Metadatenschablonen stellen im Prinzip eine gute Hilfe beim Erstellen von Metadaten dar. Sie sind in ihrer Anwendung jedoch recht starr. Wie im nächsten Abschnitt gezeigt wird, basiert die Erstellung von Schablonen neben Gemeinsamkeiten der zu beschreibenden Objekte ebenfalls auf der Abhängigkeit verschiedener Datenfelder im Metadatenschema und der Verwendung von Kontextinformationen.

5.1.3 Nutzung von Abhängigkeiten im Datenschema

Die Abhängigkeit eines Metadatenfeldes von einem oder einer Menge anderer Datenfelder im Metadatenschema liegt dann vor, wenn die Definition der Datenfelder nicht orthogonal ist, so dass sie ein nicht orthogonales Koordinatensystem im Merkmalsraum aufspannen, oder aufgrund gegebener Eigenschaften der Lernobjekte eine Abhängigkeit zwischen Datenfeldern entsteht. Das bedeutet, dass Wechselbeziehungen zwischen einzelnen oder mehreren Datenfeldern im Metadatenschema bestehen. Diese bewirken eine Abhängigkeit der Werte einer Gruppe von Datenfeldern untereinander. In [NTD03] wird eine Abhängigkeit zwischen Datenfeldern der im Ariadne-KPS gespeicherten Metadatensätze nachgewiesen.

Wechselbeziehungen zwischen einzelnen Elementen des Metadatenschemas können genutzt werden, um auf Basis eines bereits bestehenden Wertes eines anderen Datenfeldes den Wert eines mit diesem abhängigen Datenfeldes zu bestimmen oder zumindest die Wertemenge für dieses Datenfeld einzuschränken. In günstigen Fällen ist es sogar möglich, die einzelnen Werte einer Gruppe von Datenfeldern durch die Angabe eines einzelnen Wertes zu bestimmen.

Wenn keines der voneinander abhängigen Datenfelder geeignet ist, die Werte der anderen Datenfelder zu bestimmen, kann hierbei die Definition eines neuen Datenfeldes weiterhelfen, das alle Eigenschaften der zu bestimmenden Metadatenfelder beinhaltet. Mittels dieses neu definierten Datenfeldes können dann alle voneinander abhängigen Datenfelder des Metadatenschemas bestimmt werden. Das neu definierte Datenfeld wird dabei nicht im Datensatz gespeichert, sondern dient lediglich der Generierung von Werten standardisierter Datenfelder. Bei zwei voneinander abhängigen Datenfeldern im Metadatenschema kann hierdurch das Füllen beziehungsweise Bearbeiten eines Datenfeldes eingespart werden.

Ein Beispiel für die Abhängigkeit von Datenfeldern lässt sich schnell in geschlossenen Systemen finden, in denen Lernobjekte mit Metadaten beschrieben werden sollen, die fest definierte Ressourcentypen haben. In einem solchen System kann eine Abhängigkeit zwischen den Werten etwaiger Datenfelder, wie Ressourcentyp und

Aggregationsniveau, beobachtet werden. In der Regel existiert bei diesen beiden Datenfeldern eine eindeutige Abbildung von dem Ressourcentyp zum Aggregationsniveau der Lernobjekte. Diese kann genutzt werden, mittels des Wertes für den Ressourcentyp das Aggregationsniveau eines Lernobjekts zu bestimmen. Umgekehrt lässt sich in der Regel die Bestimmung des Ressourcentyps auf Basis des Aggregationsniveaus aus Mangel einer eindeutigen Abbildungsvorschrift nicht durchführen. Bestenfalls ist es in einem solchen Fall möglich den Wertebereich für den Ressourcentyp auf Basis der Aggregation der Lernobjekte einzuschränken.

Bei der Nutzung der Abhängigkeit zwischen verschiedenen Datenfeldern im Metadatenschema in der Benutzerschnittstelle ist zwischen der Erstellung und der Bearbeitung von Metadaten zu unterscheiden. Bei der Erstellung von Metadaten können Abhängigkeiten zwischen einer Gruppe von Datenfeldern im Metadatenschema dazu genutzt werden, eine vereinfachte Benutzerschnittstelle zu entwerfen, bei der für ein Gruppe abhängiger Metadatenfelder im besten Fall nur ein Datenwert eingegeben werden muss. Beim Bearbeiten von Metadatenätzen ist die Nutzung der Abhängigkeit von Datenfeldern in der Benutzerschnittstelle eines Metadateneditors aufgrund fehlender eindeutiger Abbildungsvorschriften in der Regel problematisch. Die Koppelung verschiedener voneinander abhängiger Datenfelder im Metadateneditor ist in der Regel aufgrund der fehlenden eindeutigen Abbildungsvorschriften nicht möglich. Existiert eine eindeutige Abbildung zwischen zwei voneinander abhängigen Datenfeldern in eine Richtung, so ist es möglich das Datenfeld, das aus dem Wert des anderen Datenfeldes bestimmt werden kann, vor Veränderung durch den Metadatenautoren zu schützen. Statt dieses Datenfeld direkt zu verändern, kann der Wert dieses Datenfeldes über den Umweg einer Abbildung über ein anderes Datenfeld geändert werden. Diese Umsetzung ist sicherlich gewöhnungsbedürftig und wird möglicherweise nicht angenommen. Wenn die Werte abhängiger Datenfelder im Metadateneditor jedoch nicht gekoppelt werden, besteht die Gefahr durch Fehlbedienung inkonsistente Werte für abhängige Datenfelder zu erhalten.

Die in diesem Abschnitt beschriebenen Abhängigkeiten von Datenfeldern zu anderen Datenfeldern des Metadatenschemas sind unter Anwendung der dargestellten Möglichkeiten zur Bestimmung von Metadaten als eine Art von Subschablonen für Metadaten anzusehen. Von dieser Subschablone betroffene Datenfelder beziehen sich nur auf ein Gruppe voneinander abhängiger Datenfelder. Die Auswahl der zutreffenden Subschablone hängt dabei, analog von den im vorherigen Abschnitt beschriebenen Metadaten-schablonen, von wenigen Eigenschaften der auszuzeichnenden Lernobjekte ab. Subschablonen können für jede Gruppe abhängiger Datenfelder jeweils unabhängig verwendet werden.

5.1.4 Nutzung von Kontextinformationen

Teilweise umfassen Metadaten-schemata Datenfelder, die den Kontext der zu beschreibenden Lernobjekte betreffen. Zur automatischen Bestimmung der Werte

dieser Datenfelder können deshalb zum Zeitpunkt der Erstellung vorliegende Kontextinformationen herangezogen werden. Je nach verwendetem Metadatenschema betreffen die erforderlichen Kontextinformationen in der Regel unterschiedliche Bereiche des Lebenszyklusses der zu beschreibenden Lernobjekte.

Die Nutzung von Kontextinformationen dient in der Regel hauptsächlich der automatischen Erstellung von Metadaten. Hierbei werden alle verfügbaren und verwendbaren Kontextinformationen gesammelt und in den Metadatensatz kopiert. Sind benötigte Kontextinformationen nicht verfügbar, können sie, falls möglich, prognostiziert werden und als Vorschlag in den Metadatensatz kopiert werden. Das ergibt allerdings nur dann Sinn, wenn die entsprechenden Werte mit relativ hoher Sicherheit richtig bestimmt werden können und das Verfahren, nachdem die entsprechenden Werte prognostiziert werden, bekannt ist, so dass Metadatenautoren leicht überblicken können, wann es erforderlich ist, einen Vorschlag zu ändern.

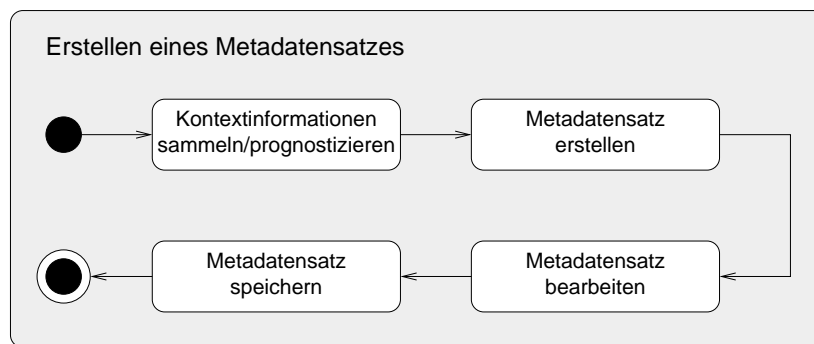


Abbildung 5.3: Kontextinformationen bei der Erstellung von Metadaten nutzen

Das Aktivitätsdiagramm in Abbildung 5.3 zeigt die Nutzung von Kontextinformationen bei der Erstellung von Metadaten. Vor der eigentlichen Erstellung eines Metadatensatzes werden die verfügbaren und nutzbaren Kontextinformationen gesammelt. Hierfür können beispielsweise Benutzerdaten herangezogen werden, die aus dem Anwenderkontext stammen. Aus dem Anwendungskontext ergeben sich je nach Anwendung weitere Metadaten, wie beispielsweise Informationen über die Versionierung von Lernobjekten. Falls möglich und erforderlich werden bei dieser Aktivität auch Kontextinformationen prognostiziert. Die gesammelten und prognostizierten Kontextinformationen dienen dann bei der zweiten Aktivität als Grundlage für die Erstellung eines neuen Datensatzes. Dabei werden die Kontextinformationen in den leeren, neu erstellten Metadatensatz eingefügt. Falls prognostizierte Kontextinformationen geändert werden müssen und weitere Metadaten dem Metadatensatz hinzugefügt werden müssen, werden sie im dritten Schritt im Metadateneditor zur Bearbeitung angeboten. Sind die beiden Voraussetzungen für die Bearbeitung des Metadatensatzes nicht erfüllt, kann dieser Arbeitsschritt entfallen. Nachdem die Metadaten bei Bedarf geändert beziehungsweise ergänzt worden sind, werden sie

im letzten Arbeitsschritt abgespeichert.

Ein Beispiel eines verwendbaren Kontextes ist der Anwenderkontext eines Autors, der seine Lernobjekte mit Metadaten beschreibt. Für den Fall, dass das Metadaten-schema die Speicherung von Informationen über die Autoren der Lernobjekte vorsieht, enthält das Benutzerprofil des Autors für die Metadatenerstellung verwertbare Informationen. In diesem Fall kann der Name des Autors und gegebenenfalls die Adresse und weitere Kontaktinformationen aus dem Benutzerprofil entnommen werden und im Metadatensatz gespeichert werden. Weitere Kontexte, die für die Erstellung von Metadaten interessant sind und die allesamt mit dem Lebenszyklus der Lernobjekte selbst in Zusammenhang stehen, sind die Kontexte, in denen die Lernobjekte erstellt und verwendet werden. Schon beginnend bei der Erstellung von Lernobjekten ist das Erfassen von Metadaten sinnvoll, um beispielsweise alle Personen, die für die Fertigstellung eines Lernobjekts zuständig sind, festhalten zu können. Bei der späteren Verwendung von Lernobjekten ist die lückenlose und automatische Erfassung über deren Verwendung mit Hilfe des Verwendungskontextes möglich.

Im Unterschied zur Nutzung von Metadaten-schablonen bietet die Nutzung von Kontextinformationen jedoch eine größere Flexibilität bei häufig wechselnden Werten, wie beispielsweise dem Erstellungsdatum von Lernobjekten, das für jedes Lernobjekt im Metadatensatz eingetragen werden kann. Ganz zu schweigen von dem Zusatzaufwand der bei der Verwendung starrer Schablonen dadurch entsteht, dass jeder Metadatenautor seine eigenen Metadaten-schablonen zu erstellen hat.

5.1.5 Nutzung bereits bestehender Metadaten

Die Nutzung bereits bestehender Metadaten hat unabhängig vom verwendeten Metadaten-schema bei der Erstellung neuer Metadaten-sätze einen hohen Stellenwert. Das trifft insbesondere dann zu, wenn es keine Möglichkeit gibt, diese auf automatischem Wege zuverlässig zu bestimmen. Zur Nutzung bereits bestehender Metadaten müssen diese jedoch oft zuvor in das Format des Zielmetadaten-schemas konvertiert werden. Besonders aufwändig wird das, wenn die einzelnen Datenfelder des Quelldaten-satzes anders definiert sind als das Datenfeld des Zieldaten-satzes. Gehen die Definition der Datenfelder so weit auseinander, dass das direkte Übertragen von Werten nicht mehr möglich ist, können unter Umständen analog zu den in Abschnitt 5.1.3 beschriebenen Wechselwirkungen zwischen Datenfeldern die Werte des Zieldaten-satzes nur dazu verwendet werden, den Wertebereich des Datenfelds im Zieldaten-satz einzuschränken.

Für die Verwendung bereits bestehender Metadaten stehen verschiedene Quellen zur Verfügung. Werden die Lernobjekte aus einem System entnommen, in dem sie bereits mit Metadaten beschrieben worden sind, kann die Metadatenbeschreibung des Quellsystems dazu verwendet werden einen neuen Metadaten-satz zu erstellen, falls dem Metadaten-satz des Quellsystems nicht das Datenschema zugrunde liegt,

das im Zielsystem verwendet wird. Für die Konvertierung von Metadatensätzen zwischen wichtigen Metadatenschemata sind hierfür bereits Anleitungen für deren Konvertierung verfügbar [LOM02]. Eine andere wichtige Quelle für bereits existierende Metadaten sind Informationen, die oft in den physikalischen Daten der Lernobjekte selbst kodiert sind. Hier lassen sich beispielsweise Angaben über die Größe von Bildern finden. Eine wichtig Quelle für bereits bestehende Metadaten ist beispielsweise die Beschreibung von Web-Seiten nach Dublin Core. In [Kun99] sind Meta-Tags definiert worden, mit deren Hilfe Metadaten direkt in den entsprechenden HTML-Seiten gespeichert werden können.

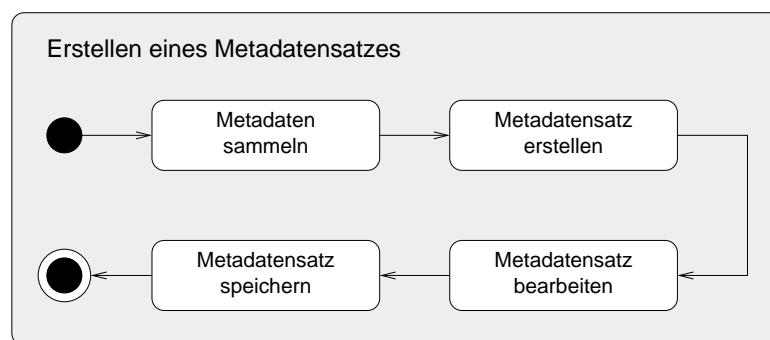


Abbildung 5.4: Bereits bestehende Metadaten bei der Erstellung von Metadatensätzen nutzen

Das Anwendungsgebiet für die Nutzung bereits bestehender Metadaten ist in der Regel die Erstellung neuer Metadatensätze. Dabei basieren die neu erstellten Metadatensätze auf den verfügbaren Metadaten. Abbildung 5.4 zeigt ein Aktivitätsdiagramm für das Erstellen von Metadatensätzen. Im ersten Schritt zum Erstellen eines Metadatensatzes auf Basis bestehender Metadaten werden die verfügbaren und nutzbaren Metadaten zusammengetragen. Hierfür werden sowohl extern verfügbare Metadaten, als auch in den physikalischen Daten der Lernobjekte vorliegenden Attribute verwendet. Auf Basis dieser Daten erfolgt im zweiten Schritt die Erstellung eines neuen Metadatensatzes. Diese kann im dritten Schritt durch den Metadatenautoren bearbeitet werden. Wenn es nicht erforderlich ist den auf diesem Wege erstellten Metadatensatz zu ändern oder zu ergänzen, kann dieser Arbeitsschritt entfallen. Im letzten Arbeitsschritt wird der neu erstellte Metadatensatz abgespeichert.

5.1.6 Gewinnung von Metadaten durch Inhaltsanalyse

Durch eine Analyse der Lernobjekte können inhaltsbezogene Metadaten gewonnen werden. Die Inhaltsanalyse ist in der Regel ein komplexer Vorgang, der auf die Eigenschaften der verschiedenen Typen der Lernobjekte angepasst werden muss. Ziel

der Inhaltsanalyse ist die Extraktion inhaltsbezogener Metadaten. Zu den inhaltsbezogenen Metadaten, die mittels der Inhaltsanalyse bestimmt werden sollen, gehören beispielsweise Titel und Beschreibung der Lernobjekte. Im Folgenden werden Ansätze und Möglichkeiten der Inhaltsanalyse beschrieben, die für die Generierung von Metadaten geeignet sind, ohne dabei einen Anspruch auf Vollständigkeit zu erheben.

Die Verfahren für die Inhaltsanalyse lassen sich analog zu den Datenformaten der Lernobjekte in Analyseverfahren für statische und kontinuierliche Verfahren gruppieren. Verfahren zur Analyse statischer Lernobjekte, wie Text und HTML, stellen in der Regel eine Verknüpfung verschiedener Ansätze zur Inhaltsanalyse dar. Hierbei werden insbesondere die Struktur und Textauszeichnung der Dokumente untersucht. Informationen, die aus diesen Merkmalen gezogen werden können, dienen zusätzlich oft dazu Ergebnisse aus der Verarbeitung natürlicher Sprache oder anderer statistischer Analysen zu bewerten. In dem in [JJBW99] beschriebenen Verfahren werden im Text Übereinstimmungen mit den Begriffen des Dewey Decimal Classification (DDC) Systems [MBMN96] gesucht. Diese werden dann mittels der Informationen aus der Dokumentstruktur gewichtet. Für die Inhaltsanalyse anderer statischer Lernobjekte, wie beispielsweise Bilder müssen andere Verfahren herangezogen werden. In [Vol97] wird ein Verfahren beschrieben, mittels dessen relevante Bildinformationen extrahiert werden können. Mit Hilfe dieser Informationen lassen sich beispielsweise ähnliche Bilder in Datenbanken finden. Ein anderer Ansatz zur Metadatengenerierung für Bilder ist das Finden und Erkennen von Gesichtern in Bildern [RBK96, TP91]. Der Analyse kontinuierlicher Lernobjekte, wie beispielsweise Audio oder Video, geht in der Regel eine Segmentierung voraus. Mit Hilfe von Hidden-Markov-Modellen (HMM) [Rab89] werden einzelne Sprachsegmente erkannt. Anschließend erfolgt die Auswertung der Texte durch Verarbeitung natürlicher Sprache. Lernobjekte vom Typ Video werden an Hand von automatischer Schnitterkennung segmentiert. Ebenso wie bei Standbildern liefert die Gesichtserkennung in Bewegtbildern [YW96] wichtige Informationen über den Inhalt.

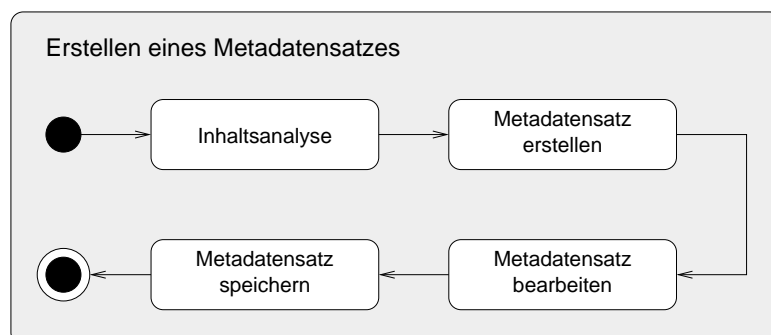


Abbildung 5.5: Bestimmung von Metadaten durch Inhaltsanalyse

Abbildung 5.5 zeigt das Aktivitätsdiagramm zur Erstellung von Metadaten mit

Unterstützung durch eine automatische Inhaltsanalyse. Im ersten Arbeitsschritt werden die Lernobjekte mittels der erforderlichen Verfahren analysiert, mit dem Ziel, so viele benötigte Metadaten wie möglich automatisch aus den Lernobjekten zu extrahieren. Im zweiten Schritt wird basierend auf den automatisch bestimmten Metadaten ein neuer Metadatensatz erstellt. Dieser kann anschließend im dritten Arbeitsschritt unter Verwendung eines Metadateneditors bearbeitet und ergänzt werden. Das ist erforderlich, da die automatisch bestimmten Metadaten häufig nachbearbeitet werden müssen, weil Verfahren zur Inhaltsanalyse nicht in allen Fällen fehlerfreie Ergebnisse erzielen.

5.1.7 Systembedingte Metadaten

Manche Metadatenfelder sind so definiert, dass es nicht sinnvoll ist, dass die Werte von Hand eingegeben werden. Für diese Datenfelder werden die entsprechenden Werte durch systembedingte Metadaten bestimmt. Systembedingte Metadaten sind in diesem Fall Informationen über die mit Metadaten zu beschreibenden Lernobjekte, die hauptsächlich von Systemen zur Verarbeitung von Lernobjekten, wie Lernobjektarchiven, Autorensystemen und Lernsystemen, genutzt werden und ebenso von diesen Systemen bestimmt werden können. Ein Beispiel für systemabhängige Metadaten sind Identifikatoren von Lernobjekten und deren zugehörigen Metadatensätzen. Neben unnötigem Zusatzaufwand, diese Metadaten von Hand zu erfassen, ist deren manuelle Erfassung in der Regel zu fehleranfällig. Gerade bei der maschinellen Verarbeitung von Metadaten sind Eingabefehler in der Regel problematisch. Aus diesem Grund ist die manuelle Modifikation dieser Werte in der Regel weder notwendig noch erwünscht.

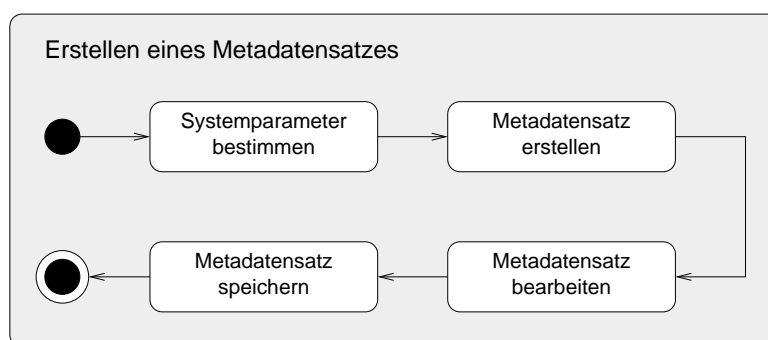


Abbildung 5.6: Metadatenerstellung auf Basis systembedingter Metadaten

Systembedingte Metadaten sind im Prinzip Metadaten aus dem Systemkontext von Lernobjekten. Dementsprechend gestaltet sich das Erstellen von Metadatensätzen auf Basis systembedingter Metadaten analog zur Erstellung von Metadatensätzen auf Basis von Kontextinformationen. Nach dem Aktivitätsdiagramm in

Abbildung 5.6 werden im ersten Arbeitsschritt die systembedingten Metadaten bestimmt. Diese werden im zweiten Schritt verwendet, um auf deren Basis einen neuen Metadatensatz zu erstellen. Im dritten Arbeitsschritt wird der neu erstellte Metadatensatz im Metadateneditor zur Ergänzung nicht systembedingter Metadaten dem Metadatenautoren zur Bearbeitung angeboten. Dabei sind die systembedingten Metadaten schreibgeschützt, so dass sie nicht modifiziert werden können. Alternativ dazu ist es möglich die systembedingten Metadaten gar nicht im Metadateneditor anzuzeigen. Im letzten Arbeitsschritt wird der neu erstellte Metadatensatz abgespeichert.

5.1.8 Ableiten von Metadatensätzen

Das Ableiten von kompletten Metadatensätzen aus bereits bestehenden kommt vor allem dann zur Anwendung, wenn bestehende, durch Metadaten zu beschreibende, Lernobjekte kopiert und leicht verändert werden. Diese Anwendung ist beim Reauthoring, dem Wiederverwenden in modifizierter Variante, von Lernobjekten sehr häufig. Für die neuen leicht modifizierten Lernobjekte werden zusätzlich neue Metadatensätze benötigt, die sie beschreiben. Da die Lernobjekte nur leicht verändert werden kann davon ausgegangen werden, dass der Metadatensatz des neuen, leicht veränderten Lernobjekts sich nur geringfügig von dem Metadatensatz unterscheidet, der das ursprüngliche Lernobjekt beschreibt. Aus diesem Grund wird für die Erstellung des Metadatensatzes, der das neue, leicht modifizierte Lernobjekt beschreibt, der Metadatensatz des ursprünglichen Lernobjekts als Basis verwendet. Der neu zu erstellende Metadatensatz wird gewissermaßen von dem alten Metadatensatz abgeleitet.

Beim Ableiten von Metadatensätzen können alle zuvor genannten Methoden zur automatischen Bestimmung von Metadaten eingesetzt werden. Metadaten, die nicht automatisch bestimmt werden können, werden von dem Metadatensatz des ursprünglichen Lernobjekts kopiert. Diese kopierten Werte stellen bei diesem Anwendungsszenario Standardwerte dar, die im Nachhinein vom Benutzer bei Bedarf geändert werden können.

Das Ableiten von Metadatensätzen ist streng genommen kein Verfahren mit dessen Hilfe Metadaten generiert werden können, weil durch das Ableiten von Metadaten keine zusätzlichen neuen Metadaten gewonnen werden. Das Ableiten von Metadaten stellt lediglich das Zusammenfassen von Metadaten aus verschiedenen Quellen dar, mittels derer ein neuer Metadatensatz erstellt werden soll. Da das Ableiten von Metadaten jedoch eine wichtige Bedeutung für die Wiederverwendung von Lernobjekten in modifizierte Form darstellt, wurde es hier erwähnt.

5.1.9 Bewertung

In den vorherigen Abschnitten sind eine Reihe grundlegender Verfahren zur manuellen und automatischen Erstellung von Metadaten vorgestellt worden. In diesem Abschnitt erfolgt eine Bewertung der einzelnen Verfahren nach eigenen Erfahrungen, wie sie im Projekt k-MED [15] von den medizinischen Fachautoren und bei der Entwicklung des ResourceCenters, vergleiche Kapitel 7, gemacht wurden. Das gewählte Schema bewertet die einzelnen Verfahren qualitativ hinsichtlich der Hilfe die durch die Anwendung bei den Autoren entsteht, der Aufwand für die technische Realisierbarkeit, sowie der Zuverlässigkeit der generierten Metadaten. Für die Bewertung der Merkmale wird ein fünfstufiger Wertebereich verwendet.

Tabelle 5.1: Bewertung der Methoden für die Erstellung von Metadaten

Hilfe Autor	Aufwand Technik	Zuverlässigkeit	Verfahren
--	--	++	Manuelle Erfassung
+	--	+	Metadatenschablonen
+	-	++	Abhängigkeiten
+ / +++	-	+ / +++	Kontextinformationen
++	-	+	Existierende Metadaten
O	++	O	Inhaltsanalyse
++	--	++	Systembedingte Metadaten
++	--	O / +	Ableiten von Metadaten

++ sehr hoch, + hoch, O neutral, - niedrig, -- sehr niedrig

Tabelle 5.1 listet die Bewertungsergebnisse der einzelnen Methoden für die Generierung von Metadaten auf. Gute Methoden für die Generierung von Metadaten zeichnen sich dadurch aus, dass sie eine sehr große Hilfe für Autoren darstellen, ein sehr niedriger technischer Aufwand für deren Realisierung erforderlich ist und die generierten Metadaten mit sehr hoher Zuverlässigkeit richtig sind. Diese Eigenschaften treffen auf die Generierung systembedingter Metadaten zu. Ihre Generierung kann ohne zusätzliche Interaktion mit den Autoren erstellt werden, der Aufwand für die technische Realisierung ist sehr gering und die generierten Metadaten sind mit sehr hoher Zuverlässigkeit richtig. Gerade der letzte Punkt ist bei systembedingten Metadaten besonders wichtig.

Unter der Annahme, dass Autoren beim Erstellen von Metadaten keine Fehler machen, stellt die manuelle Erfassung von Metadaten unter den Gesichtspunkten Zuverlässigkeit und Aufwand für die technische Realisierung ein gutes Verfahren

für die Erstellung von Metadaten dar. Bei der manuellen Erfassung von Metadaten ohne Verwendung von Metadaten-schablonen ist die Hilfe beim Erstellen von Metadaten jedoch sehr niedrig. Die manuelle Erfassung von Metadaten ohne die Verwendung von Schablonen sollte also als letztes Mittel zum Erstellen von Metadaten herangezogen werden, wenn kein anderes Verfahren greift.

Die Verwendung von Metadaten-schablonen erleichtert das Erstellen von Metadaten deutlich. Da Metadaten-schablonen zunächst selbst erstellt werden müssen und oft Vorgabewerte enthalten, die angepasst werden müssen, gibt es ein Fehlerpotential durch beispielsweise vergessene Anpassungen. Metadaten-schablonen werden deswegen sowohl bei der Hilfe für Autoren, als auch der Zuverlässigkeit nur mit hoch bewertet. Ebenso wird die Hilfe bei der Erstellung von Metadaten durch die Verwendung von Abhängigkeiten zwischen Datenfeldern im Metadaten-schema nicht als sehr hoch bewertet. Der Grund hierfür liegt darin, dass zunächst die Werte bestimmt werden müssen, die als Grundlage für die Bestimmung weiterer Werte dienen. Neben dem Aufwand, Abhängigkeiten im Datenschema zu finden, erfordert die technische Realisierung einen höheren Aufwand als die eines Metadateneditors zur manuellen Erfassung der Metadaten oder die Anwendung von Metadaten-schablonen. Deshalb wird der Aufwand für die technische Realisierbarkeit nur als niedrig bewertet.

Die Verwendung von Kontextinformationen beim Erstellen von Metadaten stellt eine Methode für die Erstellung von Metadaten dar, mittels der mit niedrigem Aufwand Metadaten mit hoher Zuverlässigkeit erstellt werden können. Für den Fall, dass Werte für Datenfelder prognostiziert werden, muss die Bewertung für die Hilfe und Zuverlässigkeit reduziert werden, da die prognostizierten Datenwerte geändert werden müssen, falls sie nicht zutreffen.

Unter der Annahme, dass bestehende Metadaten ohne Änderung übernommen werden können, ist die Verwendung bereits bestehender Metadaten ebenfalls eine gute Methode, Metadaten zu bestimmen. Die Schwierigkeit bei der technischen Realisierung dieser Methode besteht darin, für alle vorkommenden Metadaten-formate eine Import- und Konvertierungsfunktion bereitzustellen. Im Gegensatz zur Verwendung bereits bestehender Metadaten wird das Ableiten von Metadaten für den Fall eingesetzt in dem Lernobjekte in modifizierter Variante wiederverwendet werden sollen. Das Ableiten von Metadaten stellt in diesem Fall ebenfalls ein sehr große Hilfe für die Metadatenautoren dar, die sich mit sehr geringem Aufwand technisch realisieren lässt. Je nach verwendetem Metadaten-schema und kombinierten Methoden für die automatische Generierung der Metadaten, die sich geändert haben, ist die Zuverlässigkeit nur mit neutral bis gut zu bewerten.

Bei der Inhaltsanalyse zeigt sich, dass je nach Medientyp und gewähltem Algorithmus die Bestimmung der Metadaten im Vergleich zu den anderen Verfahren oft durch ein schlechtes Verhältnis zwischen Kosten und Nutzen gekennzeichnet ist. Das liegt an dem oft hohen Aufwand für die Entwicklung und Bereitstellung von Verfahren für die Inhaltsanalyse und deren Unzuverlässigkeit der Ergebnisse, wo-

durch eine manuelle Nachbearbeitung der Metadaten erforderlich wird. Die Hilfe für Autoren und die Zuverlässigkeit der generierten Metadaten werden aus diesem Grund mit neutral bewertet.

5.2 Generierung von Metadaten für Lernobjekte

In diesem Abschnitt wird ein Verfahren für die semiautomatische Erstellung von Metadaten beschrieben, das in Autorenprozessen, die auf Aggregation von Lernobjekten basieren, zur Generierung der Metadaten verwendet werden kann. Das Verfahren stellt eine Kombination der in den vorherigen Abschnitten diskutierten Ansätze für die Erfassung beziehungsweise Generierung von Metadaten dar. Hierfür werden zunächst in einer Bedarfsanalyse die obligatorischen Datenfelder des verwendeten Metadatenschemas identifiziert. Anschließend erfolgt die Bestimmung geeigneter Verfahren für die Generierung der Metadaten. Mit Hilfe der Ergebnisse aus der Bewertung der Metadatengenerierungsverfahren, werden die geeigneten Verfahren priorisiert. Das Ergebnis dieser Betrachtungen ist ein Prozess für die semiautomatische Erstellung von Metadaten.

5.2.1 Metadatenschema

Als Zielformat für die generierten Metadaten wird der Metadatenstandard Learning Object Metadata (LOM) [LOM02] des Learning Technology Standards Committee (LTSC) der IEEE verwendet. Für die Wahl von LOM gibt es verschiedene Gründe. Der wichtigste Grund für Verwendung von LOM ist, dass LOM sehr weit verbreitet ist und das Basisschema beziehungsweise die Obermenge eingeschränkter Metadatenschemata, wie beispielsweise CanCore [4] ist. Für das ebenfalls sehr wichtige Metadatenschema Dublin Core [DC003] existieren Abbildungsvorschriften, um Metadaten zwischen den beiden Metadatenformaten konvertieren zu können. Der wichtigste Grund für die Verwendung ist der, dass LOM Teil der SCORM-Spezifikation ist, und aufgrund der Verwendung von LOM zur Beschreibung von Lernobjekten die Metadaten nicht in eine andere Darstellung konvertiert werden müssen, wenn sie in ein SCORM-konformes Kurspaket importiert werden sollen.

Das LOM-Datenschema ist ein sehr komplexes Metadatenschema, das speziell für die Beschreibung von Lernobjekten entwickelt wurde. Es bietet die Möglichkeit der Beschreibung verschiedener Merkmale eines Lernobjekts. LOM ist ein stark strukturiertes und wohl definiertes Metadatenschema. Zusammengehörige Datenfelder, die thematisch verwandt sind, werden in Kategorien gruppiert. Der LOM-Standard definiert die neun Kategorien „General“, „Lifecycle“, „Meta-metadata“, „Technical“, „Educational“, „Rights“, „Relation“, „Annotation“ und „Classification“. Während die ersten sechs genannten Kategorien nur maximal einmal im Metadatensatz definiert sind, ist jeweils eine uneingeschränkte Anzahl der letzten drei Kategorien in

einem Metadatensatz erlaubt. Vereinzelt Datenfelder dieser Kategorien enthalten selbst weitere Elemente und dienen so der Gruppierung von Datenfeldern.

5.2.2 Obligatorische Datenfelder

Eine Schwäche des LOM-Standards ist die fehlende Definition obligatorischer Datenfelder, die zwingend ausgefüllt werden müssen. In der SCORM-Spezifikation werden deswegen zur Verbesserung der Interoperabilität der enthaltenen LOM-Datensätze ein Teil der im LOM-Standard definierten Datenfelder als obligatorisch definiert. Die Menge der obligatorischen Datenfelder hängt vom Typ des Lernobjekts ab. SCORM definiert die Typen „ASSET“, „SCO“ und „Content Package“. Lernobjekte der Typen SCO und Content Package besitzen dieselbe Menge obligatorischer Metadatenfelder.

Tabelle 5.2: Obligatorische Datenfelder nach SCORM 1.2

Asset	SCO	Kategorie	Datenfeld
x	x	General	Title
	x		Catalogentry
x	x		Description
	x		Keyword
	x	Lifecycle	Version
	x		Status
x	x	Meta-metadata	Metadatascheme
x	x	Technical	Format
x	x		Location
x	x	Rights	Cost
x	x		Copyrights and other Restrictions
	x	Classification	Purpose
	x		Description
	x		Keyword

x obligatorisches Datenfeld

Tabelle 5.2 listet alle Datenfelder auf, die nach der SCORM-Spezifikation in Version 1.2 als obligatorisch definiert sind. Dabei sind alle genannten Datenfelder für Lernobjekttypen SCO und dementsprechend Content Packages obligatorisch, während nur eine Auswahl der genannten Datenfelder für die Lernobjekttypen Asset obligatorisch sind. Da SCORM zwar den Metadatenstandard LOM verwendet, es

für diesen bis zur Entwicklung der Version 1.2 noch keine Abbildung nach XML gab, wird in der SCORM-Spezifikation auf die XML-Abbildung des nahezu identischen Metadatenschemas Learning Resource Metadata (LRM) [IMS01b] des IMS Global Learning Consortiums (IMS) zurückgegriffen. In der Tabelle werden aus diesem Grund die Namen der IMS-Datenfelder verwendet.

Da SCORM als Ausgabeformat für die im Autorenprozess erstellten Kurse vorgesehen ist und deswegen die erstellten Metadaten in die SCORM-Pakete integriert werden müssen, wird die Menge der nach SCORM 1.2 obligatorischen Datenfelder auch als Mindestmenge für die Generierung von Metadatensätzen im Metadaten-Wizard definiert.

Die Liste der obligatorischen Datenfelder enthält hauptsächlich Metadaten, die für eine spätere Beurteilung, Auswahl, Verwendung und Klassifikation von Lernobjekten notwendig sind. Darüber hinaus werden Metadaten gesammelt, mit deren Hilfe die Klärung von Nutzungsrechten erfolgt und die Aufschluss über die Kosten geben, die durch die Nutzung von Lernobjekten entstehen.

Zusätzlich zu den in der Scorm-Spezifikation als obligatorisch definierten Datenfeldern werden weitere Datenfelder in die Liste der obligatorischen Datenfelder aufgenommen. Hierzu gehört das Datenfeld LOM.Educational.LearningResourceType¹. Es kann bei der Suche nach geeigneten Lernobjekten im Lernobjektarchiv genutzt werden, um zwischen Typen von Lernobjekten zu differenzieren. So kann zwar mittels des Mime-Types, der im obligatorischen Datenfeld LOM.Technical.Format des Metadatensatzes eines Lernobjekts gespeichert wird, das Dateiformat bestimmt werden, hierüber lässt sich jedoch nicht weiter differenzieren, ob in einer Flash-Datei ein Bild oder eine Simulation gespeichert ist. Mit Hilfe der Erfassung des Ressourcentyps über das Datenfeld LOM.Educational.LearningResourceType wird die weitere Differenzierung zwischen verschiedenen Typen von Lernobjekten möglich. Aus diesem Grund wird das Datenfeld LOM.Educational.LearningResourceType in die Menge der obligatorischen Datenfelder aufgenommen.

Darüber hinaus werden im Metadaten-Wizard auch Informationen über die Autorenschaft von Metadatensätzen und Lernobjekten in die Menge der als obligatorisch zu erfassenden Metadaten aufgenommen. Diese Informationen dienen einer gewünschten Kontaktaufnahme des Autoren eines bestimmten Lernobjekts für den Fall, dass beispielsweise Fehler gemeldet werden sollen. Die Kontaktinformationen können aber auch leicht für den Interessenaustausch und Austausch von Quellen verwendet werden, um Lernobjekte weiterbearbeiten zu können. Hierfür werden die Datenfelder LOM.LifeCycle.Contribute und LOM.MetaMetadata.Contribute in die Liste der obligatorischen Felder aufgenommen.

Für die weitere Arbeit wird die Liste der durch die SCORM-Spezifikation in

¹Die Datenfelder LOM.Educational.LearningResourceType, LOM.LifeCycle.Contribute und LOM.MetaMetadata.Contribute werden in der SCORM-Spezifikation nicht als obligatorische Datenfelder bezeichnet. Sie werden nicht in Tabelle 5.2 aufgelistet, weil die Tabelle nur die obligatorischen Datenfelder der SCORM-Spezifikation enthält

Version 1.2 definierten Datenfelder der LOM-Spezifikation unter Erweiterung der Datenfelder LOM.Educational.LearningResourceType, LOM.LifeCycle.Contribute und LOM.MetaMetadata.Contribute als Menge obligatorischer Datenfelder definiert, die bei der Generierung von Metadatensätzen mindestens zu bestimmen beziehungsweise zu erfassen sind.

5.2.3 Semiautomatische Metadatengenerierung

Im vorherigen Kapitel ist eine Menge von obligatorischen Metadatenfeldern definiert worden, die in jedem Metadatensatz für jedes neu erstellte Lernobjekt generiert beziehungsweise erfasst werden müssen. Ziel der Metadatengenerierung ist es Metadatensätze zu erstellen, die mindestens die obligatorischen Metadatenfelder enthalten.

Eine Reihe von Verfahren und Methoden zur Erfassung und Bestimmung von Metadaten sowohl auf manuellem Weg als auch auf automatischer Weise sind in den vorangehenden Abschnitten vorgestellt worden. Keines der genannten Verfahren zur automatischen Generierung von Metadaten ist jedoch geeignet die Werte aller obligatorischen Metadatenfelder zu bestimmen. Für die vollständige Erstellung von Metadatensätzen ist deshalb die Anwendung einer Kombination verschiedener Verfahren erforderlich. Da es sich bei dem Ableiten von Metadatensätzen bereits um die Kombination verschiedener Verfahren handelt, wird das Ableiten von Metadaten nicht als gesondertes Verfahren in die Betrachtung eingeschlossen. Für den Fall, dass das Ableiten von Metadatensätzen erforderlich wird, kann dies durch den kombinierten Einsatz der anderen Methoden zur Metadatengenerierung erreicht werden.

In Tabelle 5.3 werden den einzelnen obligatorischen Datenfeldern geeignete Verfahren für deren Bestimmung zugeordnet. Mit Hilfe der manuellen Erfassung können obligatorischen Datenfelder bestimmt werden. Unter der Voraussetzung, dass alle obligatorischen Metadatenfelder in bereits bestehenden Metadaten enthalten sind, lassen sich ebenfalls alle Metadatenfelder mit Hilfe einer Konvertierung existierender Metadaten bestimmen. In der Regel reichen bereits bestehende Metadaten jedoch nicht aus, um alle Werte obligatorischer Datenfelder zu bestimmen. Ein prominentes Beispiel für Metadatensätze mittels, derer sich nicht alle obligatorischen Datenfelder bestimmen lassen, sind die Metadaten in Dokumenten, die mit Werkzeugen aus der Microsoft Office Produktfamilie erstellt wurden. In vielen Fällen liegen ohnehin keine bestehenden Metadaten vor.

Tabelle 5.3: Geeignete Methoden für die Bestimmung der Werte obligatorischer Metadatenfelder

Kategorie	Datenfeld	Manuelle Erfassung	Metadaten-schablonen	Abhängigkeiten	Kontextinformationen	Existierende Metadaten ¹	Inhaltsanalyse	Systembedingte Metadaten
General	Title	x				x	x	
	Catalog Entry	x				x		x
	Description	x				x	x	
	Keyword	x				x	x	
Lifecycle	Version	x				x		x
	Status	x				x		x
	Contribute	x	x		x	x		
Meta Metadata	Catalog Entry	x				x		x
	Contribute	x	x		x	x		
	Metadata Scheme	x	x			x		x
Technical	Format	x	x	x ²		x		x
	Location	x				x		x
Educational	Learning Resource Type	x		x ²		x		
Rights	Cost	x	x			x		
	Copyrights and other Restrictions	x	x			x		
Classification	Purpose	x	x			x	x	
	Description	x	x			x	x	
	Keyword	x	x			x	x	

¹ Die Angaben setzen voraus, dass der Quelldatensatz alle obligatorischen Datenfelder unterstützt und enthält.

² Zwischen den Werten der Datenfelder LOM.Technical.Format und LOM.Educational.LearningResourceType besteht eine Abhängigkeit.

Mit Metadaten-schablonen lassen sich vor allem die Metadatenfelder mit Werten belegen, die bei einer größeren Menge von Lernobjekten gleich sind, so dass sie als Standardwerte in den Metadaten-satz eingesetzt werden können. Zwischen den beiden Datenfeldern LOM.Technical.Format und LOM.Educational.LearningResourceType besteht eine Abhängigkeit, die bei der Generierung von Metadaten genutzt werden kann. Während LOM.Technical.Format sich mit wenig Aufwand automatisch bestimmen lässt, ist die Bestimmung von LOM.Educational.LearningResourceType nur mit vergleichsweise hohem Aufwand manuell oder durch Inhaltsanalyse zu bestimmen. Auf Basis der Abhängigkeit dieser beiden Datenfelder lässt sich jedoch leider keine eindeutige Abbildung finden mittels der Wert für LOM.Educational.LearningResourceType auf Basis des Wertes von LOM.Technical.Format bestimmt werden kann. Aus diesem Grund eignet sich die Abhängigkeit lediglich zur Einschränkung des Wertebereichs für LOM.Educational.LearningResourceType.

Kontextinformationen, wie etwa das Benutzerprofil können dazu genutzt werden, den Autoren des Metadaten-satzes und des Lernobjekts in den Metadaten einzutragen. Mittels Inhaltsanalyse lassen sich die Metadatenfelder zur Speicherung von Titel, Beschreibung und Schlüsselwörter automatisch bestimmen. Auf Basis eines eingeschränkten Wertebereichs ist es ebenfalls denkbar den Wert des Datenfelds LOM.Educational.LearningResourceType zu bestimmen. Die obligatorischen Metadatenfelder der Kategorie Classification lassen sich ebenfalls mittels Inhaltsanalyse bestimmen. Ein geeigneter Ansatz hierfür wird in [JJBW98] beschrieben. Zu den Datenfeldern, die sich über systembedingte Metadaten bestimmen lassen gehören Datenfelder mit den Identifikatoren für Lernobjekt und Metadaten-satz, Version, Status, Format und Lokation des Lernobjekts.

Tabelle 5.3 zeigt für jedes der obligatorischen Datenfelder mehrere Alternativen für die Generierung ihrer Werte. Gibt es für die Bestimmung eines Datenwertes mehrere Alternativen, dann muss eine Methode für die Bestimmung des Datenfeldes gewählt werden. Bei einer fixen Auswahl der Methoden für die Bestimmung von Datenwerten sind jedoch die Methoden problematisch, durch deren Anwendung nicht immer Datenwerte bestimmt werden können. Beispiele hierfür sind die Generierung von Metadaten auf Basis bestehender Metadaten oder die Nutzung von Abhängigkeiten zwischen Datenfeldern. Aus diesem Grund werden die Methoden für die Generierung von Metadaten priorisiert, damit geeignete Methoden flexibel unter Verwendung des Priorisierungsschemas ausgewählt werden können. Hohe Prioritäten erhalten dabei Methoden, die ein Maximum an Arbeitserleichterung für die Metadatenautoren mitsichbringt und dabei dennoch hohe Zuverlässigkeit bei der Generierung von Metadaten bieten.

Für die Priorisierung der Methoden zur Generierung von Metadaten wird die Bewertung aus Abschnitt 5.1.9 herangezogen. Als Kriterien werden für die Priorisierung der Methoden die Arbeitserleichterung, die durch die Anwendung des Verfahrens bei den Metadatenautoren zu erwarten ist, und die Zuverlässigkeit, mit der die Metadaten bestimmt werden können, verwendet. In erster Linie wird die

Tabelle 5.4: Priorisierung der Methoden für die Metadatenerstellung

Priorität	Verfahren	Manuelle Nachbearbeitung
1	Systembedingte Metadaten	
2	Existierende Metadaten	
3	Kontextinformationen	
4	Abhängigkeiten	x ¹
5	Metadatenschablonen	x
6	Inhaltsanalyse	x
7	Manuelle Erfassung	

¹ Manuelle Nachbearbeitung nur bei eingeschränkter Wertemenge notwendig

zu erwartende Arbeitserleichterung für die Priorisierung der Methoden verwendet. Methoden, die gleiche Arbeitserleichterung versprechen, werden nach der Zuverlässigkeit, mit der die Metadaten bestimmt werden können, priorisiert.

Tabelle 5.4 zeigt das Ergebnis der Priorisierung der Methoden. In den drei Spalten der Tabelle werden für jede Methode deren Priorität, Bezeichnung und Informationen darüber, ob eine manuelle Nachbearbeitung der generierten Metadaten erforderlich ist, aufgelistet. Die Priorität der Methoden sinkt mit steigenden Werten in der Spalte Priorität. Die systembedingten Metadaten stellen demnach die Metadaten mit der höchsten Priorität dar. Sie werden gefolgt von den Metadaten, die aus bestehenden Metadatensätzen gewonnen werden können, da diese Methode ebenfalls eine sehr hohe Arbeitserleichterung für die Metadatenautoren verspricht. An dritter Stelle stehen die Kontextinformationen, die hohe bis sehr hohe Hilfe bei der Erstellung von Metadatensätzen bieten. Darauf folgen in der Reihenfolge die Generierung von Metadaten unter Verwendung von Abhängigkeiten im Metadatenschema und der Verwendung von Metadatenschablonen. Bei der Verwendung von Abhängigkeiten im Metadatenschema ist eine manuelle Nachbearbeitung der Metadaten nur dann erforderlich, wenn es keine eindeutige Abbildung gibt, mittels der Werte eines Metadatenfeldes auf Basis von Werten anderer Metadatenfelder bestimmt werden können. Bei der Verwendung von Metadatenschablonen sollte eine manuelle Nachbearbeitung der Metadaten erfolgen, weil hierdurch Vorgabewerte, die nicht für alle Fälle Gültigkeit besitzen, manuell geändert werden können. Danach folgt die Generierung von Metadaten mittels Inhaltsanalyse. Da bei dieser Methode je nach eingesetztem Verfahren häufig unzuverlässige Metadaten erzeugt werden, ist ebenfalls eine manuelle Nachbearbeitung der generierten Metadaten er-

forderlich. Als Methode mit der niedrigsten Priorität folgt die manuelle Erfassung der Metadaten.

Werden alle betrachteten Methoden in Kombination angewandt, so handelt es sich bei dem resultierenden Verfahren um ein semiautomatisches Verfahren zur Erstellung von Metadaten, da für die Metadatengenerierung sowohl manuelle als auch automatische Verfahren zum Einsatz kommen.

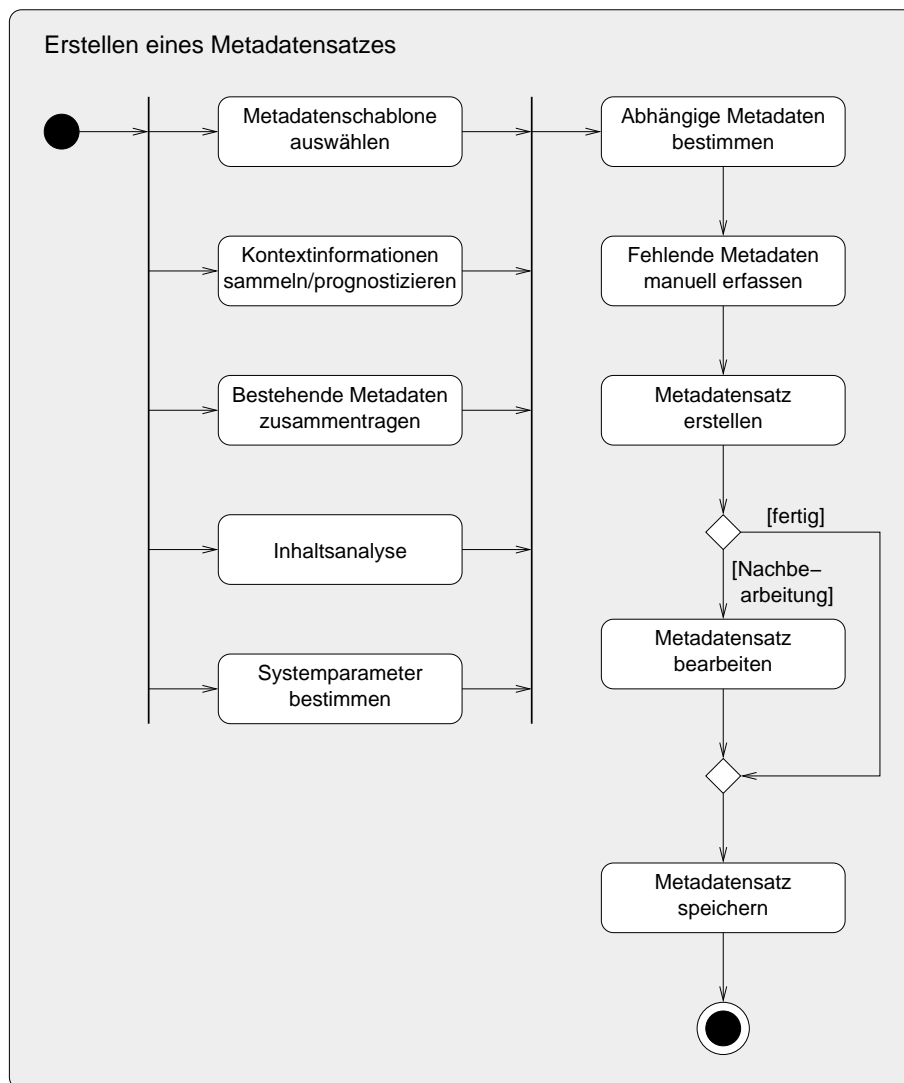


Abbildung 5.7: Semiautomatische Generierung von Metadaten

Abbildung 5.7 zeigt in einem Aktivitätsdiagramm die einzelnen Aktivitäten für die Generierung von Metadaten. Im ersten Schritt werden mittels Metadaten-schablonen, Kontextinformationen, bereits bestehender Metadaten, Inhaltsanalyse und Systemparameter aller verfügbaren Metadaten gesammelt. Im zweiten Schritt wer-

den auf Basis der gesammelten Metadaten abhängige Metadaten bestimmt. Anschließend werden alle noch fehlenden Metadaten manuell erfasst. Metadatenfelder deren Wertebereich durch die Nutzung von Abhängigkeiten im Metadatenschema nur eingeschränkt werden können, werden bei diesem Schritt ebenfalls manuell erfasst. Hierbei werden die Autoren soweit vorhanden mit Hilfe eines beschränkten Wertebereichs unterstützt. Danach liegen alle erforderlichen Metadaten vor. Auf Basis des Priorisierungsschemas wird im nächsten Schritt ein Metadatensatz erstellt. Bei Bedarf kann dieser im Metadateneditor manuell nachbearbeitet werden. Im letzten Schritt wird der Metadatensatz gespeichert.

5.3 Integration der Metadatengenerierung in den Autorenprozess

Das Erstellen von Metadaten für neu erstellte Lernobjekte ist ein Teilschritt innerhalb des Autorenprozesses, vergleiche Kapitel 4. Die Metadaten, die parallel mit den neuen Lernobjekten erstellt werden, dienen in erster Linie für das Wiederfinden der erstellten Lernobjekte im Lernobjektarchiv. Damit das Erstellen der Metadaten von den Autoren akzeptiert wird, ist es wichtig, dass die Metadatengenerierung möglichst wenig Arbeitsaufwand mitsichbringt. Aus diesem Grund soll die semi-automatische Metadatengenerierung in den Autorenprozess integriert werden. Mit ihrer Hilfe können mit wenig Aufwand Metadatenätze bei der Erstellung neuer Lernobjekte generiert werden.

Die Integration der Metadatengenerierung in den Autorenprozess erfolgt analog zu dem Aktivitätsdiagramm für das Erstellen neuer Lernobjekte in Abbildung 4.9. Nachdem ein leerer Container für die Aggregation von Lernobjekten erzeugt worden ist, wird ein dazugehöriger Metadatenatz erstellt. Dabei ist die Aktivität „Metadatenatz erstellen“ in Abbildung 4.9 der Stellvertreter für das Aktivitätsdiagramm aus Abbildung 5.7. Dadurch, dass der Metadatenatz erstellt wird, bevor Lernobjekte aggregiert werden, liegen keine Inhalte vor, die für die Inhaltsanalyse genutzt werden können. Außerdem können keine bereits bestehenden Metadaten für die Metadatengenerierung genutzt werden, weil für neu zu erstellende Lernobjekte keine Metadaten vorliegen.

Die Verwendung von Metadaten-schablonen verdient bei der Generierung von Metadaten in diesem Anwendungsfall besondere Aufmerksamkeit. Metadaten-schablonen werden für Lernobjekte mit gleichen Eigenschaften eingesetzt. Lernobjekte mit gleichen Eigenschaften sind in der Regel Lernobjekte desselben Typs. Für die Bearbeitung verschiedener Typen von Lernobjekten gibt es in der Regel verschiedene Editoren, so dass die Auswahl der zutreffenden Metadaten-schablone indirekt über die Wahl des Editors zur Erstellung eines Lernobjekts bestimmten Typs erfolgen kann. Das heißt, dass die auf den Typ des Lernobjekts bezogenen Metadaten,

die mittels Metadaten-Schablonen in die Generierung von Metadaten eingebracht werden, in diesem Anwendungsfall Kontextinformationen darstellen.

Zusätzlich zu den Kontextinformationen, die sich aus dem Lernobjekttyp ergeben, werden die Metadaten mit Daten des Benutzerprofils ergänzt. Name, Adresse und Kontaktinformationen der Autoren können verwendet werden, um die Benutzer als Autor des neu zu erstellenden Lernobjekts und des dazugehörigen Metadatensatzes auszuzeichnen.

Die Generierung von Metadaten für die Aggregation von Lernobjekten basiert nach Abbildung 5.7 im ersten Schritt allein auf systembasierten Parametern und Kontextinformationen. Die Werte der voneinander abhängigen Datenfelder LOM-Technical.Format und LOM.Educational.LearningResourceType können mit Hilfe der Kontextinformationen des jeweiligen Lernobjekttyps bestimmt werden, so dass auch die Bestimmung abhängiger Metadaten im zweiten Schritt bei der Erstellung neuer Lernobjekte durch Aggregation entfällt.

Weitere Vereinfachungen bei der Metadatengenerierung ergeben sich durch die Integration einer reduzierten Eingabemaske für Metadaten in die Editor-Komponente zum Aggregieren der Lernobjekte. Mittels dieser können Metadaten, die nicht automatisch bestimmt werden können, leicht in der Editor-Komponente erfasst und bearbeitet werden. Der Titel eines Abschnitts kann beispielsweise direkt in die Metadaten aufgenommen werden, wenn die verwendete Sprache sowohl für die Lernobjekte als auch die für Metadaten identisch sind. Die Nachbearbeitung von Metadaten kann über die Aktivität „Metadaten bearbeiten“ des Autorenprozesses erfolgen. Durch die Speicherung der Metadaten zum Zeitpunkt der Speicherung des Lernobjekts, wird die Bearbeitung der manuell zu erfassenden Metadaten während des gesamten Autorenprozesses ermöglicht.

Abbildung 5.8 zeigt den Autorenprozess mit Schwerpunkt Metadatengenerierung mit den berücksichtigten Erweiterungen und Veränderungen der Metadatengenerierung. Im ersten Schritt wird das neue Lernobjekt erstellt, das zunächst nur ein leerer Container für die Aggregation von Lernobjekten darstellt. Danach werden verfügbare Kontextinformationen und Systemparameter gesammelt. Auf Basis dieser Metadaten wird im dritten Schritt ein Metadatensatz erstellt, dem zunächst noch die manuell zu erfassenden Metadaten fehlen. Diese können während des Autorenprozesses wahlweise mit Hilfe der vereinfachten Eingabemaske, die nur noch drei Datenfelder umfasst, in der Editor-Komponente oder dem Metadateneditor erfasst werden. Stellvertretend für beide Wege wird im Diagramm die Aktivität „Metadaten bearbeiten“ gewählt. Wenn die Bearbeitung der Lernobjekte und Metadaten abgeschlossen ist, werden sowohl die Lernobjekte als auch die dazugehörigen Metadaten im Lernobjektarchiv gespeichert.

Die Implementierung der Integration des Metadateneditors sowohl in der Autorenumgebung, wie hier vorgestellt, als auch in dem Lernobjektarchiv, wird in den Abschnitten 7.2 und 7.3 beschrieben.

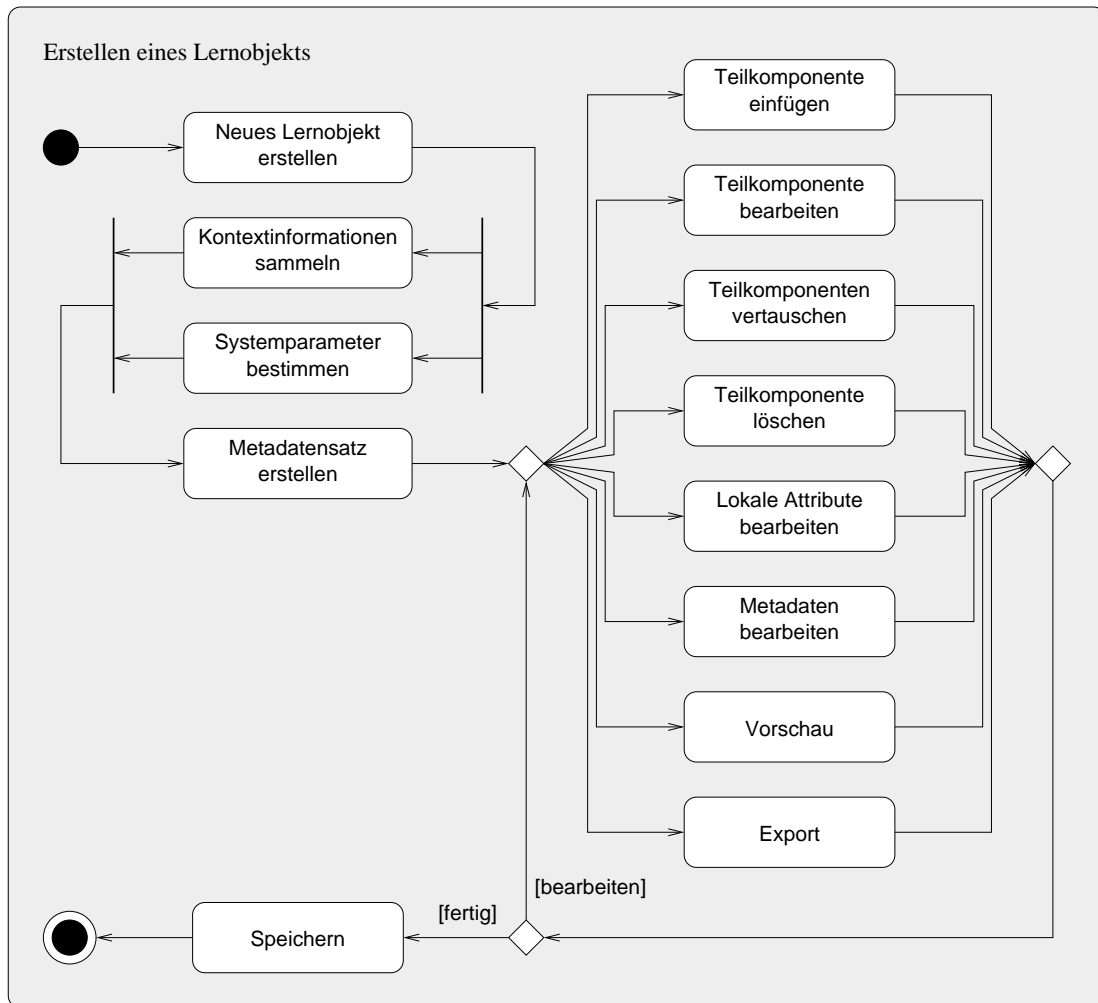


Abbildung 5.8: Metadatengenerierung im Autorenprozess

5.4 Zusammenfassung

Aussagekräftige Metadaten spielen eine Schlüsselrolle bei der Wiederverwendung von Lernobjekten, da sie in der Regel für die Indexierung wiederverwendbarer Lernobjekte herangezogen werden. Die Erstellung von Metadaten ist jedoch in der Regel mit hohem Aufwand verbunden.

In diesem Kapitel wurde deshalb ein Verfahren für die semiautomatische Generierung von Metadaten vorgestellt. Das entwickelte Verfahren ist eine Kombination verschiedener Methoden für die Metadatengenerierung, das insbesondere auf die Erstellung SCORM-konformer Metadaten ausgerichtet ist, weil die durch die SCORM-Spezifikation definierten obligatorischen Datenfelder unterstützt werden. Durch die gezielte Optimierung hinsichtlich der Zuverlässigkeit der generierten Me-

tadaten und der Reduzierung des Arbeitsaufwandes bei der Erstellung von Metadaten, kann die für die Erstellung von Metadaten benötigte Zeit deutlich reduziert werden. Das vorgestellte Verfahren ist somit für die Integration in einem auf Aggregation basierenden Autorenprozess geeignet. In diesem wird das Verfahren begleitend zur Erstellung neuer Lernobjekte und beim Einfügen von Lernobjekten von lokalen Laufwerken angewendet, damit diese von den Autoren mit Metadaten beschrieben und sogleich in einem Lernobjektarchiv für die Wiederverwendung verfügbar gemacht werden. Das Verfahren wird im Metadaten-Wizard als Teil des ResourceCenters implementiert (vgl. Kapitel 7).

6 Versionierung modularisierter Lernobjekte

Die Versionierung modularer Lernobjekte ist notwendigerweise, wie bereits in der Anforderungsanalyse in Abschnitt 4.1 festgestellt, in einem auf Aggregation modularisierter Lernobjekte basierenden Autorenprozess zu realisieren. Mit Hilfe der Versionierung wird Autoren die Möglichkeit gegeben, Lernobjekte und deren Fragmente in neuen Versionen weiterzuentwickeln, dennoch aber auf alte Versionen zurückgreifen zu können [TA03]. Durch sie wird so das Wiederverwenden von Lernobjektfragmenten in modifizierter Form ermöglicht, ohne dass sich deren Modifikation auf alle Lernobjekte überträgt, die die ursprünglichen Lernobjektfragmente verwenden, das heißt durch Aggregation aus diesen gebildet werden.

In diesem Kapitel wird der Ansatz für die Versionierung modularisierter und aggregierter Lernobjekte vorgestellt. Zunächst werden die Anforderungen an die Versionierung von Lernobjekten und deren Fragmente, die sich aus einem auf Aggregation basierenden Autorenprozess ergeben, zusammengetragen. Danach erfolgt eine allgemeine Betrachtung verfügbarer Versionierungsansätze hinsichtlich ihrer Eignung für die Versionierung von Lernobjekten in einem auf Aggregation basierenden Autorenprozess. Anschließend werden Annahmen getroffen, auf deren Basis das Konzept für die Versionierung von Lernobjekten zu entwerfen ist. Das Konzept wird zunächst für Lernobjekte eingeführt, die außer Versionsbeziehungen keine weiteren Beziehungen zu Lernobjekten besitzen. Anschließend werden die Konzepte auf die Versionierung aggregierter Lernobjekte übertragen. Hierbei wird insbesondere die Propagierung von Zuständen neuer und benannter Versionen von Lernobjektfragmenten eines Kurses betrachtet.

6.1 Anforderungen

Für die Versionierung von Lernobjekten gibt es verschiedene Gründe. In einem System, das zugleich die Modifikation und die Wiederverwendung von Lernobjekten ermöglicht, ergeben sich aus den folgenden Anwendungen die wichtigsten Gründe für die Versionierung von Lernobjekten:

- Durch die Bearbeitung eines Medienobjekts, sei es durch die Bearbeitung mit einem integrierten Werkzeug oder einem externen Werkzeug, ist das Erstellen

einer neuen Version eines Lernobjekts erforderlich, wenn der alte Zustand des Lernobjekts weiterhin verfügbar sein soll.

- Die Bearbeitung eines Lernobjektes, das in aggregierten Lernobjekten enthalten ist, macht die Versionierung des Lernobjekts erforderlich, wenn diese Änderungen nicht an allen Stellen, an denen das entsprechende Lernobjekt wiederverwendet wird, wirksam werden sollen.
- Durch die Weiterentwicklung eines Lernobjekts, das von einem aggregierenden Lernobjekt referenziert wird, ergibt sich für das referenzierende Lernobjekt ebenfalls eine Veränderung, wenn das weiterentwickelte Lernobjekt übernommen werden soll. Wenn auf das aggregierende Lernobjekt die beiden zuvor genannten Punkte zutreffen, wird ebenfalls eine Versionierung erforderlich.

Lernobjekte werden erstellt, um zwischen Autoren ausgetauscht und in verschiedenen Lernumgebungen eingesetzt werden zu können. Veränderungen an Lernobjekten resultieren einerseits aus Modifikationen durch den ursprünglichen Autor, um das Lernobjekt weiterzuentwickeln beziehungsweise zu korrigieren. Andererseits resultieren sie aus dem Wunsch, Lernobjekte nicht direkt übernehmen zu wollen, sondern diese nach eigenen Bedürfnissen anpassen zu können. Nach [BCV03] können Autoren ein Lernobjekt beispielsweise durch das Hinzufügen, Entfernen, Modifizieren und Umstrukturieren von Inhalten verändern.

Diese Aktivitäten führen dazu, dass ausgehend von dem ursprünglichen Lernobjekt ein neuer Zustand erreicht wird. Durch die Bearbeitung eines Lernobjekts LO1 lassen sich für das abgeleitete Lernobjekt LO2 in Abhängigkeit von den durchgeführten Veränderungen folgende vier Zustände identifizieren [BCV03]:

1. LO1 und LO2 sind äquivalent. Es bestehen keine Unterschiede bezüglich Inhalt und Metadaten zwischen beiden Versionen.
2. Das neu erstellte Lernobjekt LO2 enthält alle Inhalte der alten Version zuzüglich neuer Inhalte.
3. Das neu erstellte Lernobjekt LO2 enthält nur Inhalte, die aus der alten Version übernommen worden sind, aber nicht alle Inhalte der alten Version.
4. Es besteht eine bestimmte Übereinstimmung zwischen LO1 und LO2. Kein Lernobjekt ist jedoch eindeutig Fall 2 und 3 zuzuordnen.

Für Autoren, die Lernobjekte wiederverwenden möchten, kann die Beschreibung der Zustände der abgeleiteten Versionen eine wichtige Informationsquelle sein, um Unterschiede zwischen den verschiedenen Versionen eines Lernobjekts zu erkennen. Dies ist besonders wichtig für Autoren, die die alte Version eines Lernobjekts in einem ihrer Lernobjekte referenzieren. Die Unterschiede zwischen zwei verschiedenen

Versionen werden üblicherweise mit einer DIFF-Operation sichtbar gemacht. Die Beschreibung des Zustands der neuen Version eines Lernobjekts kann Autoren bei der Entscheidung unterstützen, ob sie die neue Version eines Lernobjekts in ihrem Lernobjekt verwenden möchten oder weiterhin die alte Version wiederverwenden möchten. Die Beschreibung der Unterschiede zwischen verschiedenen Versionen eines Lernobjekts ist eine Aufgabe des Versionsmanagements, das für den auf Aggregation basierenden Autorenprozess notwendig ist. Autoren muss eine Veränderung eines wiederverwendeten Lernobjekts angezeigt werden, damit sie daraufhin entscheiden können, welche der verfügbaren Versionen des Lernobjekts sie verwenden möchten. Für den Fall das verschiedene Varianten eines Lernobjekts parallel weiterentwickelt worden sind, ist eine sogenannte MERGE-Operation notwendig, um die verschiedenen Varianten des Lernobjekts wieder zusammen zu führen.

Eine besondere Anforderung des Autorenprozesses durch Aggregation an die Versionierung ist die Notwendigkeit, dass Lernobjektfragmente, die in aggregierten Lernobjekten referenziert werden, nicht verändert werden können. Die Stabilität der Lernobjektfragmente, die durch andere Lernobjekte zum Zweck der Aggregation referenziert werden, ist eine wesentliche Voraussetzung für einen auf Aggregation basierenden Autorenprozess. Autoren können Lernobjekte nur wiederverwenden, wenn die aggregierten Lernobjekte dauerhaft in unverändertem Zustand zur Verfügung stehen. Gleichzeitig muss aber gewährleistet werden, dass neue Versionen abgeleitet werden können, ohne dass eine Wiederverwendungsbeziehung beeinflusst wird [TA03].

Ein Anwendungsfall, bei dem die Stabilität von Lernobjekten vorausgesetzt wird, ist das Erstellen einer bezeichneten Version eines Kurses. Mit Hilfe der Bezeichner „Sommersemester 2005“ können Versionen von Kursen ausgezeichnet werden, die für das Sommersemester 2005 eingesetzt werden. Zu einem späteren Zeitpunkt kann dann mit Hilfe eines Versionierungssystems der Kurs in der Version „Sommersemester 2005“ wiederhergestellt werden, auch dann, wenn dieser zwischenzeitlich weiterbearbeitet worden ist. Diese Anwendungsfall tritt zum Beispiel dann ein, wenn Inhalte aus vergangenen Lehrveranstaltungen abgeprüft werden.

Das Versionsmanagement muss verhindern, dass die Suche von Autoren nach für ihre Zwecke geeigneten Lernobjekten durch viele ähnliche Lernobjekte stark behindert wird. Autoren möchten aus vielen verschiedenen Suchtreffern auswählen können und nicht viele Versionen eines Lernobjekts auffinden, die nur kleine Unterschiede aufweisen. Die Versionierung der Lernobjekte muss diesem Aspekt Rechnung tragen und das Ausgeben von mehreren Versionen eines Lernobjekts als Suchtreffer vermeiden. Gleichzeitig sollten jedoch die Autoren die Möglichkeit haben, alle Versionen eines Lernobjekts einsehen zu können.

Zusammenfassend lässt sich festhalten, dass die besonderen Anforderungen eines auf Aggregation basierenden Autorenprozesses in der Wiederverwendung von Lernobjekten begründet sind. Die Möglichkeit, Lernobjekte bestehend aus verschiedenen Lernobjektfragmenten über Referenzen zu aggregieren, führt dazu, dass Ver-

änderungen eines Lernobjekts ohne Versionsmanagement Auswirkungen auf alle Lernobjekte haben, die dieses Lernobjekt referenzieren. Dieser Effekt wird durch die Versionierung verhindert. Die Wiederverwendung von Lernobjekten in verschiedenen Kontexten und die Anpassung an die Bedürfnisse der unterschiedlichen Autoren führt weiterhin dazu, dass Lernobjekte sehr häufig verändert werden. Damit Autoren zu jeder Zeit das für sie am Besten geeignete Lernobjekt referenzieren, müssen sie benachrichtigt werden, wenn ein Lernobjekt modifiziert worden ist, das sie wiederverwenden. Der Autor muss entscheiden, welche Version des Lernobjekts er referenzieren möchte. Damit die Autoren nicht überlastet werden, sollte der Vorgang der Signalisierung von Veränderungen an Lernobjekten und die Auswahl der geeigneten Version eines Lernobjekts teilweise automatisiert werden. Im Folgenden werden Lösungsansätze entwickelt, die die Stabilität der aggregierten Lernobjekte gewährleisten und Autoren bei der Entscheidung unterstützen, welche Version des Lernobjekts referenziert werden soll.

6.2 Versionierung im Allgemeinen

Versionierung hat im Allgemeinen zur Aufgabe, Objekte mit veränderlichen Zuständen zu verwalten. Das Versionsmanagement zeichnet verschiedene Zustände eines Objekts mit einer Versionsnummer aus und macht diese Informationen einem System bzw. dessen Nutzern zugänglich. Neben dem Erstellen und Verwalten von verschiedenen Objektversionen umfasst das Versionsmanagement in der Regel auch die Speicherung der Entwicklungsgeschichte eines Objekts und die Möglichkeit, vergangene Zustände eines Objekts wiederherzustellen [BCV03].

Die Speicherung der Entwicklungshistorie der Versionen eines Dokuments verfolgt zwei Ziele: die Wiederherstellung der Lernobjekte nach Benutzer- oder Systemfehlern sowie die Analyse von Objektveränderungen. Zur Versionierung von Objekten existiert eine Vielzahl von Verfahren, die sich zwei grundsätzlich unterschiedlichen Methodenklassen zuordnen lassen. Die erste Methode speichert alle Versionen als vollständige Objekte, auf die über Referenzen zugegriffen werden kann. Der zweite Ansatz speichert eine Version vollständig. Die restlichen abgeleiteten Versionen werden als Unterschiede zwischen der aktuellen und der Vorgängerversion dargestellt. Die erste Methode ist einfach zu implementieren, ist aber sehr aufwendig bezüglich Speicherplatz und der Anzahl der verwalteten Versionen. Der zweite Ansatz ist wesentlich schwieriger zu implementieren, bietet aber Vorteile in der Darstellung kontinuierlicher, dynamischer Veränderungen der Objekte [KAM04].

Ein Fokus von Arbeiten zum Thema Versionierung im Allgemeinen liegt auf der Entwicklung von Verfahren zur Verwaltung von Objektversionen. Es wird angestrebt, die unter Umständen große Anzahl an Objektversionen auf möglichst wenig Speicherplatz abzulegen und mit geringem Aufwand wiederherstellen zu können [CTZ00]. Das Versionsmanagement von zusammengesetzten Lernobjekten und de-

ren Fragmenten wird in Arbeiten zum Thema Versionierung von Hypertext bzw. Hypermedien betrachtet [Haa96, Whi99, KNN99].

6.3 Annahmen für das Versionierungskonzept

Die Lösungsansätze für eine Versionierung von Lernobjekten sind abhängig von der Art der Speicherung der Lernobjekte und von den Zugriffsmöglichkeiten auf die verwalteten Lernobjekte. Folgende Annahmen über den Autorenprozess, innerhalb dessen die Versionierung realisiert werden soll, können für die Ausarbeitung des Versionierungskonzeptes getroffen werden:

1. Lernobjekte werden in einem zentralen Lernobjektarchiv abgelegt, mit dessen Hilfe alle verfügbaren Versionen abgerufen werden können. Die Modifikation der Lernobjekte erfolgt ausschließlich über ein zentrales Autorensystem. Es werden keine lokalen Kopien von Lernobjekten erstellt. Diese Prämisse schließt die Behandlung wie beispielsweise von Konflikten aus, die durch die Speicherung und Bearbeitung von lokalen Kopien entstehen [TA03].
2. Das Versionsmanagement muss die Versionierung gleichermaßen von Medienobjekten und von zusammengeführten Lernobjekten, unabhängig von deren Aggregationsniveau, ermöglichen. Lernobjekte höherer Aggregationsniveaus sind logische Gebilde, die sich aus Lernobjektfragmenten zusammensetzen. Die Möglichkeit der Kapselung von Lernobjektfragmenten in den zusammengeführten, monolithischen Lernobjekten wird nicht untersucht, weil dieses Verfahren dem Grundsatz der Wiederverwendung von Lernobjekten entgegensteht und eingekapselte Lernobjektfragmente nur bedingt in mehreren Lernobjekten wiederverwendet werden können [Whi99].
3. Das Versionsmanagement für modulare Lernobjekte ist als Erweiterung verbreiteter Versionierungssysteme zur differentiellen Speicherung verschiedener Versionen eines Objekts zu betrachten. Das heißt, dass keine explizite Betrachtung der differentiellen Speicherung von Versionen eines Lernobjekts erfolgt.
4. Die Versionierung basiert ausschließlich auf den Metadaten der Lernobjekte. Alle für die Versionierung erforderlichen Daten werden extern in den Metadatensätzen abgelegt, so dass die Versionierung unabhängig von den verwendeten Formaten der Lernobjekte durchgeführt werden kann und Lernobjekte nicht analysiert werden müssen, um an Versionsinformationen zu gelangen.
5. Ein Lernobjekt, das im Lernobjektarchiv abgelegt worden ist, kann nicht gelöscht werden, solange es von einem anderen Lernobjekt referenziert wird. Dies verhindert, dass Referenzen in Lernobjekten ungültig werden.

In den nächsten Abschnitten wird das Verfahren für die Versionierung modularisierter Lernobjekte entwickelt. Hierfür werden zunächst Lernobjekte mit dem Aggregationsniveau von Medienobjekten betrachtet. Danach wird gezeigt, wie die Versionierungskonzepte auf aggregierte Lernobjekte übertragen werden können.

6.4 Versionierung einzelner Lernobjekte

In diesem Abschnitt wird die Versionierung einzelner Lernobjekte beschrieben, die außer den Versionierungsbeziehungen keine weiteren Beziehungen zu Lernobjekten besitzen. Zunächst werden für diese Lernobjekte Zustände definiert, die Aussagen über deren Verwendbarkeit treffen. Danach wird die Speicherung der Versionsbeziehungen zwischen den Versionen des Lernobjekts, der Versionsbezeichner und der Sichtbarkeit von Versionen beschrieben. Anschließend wird das Erstellen von modifizierten Kopien beschrieben.

6.4.1 Zustände von Lernobjekten

Für die Versionierung und Wiederverwendung von Lernobjekten wird zwischen zwei prinzipiellen Zuständen unterschieden. Zur Klassifikation der Lernobjekte werden die Zustände „Lernobjekt ist in Bearbeitung“ und „Lernobjekt ist zur Wiederverwendung freigegeben“ eingeführt. Die Motivation für die Einführung dieser Unterscheidung der Lernobjekte ist darin begründet, dass nur Lernobjekte für die Wiederverwendung eingesetzt werden, die einen gewissen Stand erreicht haben. Hierdurch soll die Menge der zur Wiederverwendung angebotenen Lernobjekte um die Menge der noch nicht fertiggestellten Lernobjekte reduziert werden. Auf Basis des Zustands von Lernobjekten wird so eine Zugriffskontrolle für die Wiederverwendung von Lernobjekten eingeführt.

Den Zustand „Lernobjekt ist in Bearbeitung“ nehmen alle Lernobjekte ein, die durch von Autoren vorgenommene Änderungen unterliegen. Der Zeitraum, in dem sich ein Lernobjekt in diesem Zustand befindet, ist nicht an eine Sitzung im Autorensystem geknüpft. Es ist also möglich, dass sich ein Lernobjekt über einen längeren Zeitraum, über mehrere Sitzungen im Autorensystem hinweg in Bearbeitung befinden kann. So lange sich Lernobjekte in diesem Zustand befinden, ist eine Wiederverwendung ausgeschlossen, da sie noch in Bearbeitung sind.

Erst durch die explizite Anweisung des Autoren wechselt ein Lernobjekt von dem Zustand „Lernobjekt ist in Bearbeitung“ in den Zustand „Lernobjekt ist zur Wiederverwendung freigegeben“. Das geschieht dann, wenn der Autor eines Lernobjekts dieses für ausreichend ausgereift hält, so dass es wiederverwendet werden kann. Ein Lernobjekt, das diesen Zustand erreicht hat, kann nicht mehr bearbeitet werden und wird für die Wiederverwendung bereitgestellt.

Abbildung 6.1 zeigt ein Zustandsdiagramm mit den Zuständen, die ein Lernobjekt in seinem Lebenszyklus einnehmen kann. Direkt nachdem ein Lernobjekt angelegt wird, befindet es sich in dem Zustand „Lernobjekt ist in Bearbeitung“. Ausgelöst durch eine Aktion des Benutzers, wechselt das Lernobjekt in den Zustand „Lernobjekt ist zur Wiederverwendung freigegeben“ über. Ausgehend von diesem Zustand kann das Lernobjekt keinen anderen Zustand mehr einnehmen.

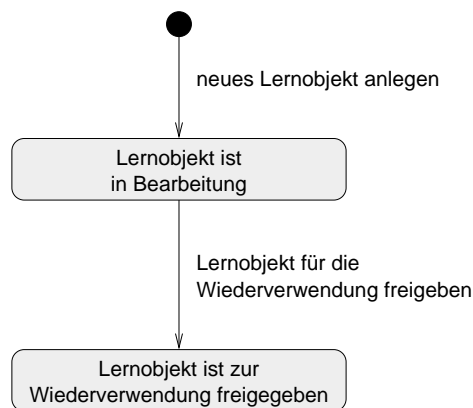


Abbildung 6.1: Zustände eines Lernobjekts

Der aktuelle Zustand, in dem sich ein Lernobjekt befindet, wird in einem Datenfeld des Metadatensatzes des jeweiligen Lernobjekts gespeichert. LOM [LOM02] bietet hierfür zwar für die Speicherung des Zustandes eines Lernobjekts das Datenfeld LOM.LifeCycle.Status, dieses beschreibt aber mit dem dafür festgelegten Vokabular „Draft“, „Final“, „Revised“ und „Unavailable“ nicht exakt den beschriebenen Verwendungszweck. Aus diesem Grund wird der Zustand eines Lernobjekts in einem erweiterten Datenfeld gespeichert. Der Name dieses Datenfeldes wird mit „IsReusable“ so gewählt, dass das feste Vokabular „false“ und „true“ ausreicht, um zu prüfen, ob sich das Lernobjekt in dem Zustand „Lernobjekt ist zur Wiederverwendung freigegeben“ befindet. Da sich ein Lernobjekt nur in einem dieser beiden Zustände befinden kann, wird per Konvention bestimmt, dass sich das Lernobjekt in dem Zustand „Lernobjekt ist in Bearbeitung“ befindet, wenn gilt, dass „IsReusable“ gleich „false“ ist. Diese Lösung ist auf die in Abbildung 6.1 beschriebenen Zustände limitiert, weil nur zwei Zustände mit dem Zustandsmarker kodiert werden können.

Abbildung 6.2 zeigt die visualisierende Notation des Zustandes eines Lernobjekts, die in den weiteren Betrachtungen verwendet werden soll. Zur Visualisierung des Zustandes eines Lernobjekts, wie sie im weiteren Verlauf der Arbeit verwendet wird, wird das Symbol aus Abbildung 4.5 durch einen Zustandsmarker in Form eines Rechtecks in der Ecke rechts oben eines Lernobjekts erweitert. Wenn das Rechteck weiß gefüllt ist, dann befindet sich das Lernobjekt in Bearbeitung. Ist das Rechteck dunkel gefüllt, so symbolisiert dies, dass das Lernobjekt zur Wiederverwendung

freigegeben ist. Das Lernobjekt in Abbildung 6.2 befindet sich demnach noch im Zustand der Bearbeitung.

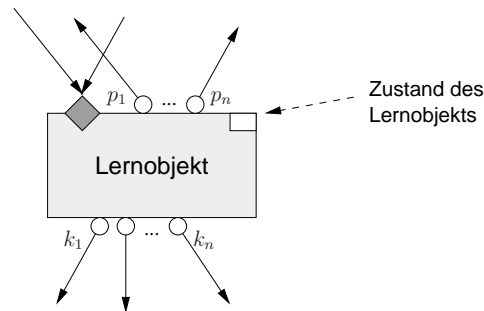


Abbildung 6.2: Zustandsmarker zum Speichern des Zustandes von Lernobjekten

Der in diesem Abschnitt beschriebene Lebenszyklus von Lernobjekten stellt sicher, dass nur die Lernobjekte wiederverwendet werden können, die es nach der Meinung des Autoren wert sind, wiederverwendet zu werden. Lernobjekte, die sich in der Bearbeitung befinden, können nicht wiederverwendet werden. Der Lebenszyklus beschreibt aber auch, dass Lernobjekte nicht mehr bearbeitet werden können, sobald diese für die Wiederverwendung freigegeben sind. Müssen sie aktualisiert oder für den Gebrauch in anderen Kontexten angepasst werden, nachdem sie zur Wiederverwendung freigegeben sind, ist die Versionierung der Lernobjekte notwendig.

6.4.2 Versionierung von Lernobjekten

Das Versionsmanagement, auch Versionierung genannt, bezeichnet das Erstellen und Verwalten von Versionen [BCV03]. Dabei handelt es sich bei der Version eines Objekts um einen bestimmten Zustand des Objekts [Whi99]. Mit Hilfe eines Versionsmanagements für Lernobjekte können dementsprechend bestimmte Zustände von Lernobjekten wiederhergestellt werden oder die historische Entwicklung von Lernobjekten angezeigt werden.

Zur Speicherung von Versionsinformationen wird das in Abschnitt 4.2 definierte Dokumentmodell entsprechend der Abbildung 6.3 erweitert. In den Metadaten wird zur Speicherung eines Bezeichners für die entsprechende Version eines Lernobjekts ein Metadatenfeld reserviert. Im LOM Metadatenchema kann hierfür das Datenfeld LOM.LifeCycle.Version verwendet werden. Dieses Datenfeld nimmt beliebige Zeichenketten zur Auszeichnung einer Version auf. In dem Beispiel enthält das entsprechende Datenfeld den Versionsbezeichner „V2“. Zusätzlich zu dem Versionskennzeichner werden in den Metadaten zwei Zeiger reserviert, die falls vorhanden auf die vorherige (v) beziehungsweise auf die nachfolgende (w) Version zeigen. Jedes Lernobjekt hat höchstens einen v -Zeiger beziehungsweise w -Zeiger. Zeiger auf

alternative Versionen werden im Gegensatz zu [Haa96] über die Definition von Varianten eines Lernobjekts realisiert. Das Erzeugen von Variationen wird in Abschnitt 6.4.5 beschrieben.

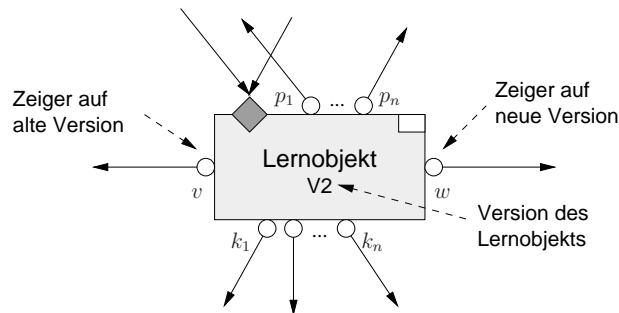


Abbildung 6.3: Versionen von Lernobjekten

Mit Hilfe der Zeiger auf die vorherige beziehungsweise nachfolgende Version können doppelt verkettete Listen von Lernobjekten in verschiedenen Versionen erzeugt werden. Diese verketteten Listen können genutzt werden, um die Historie von Lernobjekten zu betrachten. In erweiterten Anwendungen können sie auch genutzt werden, um Autoren, die Lernobjekte in älteren Versionen nutzen, darüber zu informieren, dass genutzte Lernobjekte aktualisiert worden sind. Abbildung 6.4 zeigt eine einfach verkettete Liste zweier Versionen eines Lernobjekts. Da die Version „V1“ die Urversion des Lernobjekts ist, besitzt sie keinen Zeiger v . w dagegen zeigt auf die nachfolgende Version mit dem Kennzeichner „V2“. Der Zeiger v dieser Version weist dementsprechend auf die Version mit dem Bezeichner „V1“. Da Version „V2“ die letzte Version des Lernobjekts ist, besitzt sie keinen Zeiger w .

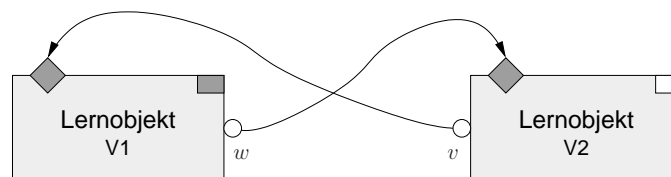


Abbildung 6.4: Verkettung verschiedener Versionen eines Lernobjekts

Das Erstellen neuer Versionen ist beschränkt auf Lernobjekte, die für die Wiederverwendung freigegeben sind. Mit Hilfe dieser Regelung wird dadurch das Erstellen neuer Lernobjekte auf der Basis von Lernobjekten unterbunden, die sich in Bearbeitung befinden. Lernobjekte, die für die Wiederverwendung freigegeben sind, können durch Erstellen einer neuen Version, die sich initial in Bearbeitung befindet, weiterbearbeitet werden. Hierdurch wird sichergestellt, dass Versionen von Lernobjekten, die wiederverwendet werden, im Lernobjektarchiv unverändert verfügbar

bleiben. Die Zustände der verschiedenen Versionen der Lernobjekte in Abbildung 6.4 sind entsprechend gesetzt.

Listing 6.1 zeigt den Algorithmus zur Erstellung einer neuen Version eines Lernobjekts. Der Algorithmus ist an eine bestimmte Instanz geknüpft. Ziel ist es, eine neue Version dieser Instanz zu erstellen. Zunächst werden die Eingangsvoraussetzungen geprüft. Eine neue Version eines Lernobjekts kann nur dann erstellt werden, wenn sich das Lernobjekt im Zustand „Lernobjekt ist zur Wiederverwendung freigegeben“ befindet und wenn noch keine neue Version existiert¹. Nach dieser Eingangsprüfung wird die neue Version erstellt. Hierfür wird das entsprechende Lernobjekt zunächst kopiert und anschließend in den Zustand „Lernobjekt ist in Bearbeitung“ gesetzt. Die Versionsnummer der neuen Version entspricht der um eins erhöhten Versionsnummer der Vorgängerversion. Nun wird das Lernobjekt mit der neu erzeugten Version verknüpft, in dem die Zeiger v und w der beiden Versionen wie in Abbildung 6.4 gesetzt werden.

Zusammen mit den im vorherigen Abschnitt beschriebenen Zuständen, die ein Lernobjekt annehmen kann, ergibt sich folgender typischer Ablauf bei stetiger Weiterentwicklung von Lernobjekten. Ein Lernobjekt wird initial angelegt und befindet sich im Zustand „Lernobjekt ist in Bearbeitung“. In diesem Zustand bleibt es so lange, bis der Autor des Lernobjekts mit der Bearbeitung fertig ist, so dass das Lernobjekt wiederverwendet werden kann. Damit dieses Lernobjekt für die Wiederverwendung freigegeben ist, versetzt der Autor des Lernobjekts dieses in den Zustand „Lernobjekt ist zur Wiederverwendung freigegeben“. Das Lernobjekt wird so in einen stabilen Zustand überführt, in dem es zwar nicht mehr zu bearbeiten ist, aber für die Wiederverwendung bereitsteht. Wenn das Lernobjekt weiterbearbeitet werden soll, wird eine neue Version des Lernobjekts erzeugt, die zunächst eine Kopie der alten Version darstellt, mit dem einzigen Unterschied, dass sich diese im Zustand „Lernobjekt ist in Bearbeitung“ befindet und dementsprechend wieder bearbeitet werden kann. Wenn alle Änderungen an dem Lernobjekt durchgeführt wurden, kann dieses wieder in den Zustand „Lernobjekt ist zur Wiederverwendung freigegeben“ wechseln und dem Kreislauf der Wiederverwendung zugeführt werden.

Die Zeiger v und w werden in den Metadaten, die das entsprechende Lernobjekt beschreiben, gespeichert. Zur Speicherung dieser Zeiger bietet LOM die Kategorie „Relation“, mit der Beziehungen zwischen dem durch den Metadatensatz beschriebenen Lernobjekt zu anderen Lernobjekten gespeichert werden können. Für jede dieser Beziehungen wird eine Instanz der Kategorie „Relation“ im Metadatensatz angelegt. Das Datenfeld LOM.Relation.Kind dient zur Klassifikation der Beziehung zwischen den Objekten. Zur Beschreibung der Beziehungen zwischen zwei Versionen eines Lernobjekts sind für LOM die Klassen „IsVersionOf“ und „HasVersion“

¹Durch das Erzeugen einer Variante, kann basierend auf einer alten Versionen die Entwicklung eines Lernobjekts fortgeführt werden. Varianten werden in Abschnitt 6.4.5 eingeführt.

```

class LearningResource {
...
    LearningResource createNewVersion() throws Exception {
        // test entry conditions
        if (!this.isReusable==true)
            throw new Exception("Illegal_state_Exception");
        if (this.nextVersion()!=null)
            throw new Exception("Resource_has_already_a_new_version");

        // create new learning object
        LearningResource newVersion;
        newVersion=this.clone();
        newVersion.setReusable(false);
        newVersion.setVersion(this.getVersion()+1);

        // link versions
        this.setNextVersion(newVersion);
        newVersion.setPreviousVersion(this);

        return newVersion;
    }
...
}

```

Listing 6.1: Erstellen einer neuen Version zur weiteren Bearbeitung eines Lernobjekts

definiert worden. Die Zeiger v und w werden dementsprechend mit „IsVersionOf“ beziehungsweise „HasVersion“ klassifiziert.

6.4.3 Bezeichnete Versionen

Ein großer Vorteil rechnergestützter Lehre ist die Möglichkeit der permanenten Weiterentwicklung von Lernobjekten, so dass im Vergleich mit dem Printbereich mit deutlich kürzeren Versionierungszyklen zu rechnen ist. Dieser oft mit sehr wenig Aufwand verbundenen Möglichkeit der Bearbeitung und Veränderung von Lernobjekten steht der Wunsch entgegen, für Prüfungen auf eine bestimmte prüfungsrelevante Version eines Lernobjekts zurückgreifen zu können. Aus diesem Grund ist das Festhalten besonderer Versionen von Lernobjekten ein wichtiger Aspekt bei der Entwicklung von Lernobjekten für die rechnergestützte Lehre. Ein Anwendungsfall hierfür ist das Erstellen einer bestimmten Version, im konkreten Fall für das Som-

mersemester 2005.

Für diese Anwendung wird das Konzept der bezeichneten Version eingeführt. Als bezeichnete Versionen werden Versionen eines Lernobjekts genannt, die zusätzlich zur Versionsnummer ein Kennzeichen tragen. Für jedes Lernobjekt gibt es maximal ein Kennzeichen. Zur Speicherung dieser Versionsbezeichners kann das LOM-Datenfeld LOM.LifeCycle.Version herangezogen werden. Da bezeichnete Lernobjekte eine bestimmte Fassung eines Lernobjekts auszeichnen, ist das Tragen von Bezeichnern ausschließlich Lernobjekten vorbehalten, die sich im Zustand „Lernobjekt ist zur Wiederverwendung freigegeben“ befinden.

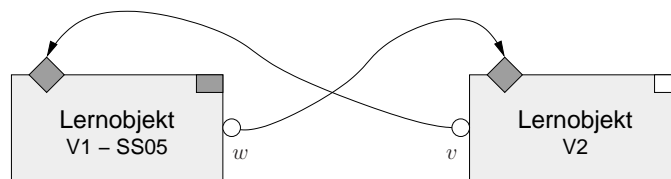


Abbildung 6.5: Bezeichnete Version eines Lernobjekts

Abbildung 6.5 zeigt zwei Versionen eines Lernobjekts. Version „V1“ ist bereits für die Wiederverwendung freigegeben und trägt den Bezeichner „SS05“. Mit Version „V2“ befindet sich das Lernobjekt wieder in Bearbeitung. Trotzdem ist es möglich zu einem späteren Zeitpunkt mit Hilfe des Bezeichners „SS05“ auf die Version des Lernobjekts zurückzugreifen, die für das Sommersemester 2005 eingesetzt wurde.

6.4.4 Sichtbarkeit von Versionen im Lernobjektarchiv

Im Laufe der Zeit entstehen durch die stetige Entwicklung eines Lernobjekts viele neue Versionen, die sich alle sehr ähnlich sind. Diese Vielfalt an sehr ähnlichen Lernobjekten erhöht die Anzahl der Suchtreffer innerhalb einer Suche nach Lernobjekten, welche so für die Autoren nur schwer überschaubar ist. Dies bringt somit letztendlich die Wiederverwendung von Lernobjekten in Gefahr.

Damit das Suchergebnis einer Suche nach Lernobjekten nicht dasselbe Lernobjekt in vielen verschiedenen Versionen anzeigt, werden alle Versionen ausgeblendet, die nicht primär für die Wiederverwendung geeignet sind. Unter der Annahme, dass neue Versionen hauptsächlich zum Zweck der Aktualisierung von Lernobjekten angelegt werden, werden ältere Lernobjekte bei der Anzeige von Suchergebnissen nicht berücksichtigt. Das gilt auch für bezeichnete Versionen von Lernobjekten, weil der Zugriff auf ältere bezeichnete Versionen aufgrund der Verkettung der Versionen über die aktuelle Version eines Lernobjekts erfolgen kann. Da Lernobjekte, die sich in Bearbeitung befinden, nicht wiederverwendet werden sollen, werden sie ebenfalls bei der Anzeige von Suchergebnissen nicht berücksichtigt.

Abbildung 6.6 zeigt ein Lernobjekt, das in den Versionen „V1“, „V2“ und „V3“ vorliegt. Bei der Suche nach wiederverwendbaren Lernobjekten wird nur Version

„V2“ berücksichtigt, weil die Versionen „V1“ und „V3“ ältere beziehungsweise zur Zeit nicht wiederverwendbare Versionen des Lernobjekts darstellen.

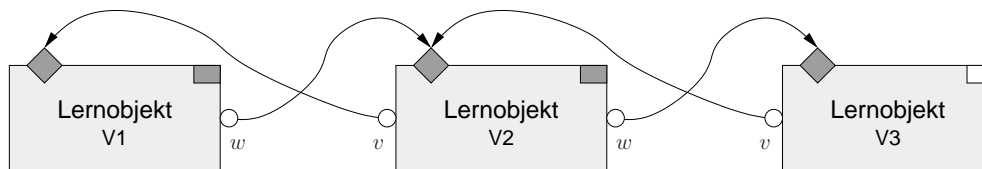


Abbildung 6.6: Sichtbarkeit von Versionen im Lernobjektarchiv

Die Anzeige alter Versionen kann dann in einem zweiten Schritt über die Anzeige einer Versionierungshistorie erfolgen, die über die doppelt verkettete Liste der Versionen ausgehend von jeder Version des Lernobjekts erzeugt werden kann. Alternative Zugänge zu bestimmten Versionen eines Lernobjekts sind vor allem für die letzten in Bearbeitung befindlichen Lernobjekte interessant. So können beispielsweise Lernobjekte eines Autors, die sich zur Zeit in Bearbeitung befinden in einem gesonderten Bereich angezeigt werden.

6.4.5 Modifizierte Kopien von Lernobjekten

Eine Anforderung aus dem Autorenprozess, die die Versionierung von Lernobjekten erforderlich macht, ist das Erstellen von Kopien von Lernobjekten zu dem Zweck, diese in modifizierter Form wiederzuverwenden. In [DH03] wird diese Art der Wiederverwendung von Lernobjekten als Repurposing beschrieben. Hierbei werden die wiederverwendeten Lernobjekte durch die Modifizierung einem neuen Zweck zugeführt. Ein weiterer Anwendungsfall der Versionierung ist beispielsweise das Erstellen von Lernobjekten, die als Vorlage für die Erstellung weiterer Lernobjekte dienen. Diese modifizierten Kopien von Lernobjekten werden Varianten genannt. Dabei handelt es sich um Lernobjekte, die im Vergleich zu deren ursprünglichen Lernobjekten im zeitfreien Sinne abgeändert wurden.

In der klassischen Versionierung wird bei der Erstellung einer Variante von einem Branch beziehungsweise Zweig gesprochen. Hierbei wird die Entwicklung eines separaten Stranges eines Lernobjekts, der auf einem bestimmten Zustand eines Lernobjekts basiert, von dem ursprünglichen Lernobjekt abgespalten. Zur Speicherung dieser Verzweigung bei der Entwicklung von Lernobjekten werden zusätzliche Zeiger im Datenmodell benötigt. Mit ihrer Hilfe soll die Verzweigung sowohl von der ursprünglichen als auch von dem neuen Lernobjekt in beide Richtungen verfolgt werden können, so dass immer nachvollzogen werden kann, von welchem Autor welche Lernobjekte stammen. Zu diesem Zweck wird für jedes Lernobjekt die Ursprungsversion gespeichert, falls dieses von einem anderen Lernobjekt abgeleitet wurde. Hierfür wird ein Zeiger benötigt, der nur dann gespeichert wird, falls ein

Lernobjekt eine Ursprungsversion hat. Zur Speicherung der abgeleiteten Lernobjekte wird eine variable Anzahl von Zeigern benötigt, da ein Lernobjekt beliebig oft kopiert und modifiziert werden kann.

Abbildung 6.7 zeigt die Erweiterung zum Datenmodell, das in Abschnitt 4.2 definiert wurde. Die Erweiterung sieht maximal einen Zeiger auf das ursprüngliche Lernobjekt u und eine beliebige Anzahl von Zeigern auf abgeleitete Lernobjekte a_n des Lernobjekts vor. Die Anzahl der Zeiger auf abgeleitete Lernobjekte stimmt dabei exakt mit der Anzahl der abgeleiteten Lernobjekte überein.

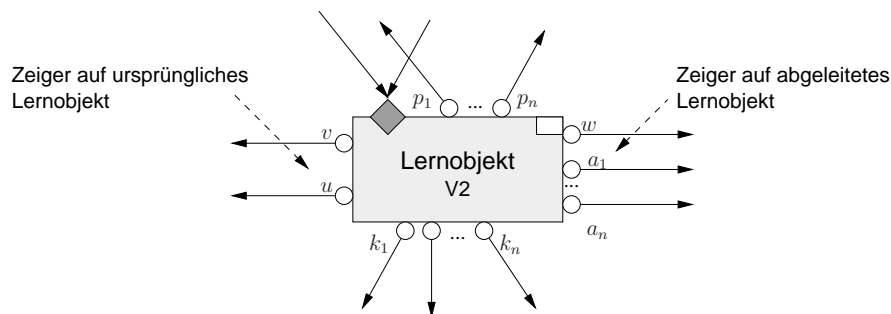


Abbildung 6.7: Ableiten von Lernobjekten

Die Zeiger a_n und u werden wie alle anderen Zeiger, die für die Versionierung benötigt werden, auch in den Metadaten des jeweiligen Lernobjekts gespeichert. Im LOM-Standard [LOM02] ist zur Speicherung dieser Zeiger die Kategorie „Relation“ vorgesehen. Die Werte, den die Felder dieser Kategorie annehmen, beschreiben eine Beziehung zwischen dem durch den Metadatensatz beschriebenen Lernobjekt und einem beliebigen anderen Lernobjekt. Zur Typisierung der Beziehung zwischen Ursprungs- und abgeleitetem Lernobjekt sind in LOM die Bezeichner „IsBasedOn“ und „IsBasisFor“ definiert worden. Für die Zeiger a_n und u werden die Bezeichner „IsBasedOn“ beziehungsweise „IsBasisFor“ verwendet.

Abbildung 6.8 zeigt ein Anwendungsbeispiel für das Ableiten eines Lernobjekts. In dem Beispiel dient die Version „V1“ des Lernobjekts A einerseits als Vorgängerversion für Version „V2“ aber auch als ursprüngliche Version des Lernobjekts B. Diese wiederum liegt bereits in der zweiten Version „V2“ vor.

Da das Wiederverwenden eines Lernobjekts in modifizierter Fassung eine spezielle Art von Wiederverwendung ist, gelten für sie dieselben Regeln wie bei Wiederverwendung durch Referenz. Dementsprechend kommen als ursprüngliche Lernobjekte nur die Lernobjekte in Betracht, die sich im Zustand „Lernobjekt ist zur Wiederverwendung freigegeben“ befinden.

Die Sichtbarkeit abgeleiteter Lernobjekte ist ebenso wie bei Lernobjekten, die in verschiedenen Versionen vorliegen, abhängig vom Zustand des Lernobjekts. Lernobjekte, die sich im Zustand „Lernobjekt ist für die Wiederverwendung freigegeben“ befinden, sind für alle Benutzer bei der Anzeige von Suchtreffern sichtbar. Für den

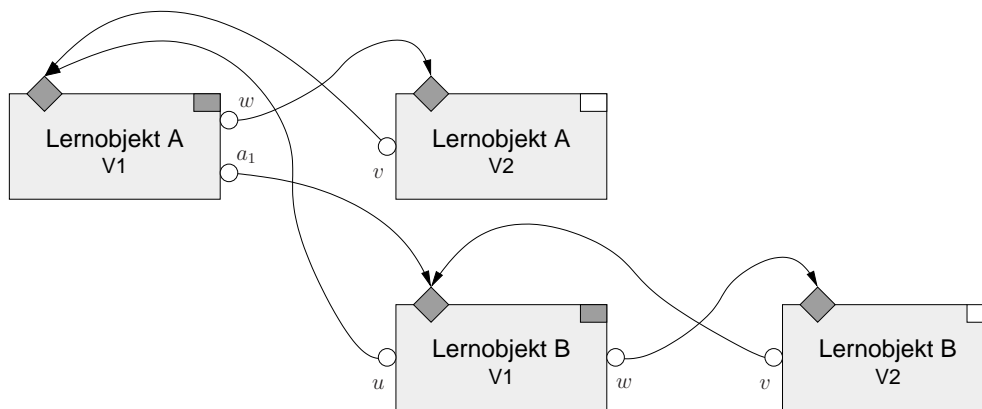


Abbildung 6.8: Beispiel eines abgeleiteten Lernobjekts

Fall, dass mehrere Lernobjekte sich in diesem Zustand befinden, ist nur die jüngste Version sichtbar.

Genauso wie beim Erstellen von neuen Versionen von Lernobjekten, können die Metadaten von abgeleiteten Lernobjekten ebenfalls aus dem Metadatensatz des ursprünglichen Lernobjekts abgeleitet werden, siehe Abschnitt 5.1.8.

6.5 Versionierung von Dokumentknoten

Bei den bisherigen Betrachtungen wurde vernachlässigt, dass nach dem in Abschnitt 4.2 beschriebenen Dokumentmodell Lernobjekte aus Lernobjektfragmenten aggregiert werden können. Diese Eigenschaft der Lernobjekte hat Auswirkungen auf Zustand und Version der Lernobjekte, die sich ausgehend von dem aktuell betrachteten Lernobjekt weiter oben oder weiter unten in der Dokumenthierarchie befinden. Im Folgenden sollen Auswirkungen auf den Versionierungszustand der Lernobjekte eines aggregierten Kurses untersucht werden, die sich durch die Veränderung eines Lernobjekts des entsprechenden Kurses ergeben. In diesem Zusammenhang wird die Bedeutung der Aggregationsbeziehungen auf die Zustandsänderung der Versionierung einzelner Lernobjekte in aggregierten Lernobjekten untersucht. In diesem Kontext ist besonders die Propagierung von Zuständen und Versionen zwischen den aggregierten Lernobjekten eines Kurses von Interesse.

6.5.1 Propagierung von Zuständen

Die in Abschnitt 6.4.1 eingeführten Zustände von Lernobjekten gelten auch für Lernobjekte, die Knoten eines Kurses nach dem in Abschnitt 4.2 definierten Dokumentmodell darstellen. Ebenso wie alleinstehende Lernobjekte befinden sich alle Lernobjekte eines Kurses zunächst im Zustand „Lernobjekt ist in Bearbeitung“.

Ausgelöst durch Autoren können einzelne Knoten und damit einzelne Teile des Kurses in den Zustand „Lernobjekt ist zur Wiederverwendung freigegeben“ versetzt werden. Durch den Inklusionscharakter der „HasPart“-Beziehung hat diese durch den Autor ausgelöste Änderung des Zustands eines Knotens Auswirkungen auf die Zustände referenzierter Lernobjekte. Davon sind alle Lernobjektfragmente betroffen, aus denen sich das entsprechende Lernobjekt zusammensetzt. Da diese Lernobjektfragmente Teil des aggregierten Lernobjekts sind und somit ebenfalls einen gewissen Bearbeitungsgrad erlangt haben, so dass sie wiederverwendet werden können, wechseln auch sie in den Zustand, in dem sie für die Wiederverwendung bereitstehen. In diesem Fall spricht man von der Propagierung des Zustandswechsels im Kurs bis hin zu den Medienobjekten des Kurses. Das entspricht einem optimistischen Zustandswechsel bei den Lernobjekten. Alternativ dazu ist auch ein Zustandswechsel ausgehend von den Blattknoten eines Kurses denkbar, was einer pessimistischen Zustandsänderung entspricht. Aufgrund des Inklusionscharakters der „HasPart“-Beziehung kann jedoch ohne Probleme der optimistische Zustandswechsel gewählt werden, der zudem benutzerfreundlicher ist.

Abbildung 6.9 zeigt in vereinfachter Darstellung ein Beispiel für die Änderung des Versionierungszustandes von Lernobjekten eines Kurses. Die einzelnen Knoten eines Kurses werden durch Rechtecke dargestellt, die in der linken oberen Ecke den in 6.4.1 definierten Zustandsmarker enthalten. Zur Erhöhung der Übersichtlichkeit der Abbildung sind die Ankerpunkte der Lernobjekte und die 'IsPartOf'-Beziehungen eingespart worden. Das Beispiel zeigt das Lernobjekt R1, das sich direkt aus den Lernobjekten R2 und R3 zusammensetzt. Diese setzen sich wiederum aus den Lernobjekten R4 und R5 beziehungsweise R6 und R7 zusammen. Lernobjekt R2, soll jetzt für die Wiederverwendung freigegeben werden. Hierzu wird der Zustandsmarker von R2 auf „Lernobjekt ist zur Wiederverwendung freigegeben“ umgeschaltet. Diese Änderung betrifft jetzt auch die Lernobjekte R4 und R5, aus denen sich Lernobjekt R2 zusammensetzt. Auf R1 und die Geschwisterknoten von R2 hat die Änderung keinen Einfluss.

Das Beispiel aus Abbildung 6.9 zeigt sehr anschaulich die Propagierung der Zustandsänderung eines Knotens an seine Kinder und Kindeskinde. Die Propagierung der Wiederverwendungszustände innerhalb des Dokumentmodells ist sehr wichtig für die Wiederverwendung, weil die Wiederverwendung von Lernobjekten ohne deren aggregierte Lernobjektfragmente nicht sinnvoll ist. Zudem wirkt sich die Propagierung der Wiederverwendungszustände positiv auf die Wiederverwendung von Lernobjektfragmenten aus.

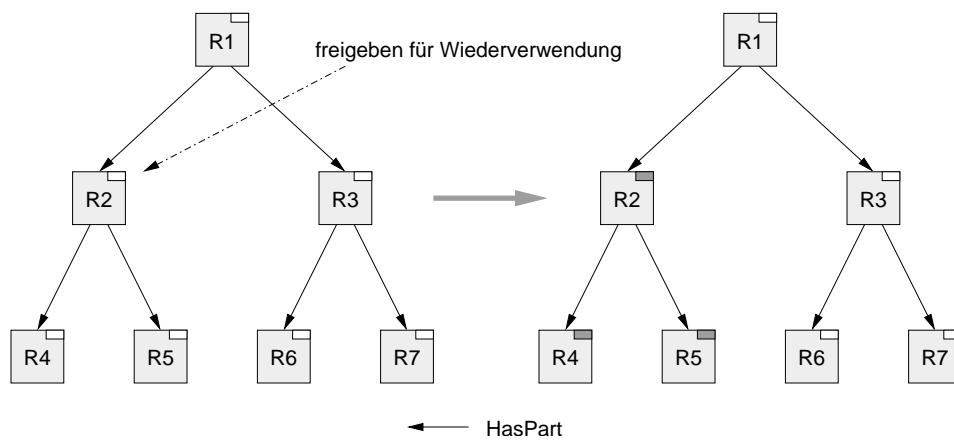


Abbildung 6.9: Propagation von Zustandsänderungen im Kurs

6.5.2 Propagierung neuer Versionen

Ebenso, wie sich die Änderung des Zustandes eines Knotens im Kursdokument auf andere Knoten auswirken, kann das Erstellen von neuen Versionen eines Knotens im Kursdokument das Erstellen neuer Versionen übergeordneter Knoten nach sich ziehen. Grund hierfür ist, dass für die neue Version eines Lernobjekts ein neues Lernobjekt mit eigenem Identifikator angelegt wird, das im Kurs aber noch nicht referenziert ist. Unter der Voraussetzung, dass die Änderungen des Knotens im ursprünglichen Kurs übernommen werden sollen, ist es erforderlich, diese neue Version statt der alten Version im Kurs zu referenzieren. Hierfür ist eine Änderung im Kurs erforderlich. Direkt betroffen von dieser Änderung ist der Vaterknoten des Lernobjekts dessen neue Version Teil des Kurses werden soll. Dieser muss stattdessen zukünftig die neue Version des Lernobjekts referenzieren. Wenn für diese Änderung eine neue Version des Vaterknotens erforderlich ist, wird auch die Änderung des Vaterknotens des Lernobjekts notwendig, das ursprünglich verändert wurde. Dabei ist es möglich, dass sich das Erstellen neuer Versionen bis hin zum Wurzelknoten des Kursdokuments fortsetzt. In der Literatur wird dies als "Version Proliferation" bezeichnet.

Ob das Erstellen neuer Versionen des Vaterknotens notwendig ist, hängt vom Zustand des Vaterknotens ab. So lange der Vaterknoten in Bearbeitung ist können die Änderungen übernommen werden, ohne eine neue Version des Vaterknotens zu erstellen. Dementsprechend wird andererseits das Erstellen einer neuen Version des Vaterknotens erforderlich, wenn er bereits für die Wiederverwendung freigegeben ist.

Abbildung 6.10 greift das Beispiel des vorherigen Abschnitts wieder auf. Das Kursdokument besteht aus den Lernobjekten R1 ... R7 von denen sich alle Lernobjekte außer den Lernobjekten R2, R4 und R5 noch in der Bearbeitung befinden. Für das Beispiel soll Lernobjekt R2 um Lernobjekt R8 ergänzt werden. Lernobjekt R8 befindet sich noch in Bearbeitung. Da Lernobjekt R2 bereits zur Wiederverwendung freigegeben ist, wird eine neue Version von R2 angelegt, um die erforderliche Änderung am Kurs vornehmen zu können. Die Änderungen werden in R2' angelegt. Wie bereits in Abschnitt 6.4.2 beschrieben, werden die beiden Versionen R2 und R2' mit den Beziehungen „HasVersion“ und „IsVersionOf“ miteinander verknüpft. Statt Lernobjekt R2 referenziert Lernobjekt R1 jetzt Lernobjekt R2' mit Hilfe einer „HasPart“-Beziehung. Zur Durchführung dieser Änderung ist es nicht erforderlich eine neue Version von R1 zu erstellen, weil sich R1 noch in Bearbeitung befindet.

Anders als in dem vorherigen Beispiel sei nun der vollständige Kurs bestehend aus den Lernobjekten R1 bis R7 für die Wiederverwendung freigegeben, wie in Abbildung 6.11 dargestellt. Auch in diesem Beispiel soll Lernobjekt R2 um Lernobjekt R8 erweitert werden. Wie in Abbildung 6.11 wird deswegen für Lernobjekt R2 die neue Version R2' erstellt. Die daraus resultierende Notwendigkeit der Änderung von Lernobjekt R1 kann nur durch Erstellen einer neuen Version von Lernobjekt R1

6.5.3 Propagierung benannter Versionen

Analog zur Propagierung des Wechsels des Zustandes eines Lernobjekts, das Teil eines Kurses ist, ist die Propagierung bezeichneter Versionen über die untergeordneten Lernobjekte eines Kursdokumentes möglich. Dies führt jedoch zu folgendem Problem. Ein Lernobjekt, das unverändert Teil verschiedener benannter Versionen eines aggregierenden Lernobjekts ist, würde aufgrund der Propagierung der Versionsbezeichner verschiedene Versionsbezeichner tragen, obwohl es sich immer um dieselbe Version handelt. Aus diesem Grund wird die Propagierung von Versionsbezeichnern im Kursdokument ausgeschlossen. Stattdessen werden jeweils nur die Wurzelemente aggregierter Lernobjekte mit dem Versionsbezeichner ausgezeichnet.

6.6 Zusammenfassung

Die Versionierung von Lernobjekten ist die Voraussetzung für die Wiederverwendung von modifizierbaren Lernobjekten in einem auf Aggregation basierenden Autorenprozess. Durch die Aggregation von Lernobjekten müssen bei der Versionierung die Beziehungen der Lernobjekte untereinander berücksichtigt werden. Das wird in den bereits existierenden Arbeiten auf diesem Gebiet mit hohem Aufwand geleistet. Es fehlt dort aber an der Berücksichtigung des auf Aggregation basierenden Autorenprozesses. In diesem Kapitel wurde ein Verfahren für die Versionierung von Lernobjekten vorgestellt, das diese Einschränkungen nicht besitzt.

Das Verfahren basiert auf der zentralen Speicherung der Lernobjekte, so dass keine Konflikte durch lokale Kopien aufgelöst werden müssen. Das Besondere an dem vorgestellten Verfahren ist die Speicherung aller für die Versionierung relevanten Informationen in den Metadaten der Lernobjekte. Die Versionierung erfolgt unabhängig von den Aggregationsniveaus der Lernobjekte. Daten für die Versionierung der Lernobjekte werden in den Metadaten der Lernobjekte gespeichert.

Zu den wichtigsten Eigenschaften des in der vorliegenden Arbeit entwickelten Versionierungsverfahrens gehört das Einführen von Zuständen für Lernobjekte, die Auskunft über die Wiederverwendbarkeit geben. Es werden sowohl Varianten und bezeichnete Versionen von Lernobjekten unterstützt, als auch die Propagierung von Zuständen und Versionen innerhalb aggregierter Kurse berücksichtigt. Außerdem wird unter Verwendung der Versionierungsbeziehungen und des Zustandes eines Lernobjekts die Sichtbarkeit der Lernobjekte bei der Anzeige von Suchergebnissen geregelt.

7 Implementierung

In diesem Kapitel wird das ResourceCenter, die Implementierung der zuvor erarbeiteten Konzepte, beschrieben. In dem ersten Teil des Kapitels wird die für das ResourceCenter gewählte Systemarchitektur begründet. Hierfür wird diese schrittweise aus einem einfachen System abgeleitet. Danach werden die wichtigsten Komponenten des ResourceCenters beschrieben. Dies ist zum einen das Lernobjektarchiv des ResourceCenters, mit dessen Hilfe die wiederverwendbaren Lernobjekte verfügbar gemacht werden. Die Beschreibung des Lernobjektarchivs behandelt das verwendete Metadatenschema inklusive dessen Erweiterungen und die Funktionen zum Verwalten von Lernobjekten und den Metadaten-Wizard zur Erstellung von Metadaten auf Basis des in Kapitel 5 entwickelten Verfahrens. Die zweite wichtige Komponente ist die auf Aggregation basierende Autorenumgebung. Bei deren Betrachtung wird zunächst das umgesetzte Datenmodell für Kurse im ResourceCenter erläutert, das einen entscheidenden Einfluss auf die Realisierung der Autorenumgebung hat. Anschließend werden die EditorKomponenten beschrieben, mittels der die Lernobjektfragmente auf verschiedenen Aggregationsniveaus bearbeitet und aggregiert werden können. Zum Schluss werden verschiedene Wege für das Einfügen von Lernobjekten beim Aggregieren genauer betrachtet.

7.1 Systemarchitektur

In diesem Abschnitt wird eine Architektur für ein System zur Erstellung von Lernobjekten entworfen. Der Entwurf der Systemarchitektur entspricht den in Kapitel 4 ausgearbeiteten Anforderungen an einen auf Aggregation basierenden Autorenprozess. Hierfür werden verschiedene alternative Architekturen vorgestellt und bewertend diskutiert.

7.1.1 Anwendungsfälle

Abgeleitet aus der Analyse des Autorenprozesses, vergleiche Abschnitt 4.3, ergeben sich für die Nutzer einer auf Aggregation basierenden Autorenumgebung verschiedene Anwendungsfälle. Die Anwendungsfälle lassen sich in zwei Funktionsbereiche gliedern. Der erste Funktionsbereich stellt Funktionen zum Erstellen und Bearbeiten von aggregierten Lernobjekten dar. Der zweite Funktionsblock besteht aus den

Funktionen für das Arbeiten mit Lernobjekten in einem Lernobjektarchiv, vergleiche Abschnitt 3.5.

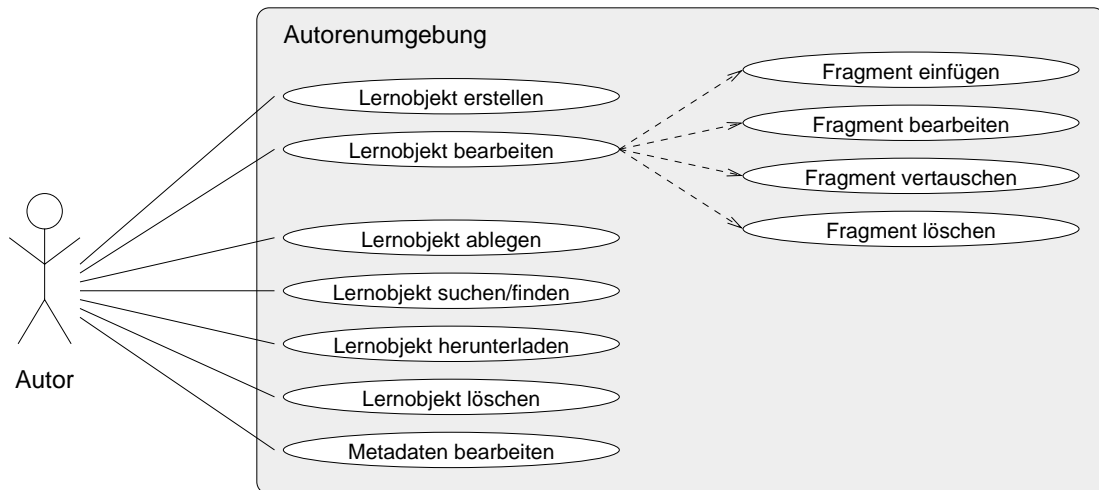


Abbildung 7.1: Anwendungsfälle für die Autorenenumgebung

Abbildung 7.1 zeigt diese Anwendungsfälle im Überblick. Die obere Gruppe der Anwendungsfälle stellt die Kernfunktionen der Autorenenumgebung dar. Zu den Kernfunktionen gehört das Erstellen und Bearbeiten von Lernobjekten. In dem Autorenprozess basiert der Anwendungsfall „Lernobjekt bearbeiten“ auf weiteren Anwendungsfällen zum Einfügen, Bearbeiten, Vertauschen und Löschen von Fragmenten, die den Anwendungsfall „Lernobjekt bearbeiten“ erweitern. Diese Anwendungsfälle können nur im Kontext von „in Bearbeitung“ befindlichen aggregierten Lernobjekten aufgerufen werden.

Der zweite Block besteht aus Anwendungsfällen zum Einstellen beziehungsweise Ablegen, Suchen und Finden, Herunterladen und Löschen von Lernobjekten sowie dem Bearbeiten von Metadaten. Diese Anwendungsfälle entsprechen den Funktionen eines Lernobjektarchivs. Sie werden hier trotzdem als Anwendungsfälle der Autorenenumgebung zur Aggregation von Lernobjekten aufgezählt, weil sie gerade bei der Erstellung von Lernobjekten nach dem Bottom-Up-Verfahren genutzt werden können. Bei diesem Verfahren werden Kurse beginnend von Medienobjekten durch sukzessive Aggregation erstellt.

In den nächsten Abschnitten wird eine Architektur für ein System entwickelt, mit dessen Hilfe eine Gruppe von Autoren Lernobjekte auf Basis von Aggregation unter Verwendung bereits bestehender und Einspeisung beziehungsweise Entwicklung neuer Lernobjektfragmente erstellen kann.

7.1.2 Einfaches Einzelplatzsystem

In einem ersten Entwurf wird eine Autorenumgebung zum Aggregieren von Lernobjekten vorgestellt, die zeitgleich nur einem Autoren erlaubt, Lernobjekte zu aggregieren. Die Verwendung eines Einzelplatzsystems entspricht jedoch nicht den definierten Anforderungen, weil mit einem Einzelplatzsystem Lernobjekte nicht für viele Autoren verfügbar gemacht werden und diese dementsprechend auch nicht von fremden Autoren wiederverwendet werden können. Die Diskussion dieses Einzelplatzsystems soll aber genutzt werden, um verschiedene andere Aspekte der Systemarchitektur zu erörtern und diese weiter zu spezifizieren.

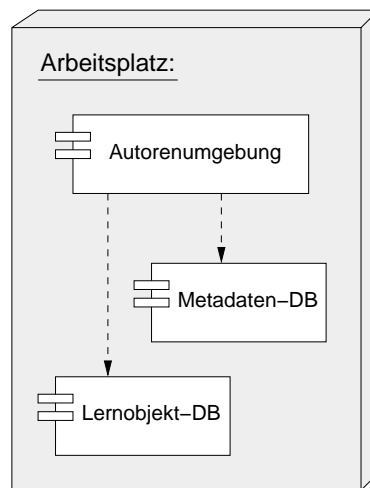


Abbildung 7.2: Systemkomponenten eines einfachen Einzelplatzsystems

Abbildung 7.2 zeigt das Komponentendiagramm einer Systemarchitektur einer Autorenumgebung, die für die weitere Betrachtung als „einfaches Einzelplatzsystem“ bezeichnet wird. Das Komponentendiagramm beschreibt die Zusammensetzung und Abhängigkeit der Teilkomponenten des Einzelplatzsystems, den „Arbeitsplatz“ eines Autoren. Die zentrale Komponente des Arbeitsplatzes ist die Autorenumgebung, vergleiche Abbildung 7.2. Die Autorenumgebung greift zur Speicherung der modularisierten Lernobjekte und deren Metadatenbeschreibungen direkt auf Datenbanken zu. Die Metadatensätze, die in der Metadatenbank abgelegt sind, beschreiben die modularisierten Lernobjekte des Lernobjektarchivs.

Diese Architektur impliziert neben dem bereits erwähnten Problem der fehlenden Unterstützung mehrerer Autoren in der Wiederverwendung von Lernobjekten verschiedene weitere Probleme. Diese lassen sich alle auf die fehlende Modularisierung der Komponente „Autorenumgebung“ zurückführen. Zusätzlich zu der eigentlichen Kernaufgabe dieser Komponente, der Aggregation bestehender und neu erstellter Lernobjekte, müssen in der Komponente Funktionen eines Lernobjektarchivs zur Speicherung und Wiederverwendung von Lernobjekten unterstützt werden. Außer-

dem müsste die Autorenumgebung zusätzlich die Bearbeitung von Medienobjekten mittels integrierter Werkzeuge unterstützen. Die Entwicklung von Funktionen beispielsweise zur Bearbeitung von Bildern und Animationen und dem Erstellen von Flash, mittels dem Animationen erstellt werden können, würde jedoch einen enormen Zusatzaufwand bedeuten.

7.1.3 Erweitertes Einzelplatzsystem

Aufgrund dieser Probleme wird die Architektur des einfachen Einzelplatzsystems insbesondere die Komponente „Autorenumgebung“ weiterentwickelt.

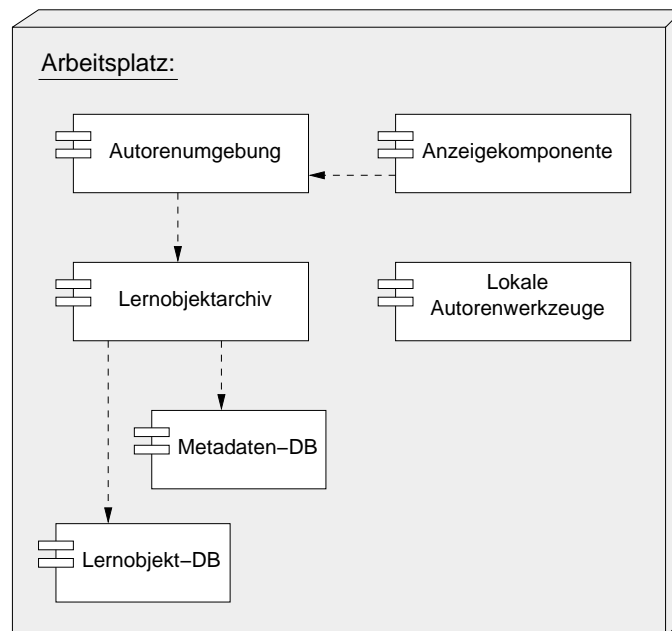


Abbildung 7.3: Systemkomponenten des erweiterten Einzelplatzsystems

Wie das Komponentendiagramm in Abbildung 7.3 zeigt, sind in dem „erweiterten Einzelplatzsystem“, die Lernobjektarchivfunktionen von der Autorenumgebung getrennt. Analog zu der Gliederung einer Netzwerkimplementierung in verschiedene Schichten, wie in [Tan03, SN04b] beschrieben, führt die Trennung von Autorenumgebung und Lernobjektarchiv zu einer übersichtlicheren Verteilung der Funktionen auf die einzelnen Komponenten. Durch die Trennung der Komponenten wird deren Komplexität auf mehrere Komponenten verteilt. Analog zu einer Netzwerkimplementierung sind untergeordnete Komponenten, wie das Lernobjektarchiv Dienstbringer und die Autorenumgebung Dienstanwender. Die Autorenumgebung nutzt die Dienste des Lernobjektarchivs, um Autoren bestehende Lernobjekte für die Aggregation verfügbar zu machen und um mittels der Autorenumgebung aggregierte

Lernobjekte zu speichern und diese so für die Wiederverwendung verfügbar zu machen. Aufgrund dieser Trennung können bestehende Lernobjektarchive Bestandteil des Gesamtsystems sein.

Zusätzlich zu der Herauslösung des Lernobjektarchivs aus der Autorenumgebung, sind in der Architektur des erweiterten Einzelplatzsystems die Komponenten zur Anzeige der Lernobjekte und zum Erstellen von Medienobjekten, wie Bilder und Animationen, von der Autorenumgebung abgetrennt worden. Hierdurch wird einerseits die Komplexität der Autorenumgebung reduziert und andererseits die Möglichkeit geschaffen, beispielsweise für die Produktion von Medienobjekten bestehende Produkte, wie Adobe Photoshop und Macromedia Flash, zu nutzen. Diese Werkzeuge sind dann als externe Werkzeuge für die Autorenumgebung verfügbar.

Durch die Trennung der Anzeigekomponente kann auf bereits existierende Komponenten zur Visualisierung der Lernobjekte zurückgegriffen werden. Durch die Verwendung eines Browsers als Anzeigekomponente der Lernobjekte während des Autorenprozesses können alle gängigen webfähigen Lernobjekte für rechnergestützte Lehre angezeigt werden. Zugleich ist es damit auch möglich mittels einer integrierten Vorschaufunktion die erstellten Lernobjekte so darzustellen, wie sie den Lernern angezeigt werden, da Lernobjekte eines WBTs zumeist in einem Browser angezeigt werden.

Das Zusammenspiel der einzelnen Komponenten sieht bei dieser Architektur wie folgt aus: Mit Hilfe der Autorenumgebung können bestehende Lernobjekte und neu erstellte Lernobjekte aggregiert werden. Auf bereits bestehende Lernobjekte können Autoren über das an die Autorenumgebung angeschlossene Lernobjektarchiv zugreifen. Lernobjekte, die neu erstellt werden müssen, werden mit der Autorenumgebung erstellt. Sämtliche Lernobjekte, die mit Hilfe der Autorenumgebung aggregiert und neu erstellt werden, werden modular mit den dazugehörigen Metadaten in dem Lernobjektarchiv abgelegt, damit sie wiederverwendet werden können.

Trotz der Erweiterungen bleibt auch das erweiterte System ein Einzelplatzsystem zur Aggregation von Lernobjekten. Damit Lernobjekte von einer größeren Gruppe von Autoren wiederverwendet werden können, müssen Wege gefunden werden, die Lernobjekte einer Gruppe von Autoren verfügbar zu machen. Im nächsten Abschnitt werden hierfür verschiedene Ansätze diskutiert.

7.1.4 Mehrplatzsysteme

Die Systemarchitektur des erweiterten Einzelplatzsystems besitzt zwar alle Systemkomponenten, um bestehende und neu erstellte Lernobjekte zu aggregieren, jedoch erlaubt die Systemarchitektur nicht den Austausch der Lernobjekte zwischen mehreren Autoren. In diesem Abschnitt werden verschiedene alternative Architekturen vorgestellt, mit deren Hilfe bereits bestehende und neu erstellte Lernobjekte zwischen mehreren Autoren ausgetauscht werden können, so dass eine größere Gruppe von Autoren gemeinsam Lernobjekte wiederverwenden kann.

Aus der Systemarchitektur des erweiterten Einzelplatzsystems lässt sich durch eine Verteilung der Systemkomponenten auf verschiedene Subsysteme eine Architektur entwickeln, mittels der Lernobjekte einer Gruppe von Autoren zugänglich gemacht werden können. Hierfür ist es erforderlich die Systemkomponenten auf verschiedenen Systemen zu verteilen. Eine Möglichkeit bietet die Verteilung der Systemkomponenten auf ein Server- und ein Client-System zu verteilen. Das Client-System stellt bei diesem Ansatz den Arbeitsplatz eines Autoren dar. Aus diesem Grund muss ein Client-System für jeden Autor existieren.

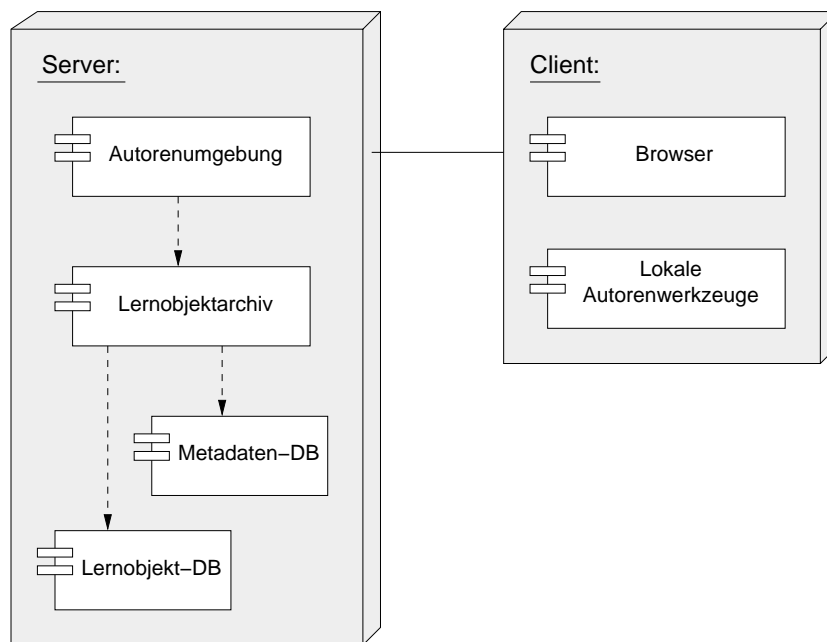


Abbildung 7.4: Zentrale Autorenumgebung mit zentralem Lernobjektarchiv

Abbildung 7.4 zeigt mittels des Verteilungsdiagramms die Verteilung der Systemkomponenten auf das Server- und Client-System. Demnach befinden sich bei dieser Systemarchitektur die Autorenumgebung und das Lernobjektarchiv mit den dazugehörigen Datenbanken für die modularisierten Lernobjekte und Metadaten auf dem Server-System. Die Client-Systeme enthalten lediglich den Browser zur Bedienung der Autorenumgebung und die lokalen Werkzeuge zum Erstellen und Bearbeiten von Medienkomponenten.

Unter der Bedingung, dass ein Web-Browser als Anzeigekomponente eingesetzt wird, kann die Verteilung der Systemkomponenten auf die Clients und den Server ohne zusätzlichen Aufwand realisiert werden, wenn die Autorenumgebung bereits für die gleichzeitige Nutzung durch mehrere Autoren geeignet ist. Sie stellt also eine einfache Lösung zur Verteilung der Lernobjekte dar. Der Nachteil dieses Systems ist, dass ein Arbeiten ohne direkte Netzwerkverbindung zum Server nicht möglich ist.

Die Weiterentwicklung der Client-/ Server-Architektur zu einem System, mit dessen Hilfe Lernobjekte auch ohne Netzwerkverbindung zum zentralen Server aggregiert werden können, macht die Verschiebung der Autorenumgebung auf das Client-System nötig. Für die Wiederverwendung bereits bestehender Lernobjekte und das Speichern neu erstellter Lernobjekte wird zudem ein Lernobjektarchiv auf dem Client-System erforderlich. Auf dem zentralen Server verbleibt das zentrale Lernobjektarchiv, das dazu genutzt wird die Lernobjekte zwischen den Autoren auszutauschen, die nicht immer eine Netzwerkverbindung zum zentralen Server besitzen. Für die Synchronisierung der Lernobjekte eines lokalen Lernobjektarchivs mit dem zentralen Lernobjektarchiv besitzen sowohl die Archive auf dem Client-System, als auch das Archiv des zentralen Archivs jeweils eine Synchronisierungskomponente.

Abbildung 7.5 zeigt das weiterentwickelte Client-/ Server-System mit zentralem Lernobjektarchiv. Das zentrale Server-System enthält jetzt nur noch ein Lernobjektarchiv mit den dazugehörigen Datenbanken und einer Synchronisierungskomponente. Das Client-System gleicht vom Aufbau dem erweiterten Einzelplatzsystem. Es wird lediglich durch die Synchronisierungskomponente erweitert, mittels der das lokale Lernobjektarchiv mit dem zentralen Lernobjektarchiv synchronisiert werden kann.

Diese Erweiterung des Systems zu einem System mit dessen Hilfe auch ohne Netzwerkverbindung Lernobjekte aggregiert werden können, kommt mit Ausnahme der Synchronisierungskomponente mit bereits bestehenden Komponenten aus. Die Kosten der Erweiterung umfassen deswegen im Prinzip die Entwicklung einer Synchronisierungskomponente zur Synchronisierung der lokalen Lernobjektarchive mit dem zentralen Lernobjektarchiv. Bei der Entwicklung der Synchronisierungskomponente sind jedoch verschiedene Hindernisse zu überwinden. Hierbei ist vor allem die Konsistenzhaltung der Datensätze durch Synchronisierung der Lernobjektarchive eine Herausforderung. Sowohl Referenzen auf nicht vorhandene Lernobjekte als auch die Konsistenz bidirektionaler Referenzen stellen hierbei Problemfälle dar. Aber auch die Transaktionssicherheit bei der Synchronisation lokaler Lernobjektarchive mit zentralen Lernobjektarchiven muss dabei berücksichtigt werden.

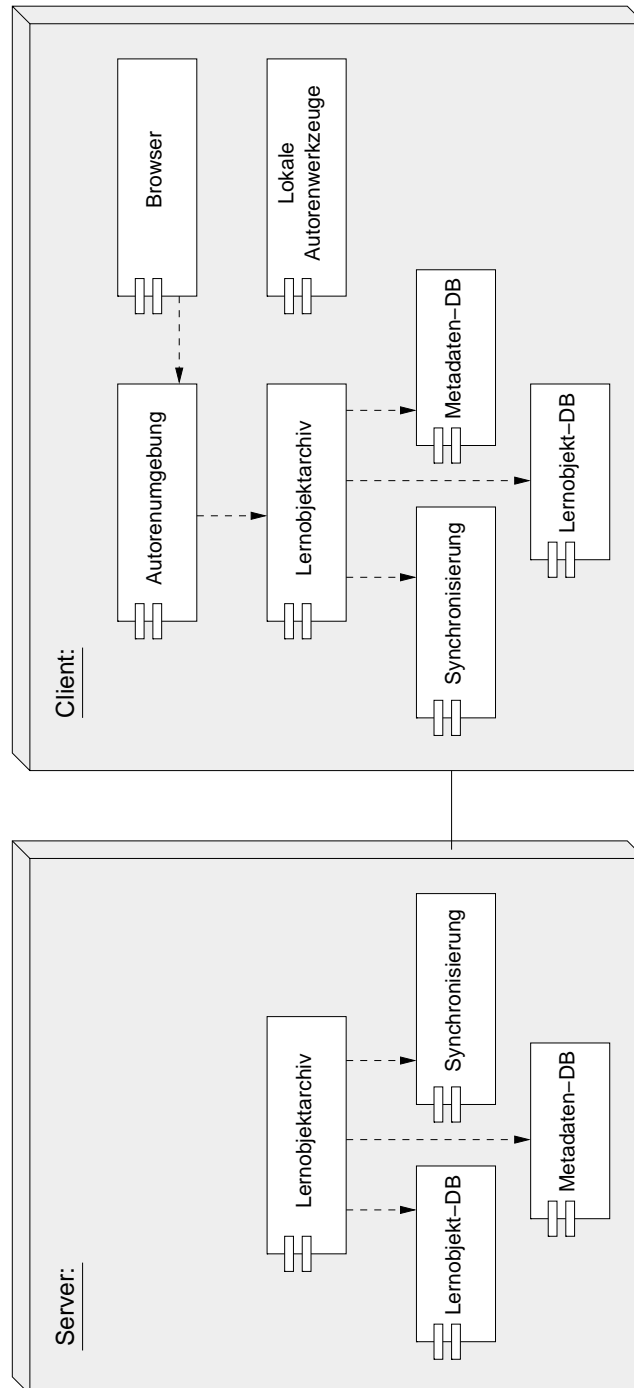


Abbildung 7.5: Autorenumgebung mit zentralem Lernobjektarchiv

Eine andere mögliche Systemarchitektur für eine Autorenumgebung, zur Aggregation und Wiederverwendung von Lernobjekten in einer Gruppe von Autoren, basiert auf einem Peer-to-Peer-Netzwerk (P2P). Dabei wird ein Peer-to-Peer-System als ein sich selbst organisierendes System gleichberechtigter, autonomer Einheiten (Peers) verstanden, das vorzugsweise ohne Nutzung zentraler Dienste auf der Basis eines Rechnernetzes mit dem Ziel der gegenseitigen Nutzung von Ressourcen operiert [SW04]. Diese Systemarchitektur kommt ohne zentralen Server aus, weil alle Funktionen des Lernobjektarchivs und zur Organisation des Overlay-Netzwerks von den lokalen Arbeitsplätzen der Autoren übernommen werden. Die Client-Systeme übernehmen damit auch Server-Funktionen. Im Vergleich besitzt diese Systemarchitektur sehr viele Ähnlichkeiten zu dem System mit zentralem Lernobjektarchiv. Im Unterschied dazu entfällt der Server ohne Ersatz und auf dem Clientsystem wird statt der Synchronisierungskomponente ein P2P-Adapter verwendet, mittels dessen Lernobjekte ausgetauscht werden können. Problematisch bei der Verwendung eines Peer-to-Peer-Netzwerks zum Austausch der Lernobjekte ist jedoch die Konsistenzhaltung der Lernobjekte, weil es in einem solchen Netzwerk keine zentrale Stelle zur Speicherung der Lernobjekte gibt und die einzelnen Peers beliebig am Netzwerk angemeldet beziehungsweise vom Netzwerk getrennt werden können.

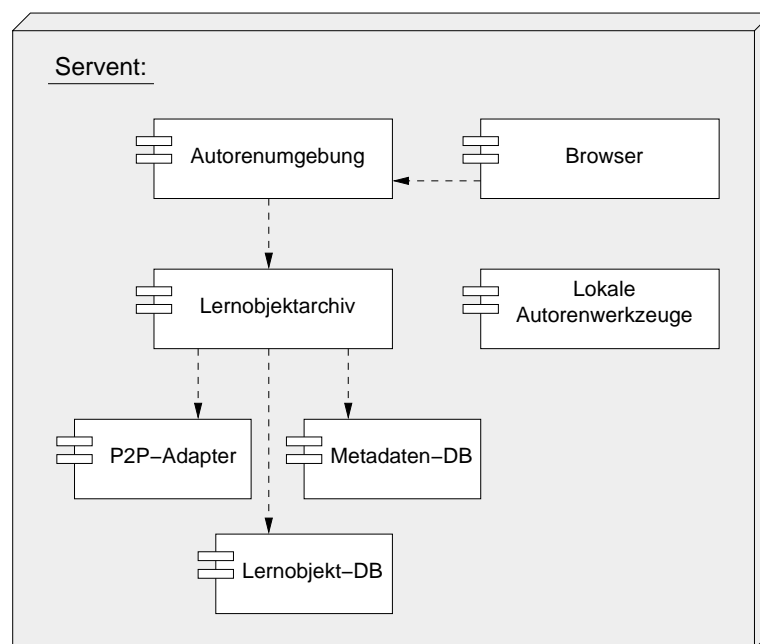


Abbildung 7.6: Autorenumgebung mit P2P-basierter Verteilung der Lernobjekte

Abbildung 7.6 zeigt die entsprechende Systemarchitektur. Das System enthält alle Komponenten des erweiterten Einzelplatzsystems. Zusätzlich enthält es die Komponente „P2P-Adapter“, mittels der Lernobjekte ausgetauscht werden können.

7.1.5 Bewertung

Alle drei Systemarchitekturen, die im Abschnitt über Mehrplatzsysteme beschrieben worden sind, erfüllen die in Kapitel 4 aufgestellten Anforderungen an ein System zur Aggregation und Wiederverwendung von Lernobjekten in einer Gruppe von Autoren. Aus diesem Grund stellen alle drei Alternativen geeignete Systemarchitekturen dar. Die beiden letzten Systemarchitekturen mit lokaler Autorenumgebung erfordern jedoch einen zusätzlichen Aufwand zur Verteilung und Synchronisation der Lernobjekte mit einem zentralen Lernobjektarchiv oder die Konsistenthaltung von Lernobjekten in einem P2P-Netzwerk. Für die Realisierung dieser Verteilungsmechanismen sind weitere forschungsrelevante Probleme zu lösen. Da in der vorliegenden Arbeit jedoch der Schwerpunkt auf der Wiederverwendung von Lernobjekten durch Aggregation liegt, wird deshalb die Systemarchitektur mit zentraler Autorenumgebung und zentralem Lernobjektarchiv, wie in Abbildung 7.4 dargestellt, als Systemarchitektur für die Implementierung des ResourceCenters gewählt. Dabei entspricht die Architektur des ResourceCenters der Server-Komponente aus Abbildung 7.4.

7.2 Lernobjektarchiv

In diesem Abschnitt werden die Grundfunktionen des Lernobjektarchivs ResourceCenter [HFSS04, HSS05][25] beschrieben. Hierfür wird zunächst das verwendete Metadatenchema inklusive der Erweiterungen zur korrekten Einbettung von Lernobjekten vorgestellt. Anschließend werden die wichtigsten Funktionen des Lernobjektarchivs beschrieben, die verwendet werden können, um WBTs nach dem Bottom-Up-Verfahren zu erstellen. Danach wird die Generierung von Metadaten beim Ablegen neuer Medienobjekte in das Lernobjektarchiv beschrieben.

7.2.1 Metadatenchema

Im Lernobjektarchiv werden alle Lernobjekte und deren Fragmente gespeichert und mit Metadaten beschrieben. Die Metadatenbeschreibungen werden dazu genutzt, Lernobjekte zu finden und hinsichtlich einer möglichen Verwendung zu beurteilen. Hierfür werden typische Metadaten über die Lernobjekte wie Autor, Titel und Zeitpunkt der Veröffentlichung gespeichert.

Wie bereits in Abschnitt 5.2.1 beschrieben, werden Lernobjekte und deren Fragmente im ResourceCenter mit IEEE Learning Object Metadata (LOM) [LOM02] des Learning Technology Standards Committees (LTSC) [12] der IEEE beschrieben. Dieses Metadatenchema ist speziell für die Beschreibung von Lernobjekten definiert worden und aufgrund der Einbettung von LOM in die SCORM-Spezifikation [Adv01] weit verbreitet und ohne Aufwand in SCORM-konforme Content-Pakete zu integrieren.

Jedes Lernobjekt wird mit genau einem Metadatensatz beschrieben. Die Metadaten nach LOM reichen jedoch nicht aus, um ein Lernobjekt im ResourceCenter vollständig zu beschreiben. Zusätzlich zu den Metadaten der Lernobjekte werden administrative Daten und Daten für die Präsentation der Lernobjekte benötigt, wie beispielsweise der Eigentümer und die Darstellungsgröße des Lernobjekts.

Metadaten für die Administration von Lernobjekten umfassen hauptsächlich Informationen die den Zugriff auf die Lernobjekte und deren Versionierung regeln. Viele dieser Informationen lassen sich in der benötigten Form nicht im Metadatenschema kodieren oder gehen über die im Schema definierten Metadaten hinaus. Aus diesem Grund werden sie separat in einer zusätzlichen Administrations-Kategorie innerhalb jedes Metadatensatzes gespeichert. Tabelle 7.1 enthält alle Datenfelder, die zusätzlich im ResourceCenter in der Administrations-Kategorie gespeichert werden. Alle Datenfelder können jeweils nur einen Wert aufnehmen.

Tabelle 7.1: Erweiterung der Metadaten um administrative Informationen der Lernobjekte

Name	Beschreibung	Wertebereich	Beispiel
Owner	Systemkennzeichner des Eigentümers	Zeichenkette	hoermann
LastModified	Datum der letzten Änderung	Millisekunden seit 00:00 1. Januar 1970 UTC	00011207 38272456
MimeType	MIME-Typ des Lernobjekts oder der Ankerdatei	MIME-Typen nach RFC2048:1996	application/ x-shockwave- flash
ResourceType	Technischer Ressourcentyp	image, chemical, flash, testitem, table, lection, course	flash
Status	Bearbeitungsstatus	draft, final	draft
Visible	Sichtbarkeit	true, false	true
Editable	Bearbeitbarkeit	true, false	true
Locked	Schreibsperre	true, false	false

Das Datenfeld „Owner“ enthält einen auf das System bezogenen Kennzeichner eines Benutzers, der der Eigentümer des Lernobjektes ist. Diese Information wird genutzt um ein einfaches Rechtemanagement der gespeicherten Lernobjekte abzubilden, bei dem nur Eigentümer ihre Lernobjekte bearbeiten dürfen. Die im LOM-

Datenschema vorgesehenen Einträge für Entitäten mit bestimmten Rollen unter *LOM.LifeCycle.Contribute* sind für diese Aufgabe nicht geeignet, weil sie lediglich vCards [DH98, HSD98] für die Identifikation der Entitäten zulassen. Diese sind nicht als Kennzeichner einer Person oder vielmehr eines Benutzers beziehungsweise Autoren zu verwenden. Das Datum der letzten Änderung wird im Datenfeld „LastModified“ eingetragen. Hierbei kommt eine Zeitdarstellung ähnlich der Unix-Zeit zur Anwendung. Es werden die Millisekunden nach dem 1. Januar 1970 00:00 UTC gezählt. Das Datenfeld „MimeType“ wird zur Speicherung des MIME-Typs genutzt. Der MIME-Typ, der in diesem Feld eingetragen wird, entspricht dem MIME-Typ des Lernobjekts oder, falls angegeben, dem der Ankerdatei (siehe präsentationsspezifische Metadaten). Dieses Datenfeld wird für die korrekte Anzeige von Lernobjekten im Web-Browser benötigt. Das Datenfeld *LOM.Technical.Format* ist für diese Aufgabe nicht geeignet, weil damit nicht nur der MIME-TYPE des Containerformats der Lernobjekt erfasst wird, sondern alle im Lernobjekt verwendeten MIME-Typen. Es ist deswegen besser dafür geeignet, festzustellen, ob alle technischen Voraussetzungen gegeben sind, um ein Lernobjekt abzuspielen. Der technische Ressourcentyp wird in dem Datenfeld „ResourceType“ gespeichert. Im Gegensatz zu dem Datenfeld *LOM.Educational.LearningResourceType*, nimmt dieses Datenfeld nur einen Wert auf, welcher im Vergleich eine weniger feingranulare Klassifizierungsmöglichkeit für Lernobjekte besitzt, jedoch zur Klassifikation von Kursen und Lektionen ein erweitertes Vokabular enthält. Die Werte dieses Datenfeldes werden für die Suche nach geeigneten Lernobjekten und zur Anzeige von Zeichen neben den Suchtreffern verwendet. Die Datenfelder „Status“ und „Visible“ werden wie in Kapitel 6 beschrieben, für die Speicherung von Bearbeitungsstatus und der Sichtbarkeit der Lernobjekte in Zusammenhang der Versionierung verwendet. Die Datenfelder „Editable“ und „Locked“ werden als Merker für die Bearbeitbarkeit von Lernobjekten und eine Schreibsperre für Lernobjekte eingesetzt.

Die technischen Informationen, die in der Kategorie „Technical“ im LOM-Datenschema gespeichert werden, enthalten hauptsächlich Informationen, anhand derer geprüft werden kann, ob das beschriebene Lernobjekt mit der verfügbaren Software abgespielt werden kann. Informationen darüber, wie Lernobjekte angezeigt werden sollen, enthält die Kategorie „Technical“ nicht. Diese sind jedoch für Lernobjekte, insbesondere Medienobjekte, besonders wichtig. Aus diesem Grund werden Informationen, wie Lernobjekte anzuzeigen sind, in einer weiteren Kategorie mit präsentationsspezifischen Metadaten gespeichert, die den Namen „Presentational“ trägt. Tabelle 7.2 beschreibt die Datenfelder dieser Kategorie. Die Spalte „Multiplizität“ enthält Angaben darüber, ob ein Datenfeld nur einen oder beliebig viele Werte aufnehmen kann.

Das Datenfeld „AnchorFile“ gibt die Hauptdatei eines Lernobjekts an, das aus mehreren Dateien besteht. Diese Lernobjekte können zusammen in einem Archiv verpackt im ResourceCenter aufgeladen werden. Die Hauptdatei gibt die Datei des Archivs an, die für die Anzeige des Lernobjekts gestartet werden muss. Alle anderen

Tabelle 7.2: Erweiterung der Metadaten um präsentationsspezifische Eigenschaften der Lernobjekte

Name	Multiplizität	Beschreibung	Wertebereich	Beispiel
AnchorFile	1	Hauptdatei einer Menge von Dateien	jede Datei einer Menge von Dateien, die gestartet werden kann.	main.swf
Width	1	Breite in Pixeln	N_0^+	360
Height	1	Höhe in Pixeln	N_0^+	240
Parameter	n	Parameter für die Parametrisierung eines Lernobjekts	Paare aus Parameternamen und Wert	bgColor=#ffffff

Dateien des Archivs werden dann in der Regel von der Hauptdatei referenziert.

Nicht immer ist die zur vollständigen Darstellung von Lernobjekten notwendige Größe des Bildschirmbereichs aus den Daten der Lernobjekte selbst zu bestimmen. Für diese Fälle können die Werte für die horizontale und vertikale Größe des Bildschirmbereichs zur Darstellung eines Lernobjekts in den Datenfeldern „Width“ und „Height“ in Pixeln gespeichert werden. Das Datenfeld „Parameter“ wird dazu verwendet Parameter zu speichern, mit denen Lernobjekte gestartet werden können. Dieses Datenfeld kann als einziges Datenfeld der Kategorie „Presentational“ mehr als einen Wert aufnehmen.

Die Beispielwerte stehen für ein Lernobjekt, das in einem Archiv verpackt in das ResourceCenter abgelegt wurde. Die Hauptdatei ist „main.swf“. Sie wird mit einer Breite von 360 und Höhe von 240 Pixeln in Web-Seiten eingebettet. Über den Parameter „bgColor=#ffffff“ wird die Hintergrundfarbe des Lernobjekts auf weiß eingestellt.

Die Werte der Datenfelder „Width“, „Height“ und „Parameter“ stellen Vorgabewerte dar, die jedes Mal, wenn ein Lernobjekt als Fragment in einem aggregierenden Lernobjekt verwendet wird, zur Parametrisierung des Lernobjektfragments in die Beschreibung des aggregierenden Lernobjekts eingetragen werden. Die aus den Metadaten kopierten Vorgabewerte können anschließend nach Bedarf für jedes Lernobjekt getrennt geändert werden. Würden die Parameter bei jeder Anwendung, wie beispielsweise einem Export eines WBTs, aus den Metadatensätzen gelesen, dann

würde die Parametrisierung von Lernobjektfragmenten zentral über den Metadatenatz erfolgen. Das würde bedeuten, dass die Parametrisierung von Lernobjektfragmenten nicht für jede Verwendung getrennt angepasst werden könnte.

Die administrativen und präsentationsspezifischen Metadaten sind nicht Teil des LOM-Datenschemas. Sie werden daher in einer Erweiterung des standardisierten Metadatenschemas für jedes Lernobjekt gespeichert. Die Erweiterung der Metadaten werden jedoch nur intern im ResourceCenter verarbeitet und gespeichert. Bei der Ausgabe der Metadaten, beispielsweise in SCORM-konforme Kurspakete, werden die Erweiterungen der Metadaten nicht in das Paket exportiert.

7.2.2 Funktionen des Lernobjektarchivs

In diesem Abschnitt werden die Basisfunktionen des Lernobjektarchivs innerhalb des ResourceCenters beschrieben. Dabei handelt es sich um das Ablegen, Abrufen und Verwalten von Lernobjekten. Diese Funktionen werden insbesondere dann benötigt, wenn Lernobjekte ohne Kontext, das heißt ohne dass sie beispielsweise durch einen Aggregationsprozess neu gebildet werden, im Lernobjektarchiv abgelegt werden sollen.

7.2.2.1 Ablegen von Lernobjekten

Lernobjektarchive dienen dazu Lernobjekte verfügbar zu machen. Hierzu müssen die Lernobjekte im Lernobjektarchiv ResourceCenter abgelegt werden. Abbildung 7.7 zeigt das Aktivitätsdiagramm für das Ablegen von Lernobjekten im Lernobjektarchiv ResourceCenter. Das Ablegen von Lernobjekten beginnt mit dem Aufladen¹ des Lernobjekts. Nach dem das Lernobjekt aufgeladen worden ist, wird es analysiert und festgestellt, ob für die Präsentation zusätzliche Metadaten benötigt werden. Können diese nicht automatisch auf Basis des Lernobjekts bestimmt werden, dann werden sie in einem Dialog abgefragt. Anschließend werden alle übrigen Metadaten nach dem in Abschnitt 5.7 beschriebenen Verfahren durch den Metadaten-Wizard weitestgehend automatisch bestimmt. Metadaten, die sich dabei nicht automatisch bestimmen lassen, werden manuell abgefragt. Anschließend erhalten die Autoren die Möglichkeit den Metadatenatz des neuen Lernobjekts im Metadateneditor zu bearbeiten. Sobald die Metadatenbeschreibung fertiggestellt ist, wird sie in der XML-Datenbank des ResourceCenters gespeichert [HJSS03, HJ03]. Ab diesem Zeitpunkt steht das Lernobjekt für die Wiederverwendung im Lernobjektarchiv zur Verfügung. Nach dem der Metadatenatz erstellt worden ist, wird den Autoren zum Abschluss des Bereitstellungsprozesses eine Zusammenfassung des Lernobjekts bestehend aus einem Auszug der Metadaten angezeigt und es werden ihnen weitere Funktionen

¹In der vorliegenden Arbeit wird für das englische Verb „to upload“ die Übersetzung „aufladen“ verwendet. Damit wird die Übertragung einer Datei von dem Client-System auf das Server-System bezeichnet.

angeboten, die auf dem Lernobjekt ausgeführt werden können. Für den Fall, dass das Lernobjekt in eine Web-Seite eingebettet werden kann, wird es zusätzlich zu dem Metadatenauszug direkt in der Zusammenfassung angezeigt.

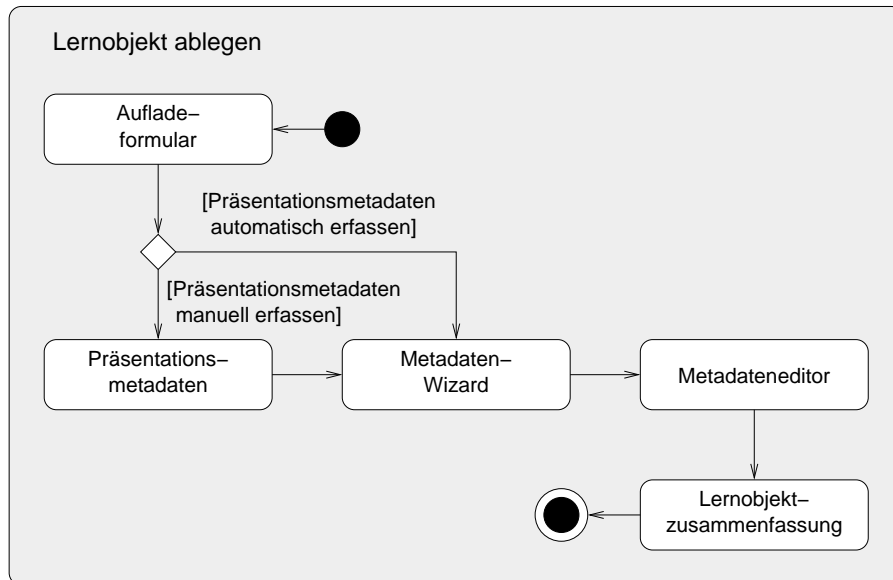


Abbildung 7.7: Ablegen von Lernobjekten im Lernobjektarchiv

Im Folgenden werden die einzelnen Aktivitäten für das Ablegen von Lernobjekten im Lernobjektarchiv mit Hilfe von Screenshots dokumentiert.

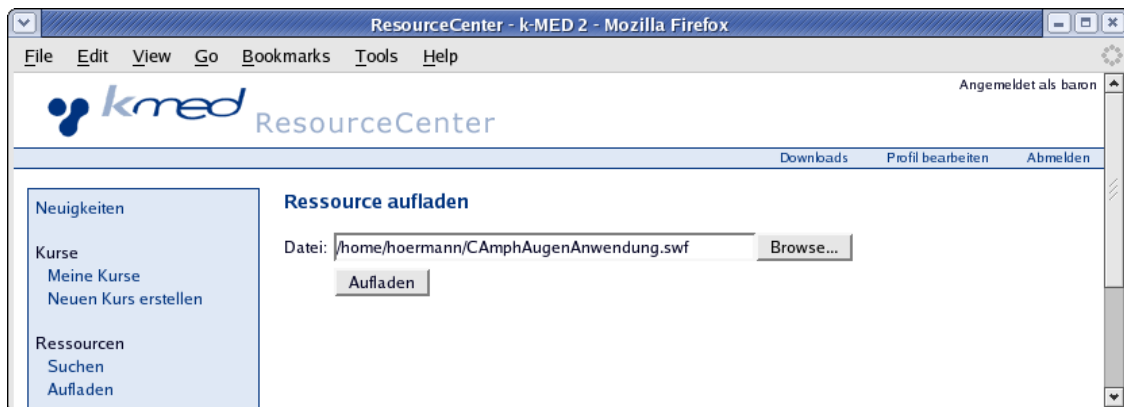


Abbildung 7.8: Aufladeformular für das Ablegen neuer Lernobjekte

Abbildung 7.8 zeigt das Aufladeformular zum Ablegen der Lernobjekte. Das Ablegen von Lernobjekten wird über einen Mausklick auf die Schaltfläche „Ressource aufladen“ links in der Navigationsleiste gestartet. Bevor mit Hilfe des Aufladeformulares ein Lernobjekt aufgeladen werden kann, muss es ausgewählt werden. Dieses

erfolgt unter Verwendung des Dateidialogs, der durch Klicken auf die Schaltfläche „Browse“ geöffnet wird. Ist das Lernobjekt ausgewählt, kann es mit einem Mausklick auf die Schaltfläche mit der Aufschrift „Aufladen“ in das ResourceCenter aufgeladen werden.

Die Analyse der ausgewählten Datei ergibt im gezeigten Beispiel, dass die Größe, mit der das Lernobjekt angezeigt werden soll, manuell erfasst werden muss. Aus diesem Grund erfolgt im nächsten Schritt die manuelle Erfassung der präsentationsspezifischen Metadaten, die nicht automatisch bestimmt werden konnten, wie in Abbildung 7.9 gezeigt.

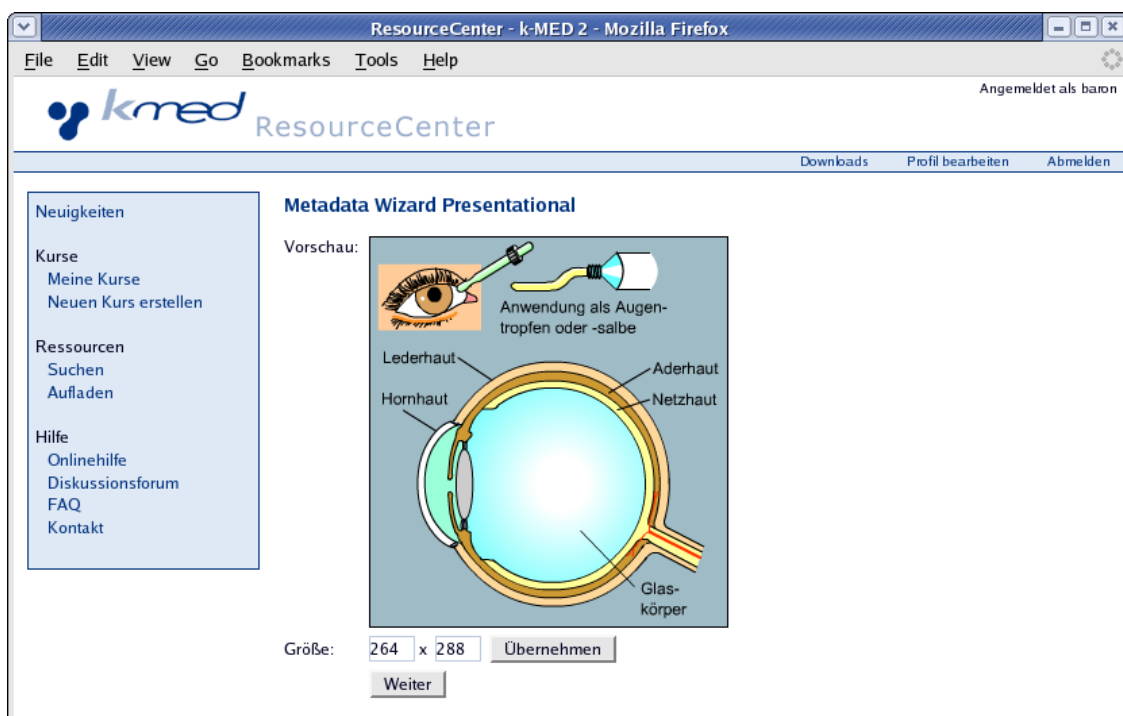


Abbildung 7.9: Beschreibung präsentationsspezifischer Metadaten

Zur Kontrolle der eingegebenen Parameter erfolgt eine Vorschau des Lernobjekts mit den gewählten Parametern. Die Vorschau wird durch Mausklick auf die Schaltfläche mit der Beschriftung „Übernehmen“ aktualisiert. Durch einen Mausklick auf die Schaltfläche „Weiter“ wird der Dialog zur Eingabe der präsentationsspezifischen Metadaten beendet.

Danach wird versucht die übrigen Metadaten weitestgehend automatisch zu bestimmen. Da das ResourceCenter über keine Verfahren zur Inhaltsanalyse von Lernobjekten verfügt, bleiben mindestens die Datenfelder „LOM.General.Title“, „LOM.General.Description“ und „LOM.General.Keywords“ unbestimmt. Diese werden im nächsten Dialog des Metadaten-Wizards manuell abgefragt.

The screenshot shows a web browser window titled "ResourceCenter - k-MED 2 - Mozilla Firefox". The page header includes the "kmed ResourceCenter" logo and the text "Angemeldet als baron". A navigation bar contains links for "Downloads", "Profil bearbeiten", and "Abmelden".

The main content area is titled "Metadaten-Wizard" and contains the following form fields:

- Metadatensprache:** A dropdown menu with "de" selected.
- Inhaltssprache:** A dropdown menu with "de" selected.
- Titel:** A text input field containing "Applikation von Chloramphenicol am Auge".
- Beschreibung:** A text area containing "Applikation von Chloramphenicol am Auge: Zum Anzeigen die Bildelemente mit dem Cursor überfahren."
- Schlüsselwort:** A text input field containing "Chloramphenicol".
- Ressourcentyp:** A dropdown menu with "Bild" selected.

A "Weiter" button is located at the bottom of the form. On the left side of the page, there is a sidebar menu with categories: "Neuigkeiten", "Kurse" (with sub-items "Meine Kurse" and "Neuen Kurs erstellen"), "Ressourcen" (with sub-items "Suchen" and "Aufladen"), and "Hilfe" (with sub-items "Onlinehilfe", "Diskussionsforum", "FAQ", and "Kontakt").

Abbildung 7.10: Vereinfachte Eingabemaske des Metadaten-Wizards zur Abfrage nicht automatisch bestimmter Metadaten

Abbildung 7.10 zeigt den Dialog des Metadaten-Wizards zur manuellen Eingabe der nicht automatisch zu bestimmenden Metadaten. Neben den genannten Datenfeldern für Titel, Beschreibung und Schlüsselwörter, werden die Datenfelder „LOM.General.Language“, „LOM.MetaMetadata.Language“ und „LOM.Educational.LearningResourceType“ abgefragt. Dabei enthalten die Datenfelder zur Bestimmung der Sprachen für Lernobjekte und Metadatensatz bereits Standardwerte, die aus dem Benutzerprofil stammen. Für den Typ des Lernobjekts steht im Metadaten-Wizard bereits auf Basis des Lernobjektyps eine reduzierte Auswahlliste bereit. Manuell sind dementsprechend die Datenfelder für Titel, Beschreibung, Schlüsselwort und den Typ des Lernobjekts anzugeben. Die Vorgabewerte der Sprachen können in der Regel übernommen werden.

Mit Hilfe der automatisch erzeugten und manuell eingegebenen Metadaten erstellt das ResourceCenter jetzt einen LOM-konformen Metadatensatz. Dieser Metadatensatz enthält alle durch die SCORM-Spezifikation als obligatorisch definierten Datenfelder und darüber hinaus Datenfelder, die sich ohne großen Aufwand automatisch bestimmen lassen. Im nächsten Arbeitsschritt kann der erstellte Metadatensatz manuell nachbearbeitet werden.

Abbildung 7.11 zeigt den Metadateneditor zum Bearbeiten des Metadatensatzes, der für das abzulegende Lernobjekt erstellt worden ist. Mit Hilfe des Metadateneditors können sechs der neun Kategorien des LOM-Schemas und die Kategorie mit den präsentationsspezifischen Metadaten bearbeitet werden. Für jede Kategorie ist eine Eingabemaske vorgesehen. Durch einen Mausklick auf die Karteikartenreiter

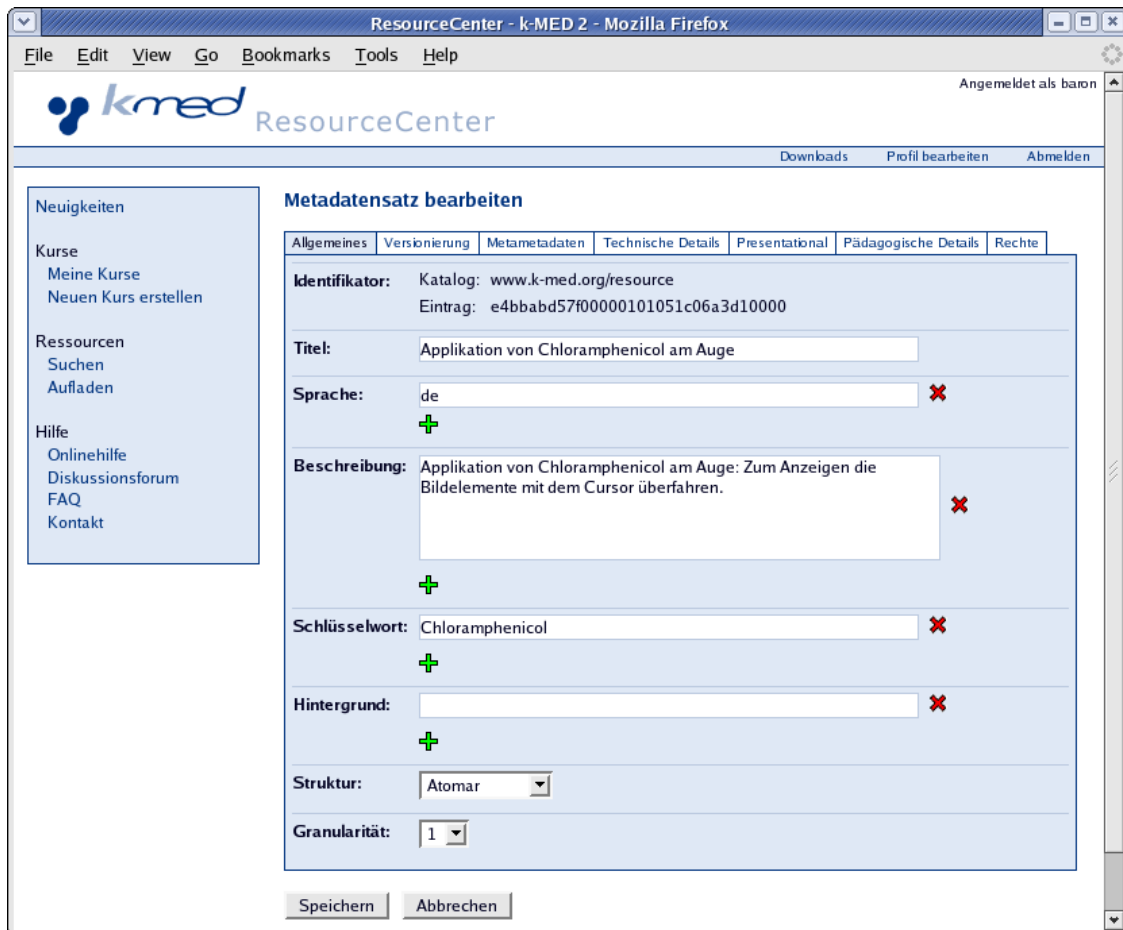


Abbildung 7.11: Metadateneditor zur Bearbeitung der generierten Metadaten

kann jeweils eine dieser Eingabemasken angezeigt werden. In der Abbildung ist die Kategorie mit allgemeinen Metadaten des Lernobjekts aktiv. Die Kategorie „Relation“ wird vom ResourceCenter vollkommen automatisch behandelt, eine direkte Visualisierung der in den Metadatenätzen gespeicherten Relationen ist nicht vorgesehen. Da es für die Kategorien „Annotation“ und „Classification“ keine Verwendung in k-MED gibt, werden sie bisher noch nicht vom Metadateneditor des ResourceCenters unterstützt.

Nachdem die Bearbeitung des Metadatenatzes im Metadateneditor abgeschlossen ist, wird das Lernobjekt zusammen mit dem Metadatenatz im ResourceCenter abgelegt. Ab diesem Zeitpunkt kann das abgelegte Lernobjekt abgerufen und wiederverwendet werden. Im letzten Arbeitsschritt wird für das neu abgelegte Lernobjekt eine Zusammenfassung angezeigt, die dazu dient einen Teil der Metadaten zu überprüfen, gegebenenfalls Korrekturen durchzuführen und verschiedene Aktionen mit dem neu aufgeladenen Lernobjekt durchzuführen.

Abbildung 7.12: Zusammenfassung des neu abgelegten Lernobjekts

Abbildung 7.12 zeigt die Zusammenfassung für das neu abgelegte Lernobjekt. Im oberen Bereich werden Titel, Beschreibung und Schlüsselwörter des Metadatensatzes angezeigt. Das Stift- und Lupensymbol, das sich neben den Metadaten befindet, kann genutzt werden, um die Metadaten des Lernobjekts zu bearbeiten beziehungsweise zu betrachten. Unterhalb des Auszugs aus den Metadaten des Lernobjekts befinden sich Schaltflächen für Aktionen, die auf dem Lernobjekt ausgeführt werden können. Falls sich das Lernobjekt direkt in HTML-Seiten einbetten lässt, so dass es direkt betrachtet werden kann, wird das Lernobjekt in der Zusammenfassung angezeigt.

7.2.2.2 Abrufen von Lernobjekten

Durch das Abrufen von Lernobjekten können im Lernobjektarchiv des ResourceCenters gespeicherte Lernobjekte wiederverwendet werden. Hierzu müssen die gespeicherten Lernobjekte jedoch zunächst gefunden werden können. Das ResourceCenter bietet zum Finden geeigneter Lernobjekte eine auf den Metadaten der Lernobjekte basierende Suche an.

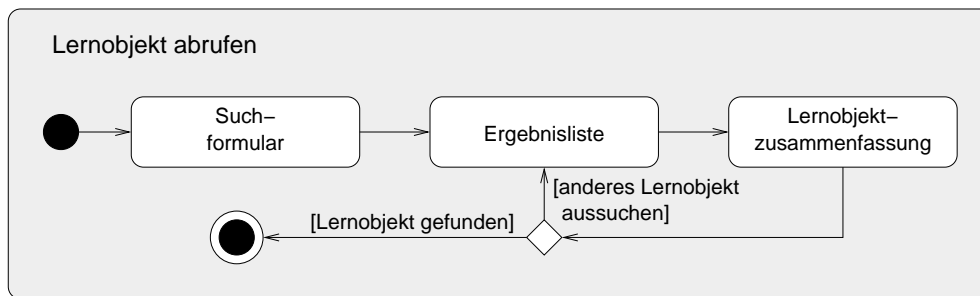


Abbildung 7.13: Abrufen verfügbarer Lernobjekte

Das Aktivitätsdiagramm für das Abrufen von Lernobjekten ist in Abbildung 7.13 dargestellt. Für das Abrufen müssen die entsprechenden Lernobjekte jedoch zunächst gefunden werden. Die erste Aktivität zum Abrufen von Lernobjekten stellt demnach das Suchen nach geeigneten Lernobjekten dar. Innerhalb der Aktivität „Suchformular“ wird eine Suchanfrage formuliert, mittels der das Archiv des ResourceCenters durchsucht wird. Lernobjekte deren Metadatenbeschreibung auf diese Suchanfrage passen, werden anschließend in einer Ergebnisliste dargestellt. Aus dieser Ergebnisliste kann ein Lernobjekt ausgewählt werden. Dieses wird in der Lernobjektzusammenfassung angezeigt, die bereits als letzte Aktivität beim Ablegen von Lernobjekten vorgestellt wurde. Handelt es sich bei dem ausgewählten Lernobjekt um das gesuchte Lernobjekt, ist das Ziel erreicht und die eigentliche Verwendung des Lernobjekts beginnt. Andernfalls kann über die Ergebnisliste ein weiteres Lernobjekt ausgewählt werden.

Das Abrufen von Lernobjekten wird im ResourceCenter durch Klicken auf die Schaltfläche „Ressourcen Suchen“ gestartet. Danach erhält der Benutzer die Möglichkeit im Suchformular die Suche nach verfügbaren Lernobjekten zu formulieren. Im ResourceCenter stehen hierfür zwei verschiedene Suchformulare zu Verfügung.

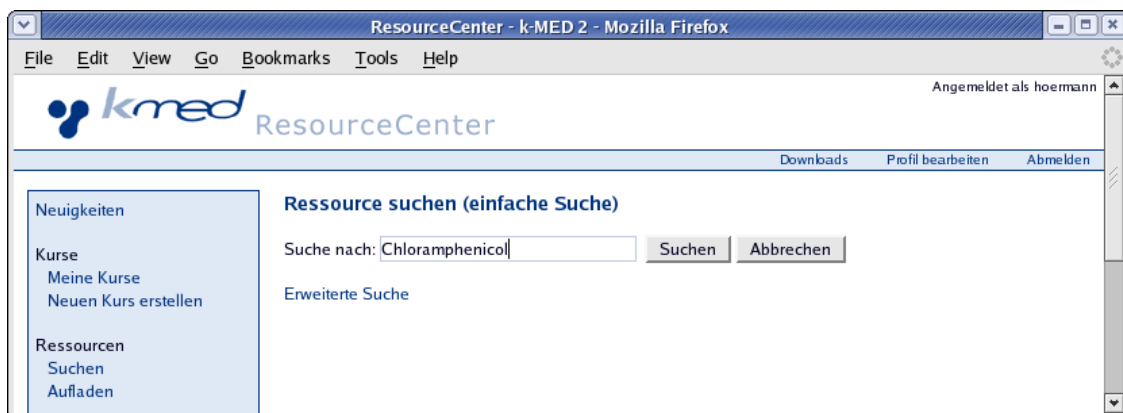


Abbildung 7.14: Suchformular zur Suche verfügbarer Lernobjekte

Abbildung 7.14 zeigt das Formular für die vereinfachte Formulierung zur Suche nach verfügbaren Lernobjekten. Das Textfeld des Formulars nimmt beliebig viele Begriffe auf, die bei der Suche mit logischen UND verknüpft werden. Das heißt alle angegebenen Begriffe müssen in einem Metadatensatz eines Lernobjekts enthalten sein, damit das Lernobjekt in der Ergebnisliste erscheint. Begriffe, die ein Leerzeichen enthalten, werden wie bei Internet-Suchmaschinen in Anführungsstriche gesetzt. Die eingegebenen Begriffe werden im Titel, der Beschreibung und den Schlüsselwörtern der Lernobjekte gesucht. Mit der optional verfügbaren „Erweiterten Suche“ kann ausgewählt werden, in welchen Metadatenfeldern die Begriffe gesucht werden. Darüberhinaus kann die Suche auf Lernobjekte bestimmter Autoren eingegrenzt werden. Die Angabe des Typs des gesuchten Lernobjekts und des Zeitraums, in dem das Lernobjekt das letzte Mal aktualisiert wurde, schränkt den Suchraum weiter ein.



Abbildung 7.15: Ergebnisliste gefundener, verfügbarer Lernobjekte

Lernobjekte, auf die die Suchkriterien der einfachen oder der erweiterten Suche passen, werden in einer Ergebnisliste aufgelistet. Abbildung 7.15 zeigt die Auflistung der verfügbaren Lernobjekte, die mit Hilfe des Begriffs „Chloramphenicol“ unter Verwendung der einfachen Suche in der k-MED-Instanz des ResourceCenters gefunden wurden. Das erste Element der Ergebnisliste ist das im vorherigen Beispiel abgelegte Lernobjekt. Die einzelnen Einträge der Liste setzen sich aus verschiedenen Elementen zusammen. Der Typ jedes gefundenen Lernobjekts wird mit Hilfe eines Piktogramms visualisiert. Anschließend erfolgt ein Auszug aus den Metadaten der Lernobjekte. Es werden Titel, Beschreibung und Schlüsselwörter angegeben. Dabei ist der Titel als Schaltfläche realisiert, über die das entsprechende Lernobjekt

ausgewählt werden kann.

Durch Mausklick auf den Titel eines Lernobjekts in der Ergebnisliste, wird die Zusammenfassung des entsprechenden Lernobjekts angezeigt. Abbildung 7.12 zeigt die Zusammenfassung des ausgewählten Lernobjekts. Im Unterschied zur Ergebnisliste enthält die Zusammenfassung des Lernobjekts die vollständigen Metadatenfelder „Titel“, „Beschreibung“ und „Schlüsselwörter“, die in der Ergebnisliste teilweise nur in gekürzter Form angezeigt werden können. Die Zusammenfassung eines Lernobjekts enthält somit alle Informationen, die benötigt werden, um zu entscheiden, ob es das betreffende Lernobjekt ist, das heruntergeladen oder im Autorenprozess verwendet werden soll. Falls das gefundene Lernobjekt nicht passt, kann über die Ergebnisliste ein anderes Lernobjekt ausgewählt werden.

7.2.2.3 Verwalten von Lernobjekten

Die Lernobjektzusammenfassung ist jeweils die letzte Aktivität in den Aktivitätsdiagrammen für das Ablegen und Abrufen von Lernobjekten. Die Lernobjektzusammenfassung dient der Darstellung von Lernobjekten und stellt den zentralen Wartungspunkt jedes im ResourceCenter gespeicherten Lernobjekts dar. Ausgehend von diesem Punkt haben Autoren, je nach verfügbaren Rechten, die Möglichkeit Metadaten des Lernobjekts einzusehen, zu bearbeiten oder das Lernobjekt anzuzeigen, es herunterzuladen oder es zu löschen.

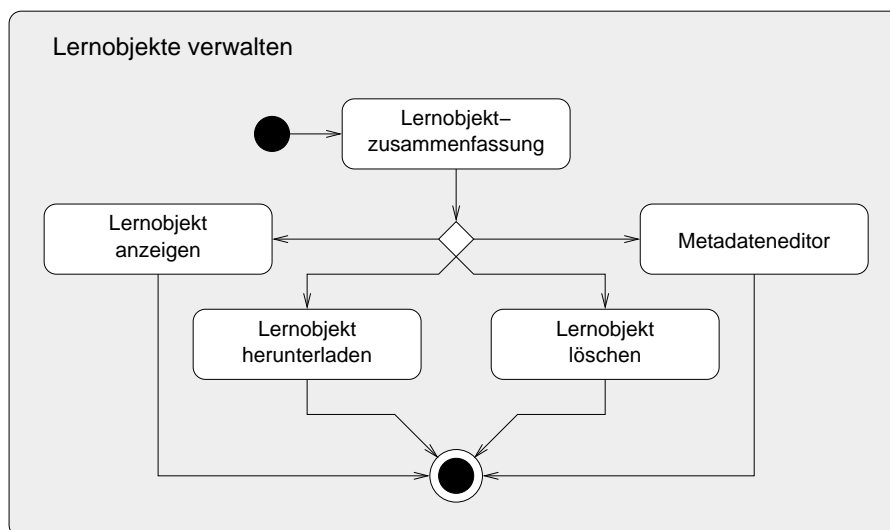


Abbildung 7.16: Lernobjekte im ResourceCenter verwalten

Abbildung 7.16 zeigt das Aktivitätsdiagramm für das Verwalten von Lernobjekten aus der Lernobjektzusammenfassung heraus. Das Aktivitätsdiagramm sieht das Anzeigen, Herunterladen und Löschen von Lernobjekten vor. Alternativ können aber auch die Metadaten des Lernobjekts bearbeitet werden.

http://localhost:8080 - Course Preview - Mozilla Firefox

Spezielle Antibiotika (Vorschau)

Inhaltsverzeichnis anzeigen

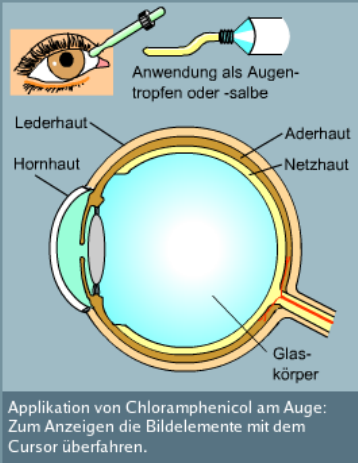
3 Anwendung von Chloramphenicol

Indikationen:

1. Schwere bzw. lebensgefährliche Salmonellosen (Typhus, Paratyphus, Salmonellen-Meningitis)
2. lebensbedrohliche Augeninfektionen, falls die risikoärmeren Antibiotika (v.a. Chinolone) unwirksam oder kontraindiziert sind
3. u.U. bei Hirnabszeß, Rickettsiose (Fleckfieber), Brucellose.

Dosierung:

Bei Erwachsenen: 1,5 - 4 g/d, in 3 bis 4 Gaben (stets Blutbildkontrolle!); bei Kindern und Säuglingen: 50-80 mg/kg/d (Blutbild- und möglichst Blutspiegelkontrolle!); bei Neugeborenen nur bis höchstens 50 mg/kg/d. (Vorsicht vor Akkumulation des Pharmakons in der unreifen Leber und einem daraus



Applikation von Chloramphenicol am Auge:
Zum Anzeigen die Bildelemente mit dem Cursor überfahren.

Abbildung 7.17: Player für WBTs

Das Anzeigen von Lernobjekten kann nicht immer in der Lernobjektzusammenfassung geschehen, da beispielsweise ganze Kurse oder Lektionen nicht direkt in die Lernobjektzusammenfassung eingebettet werden können. Aus diesem Grund befinden sich in den Lernobjektzusammenfassungen von Kursen und Lektionen Schaltflächen durch deren Betätigung die entsprechenden Lernobjekte in einem separaten Fenster angezeigt werden können. Abbildung 7.17 zeigt den Player zum Anzeigen der Lektion, die sich unter anderem aus dem Lernobjekt zusammensetzt, dessen Ablage im vorherigen Beispiel beschrieben wurde.

7.2.2.4 Versionierung von Lernobjekten

Bei den bisherigen Betrachtungen der Basisfunktionen des ResourceCenters wurde die Versionierung von Lernobjekten in dem Lernobjektarchiv vernachlässigt. Für ihre Betrachtung werden die zuvor vorgestellten Funktionen in einem Diagramm zusammengefasst. Abbildung 7.18 zeigt das Aktivitätsdiagramm der Basisfunktionen des ResourceCenters für das Ablegen, Abrufen und Verwalten von Lernobjekten. Zur Vereinfachung fehlt in diesem Diagramm die Möglichkeit von der Lernobjektzusammenfassung zurück zur Ergebnisliste zu gelangen. Diese Kante wurde aus dem Diagramm entfernt, weil sie nur dann verwendet werden kann, wenn die Lernob-

jektzusammenfassung über die Ergebnisliste erreicht wurde.

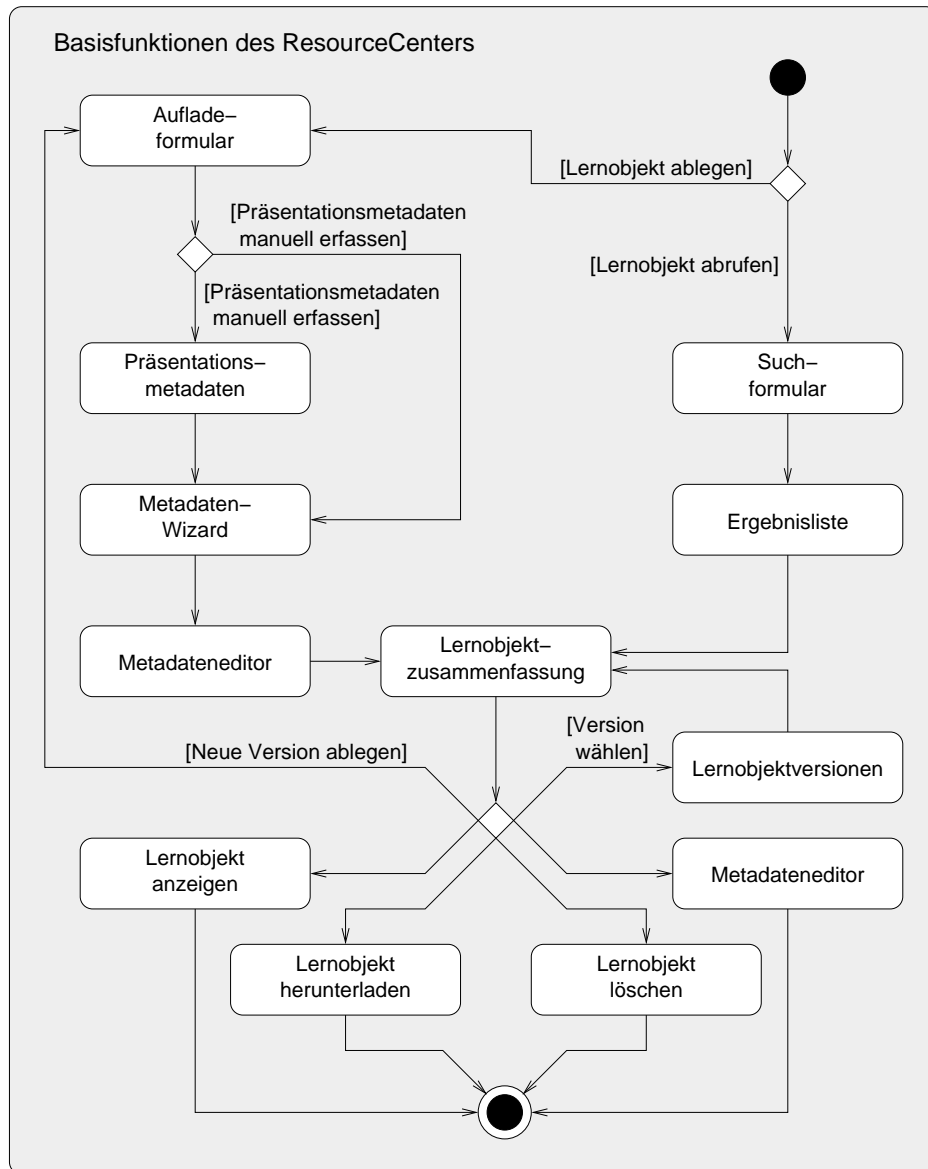


Abbildung 7.18: Basisfunktionen des ResourceCenters als Lernobjektarchiv

Abbildung 7.18 enthält neben den zuvor dargestellten Basisfunktionen zusätzliche Funktionen zur Versionierung von Lernobjekten im Lernobjektarchiv. Die Erweiterung zur Versionierung der Lernobjekte umfasst eine Tabelle, mit deren Hilfe die Historie von Lernobjekten verfolgt werden kann. Diese Tabelle kann ausgehend von der Zusammenfassung eines Lernobjekts abgerufen werden. Über die verschiedenen Einträge der Historientabelle können verschiedene Versionen eines Lernobjekts abgerufen werden. Diese werden dann jeweils in der Zusammenfassung

der Lernobjekte angezeigt. Ausgehend von der Zusammenfassung der Version eines Lernobjekts können, soweit verfügbar, die im vorangehenden Abschnitt beschriebenen Aktionen auf der ausgewählten Version des Lernobjekts ausgeführt werden. Für das Ablegen einer neuen Version eines Lernobjekts muss zunächst der Zusammenhang zwischen den beiden Versionen des Lernobjekts hergestellt werden. Ein geeignetes Vorgehen hierfür ist das Ablegen neuer Versionen aus der Zusammenfassung eines Lernobjekts heraus. Der Zusammenhang kann so zwischen dem im Lernobjektarchiv gespeicherten und dem neu aufzuladenden Lernobjekt hergestellt werden.

7.3 Autorenumgebung

In diesem Abschnitt wird die auf Aggregation basierende Autorenumgebung des ResourceCenters beschrieben, die entsprechend der in Kapitel 4 genannten Anforderungen und entwickelten Konzepte implementiert wurde. Zunächst wird hierfür das Datenmodell für Kurse, die mit Hilfe der Autorenumgebung erstellt werden, als Implementierung der in Abschnitt 4.2.5 geforderten Erweiterung des in Abschnitt 4.2 vorgestellten Dokumentmodells beschrieben. Anschließend werden die Editor-komponenten der Autorenumgebung zur Bearbeitung von Kursen, Lektionen und Blöcken an Hand von Ableitungen des in Abschnitt 4.3.2 aufgestellten abstrakten Prozessmodells und Screenshots der laufenden Autorenumgebung beschrieben. Danach werden verschiedene Möglichkeiten für das Einfügen beziehungsweise Wiederverwenden von Lernobjekten in der Autorenumgebung betrachtet. Zum Schluss erfolgt eine Betrachtung des Zusammenspiels der einzelnen Editorkomponenten beim Erstellen eines neuen Kurses.

7.3.1 Datenmodell

In Abschnitt 4.2 wurde ein Dokumentmodell beschrieben, das geeignet ist, modularisierte Lernobjekte bis hin zu WBTs zu aggregieren. In diesem Dokumentmodell werden die Lernobjekte aller Aggregationsniveaus als „Blackbox“, das heißt als beliebiges binäres Objekt, betrachtet. Die Aggregationsbeziehungen zwischen den Lernobjekten werden unabhängig von deren Inhalt und deren Datenformat in den Metadaten gespeichert. Das erleichtert die Verwaltung der Lernobjekte, weil zum Feststellen der Aggregationsbeziehungen zwischen den Lernobjekten statt der eigentlichen Daten nur die Metadaten der Lernobjekte gelesen und interpretiert werden müssen. Die externe Beschreibung der Aggregationsbeziehungen mittels Metadaten reicht jedoch nicht aus, um die Aggregation von WBTs vollständig zu beschreiben. Verschiedene Gründe hierfür wurden in Abschnitt 4.2.5 beschrieben.

Im ResourceCenter werden aus diesem Grund zusätzlich zu dem in Abschnitt 4.2 eingeführten Dokumentmodell verschiedene XML-Schemata für die Beschreibung

der aggregierten Lernobjekte verwendet. Für die Beschreibung der in Abschnitt 4.2.4 eingeführten Lernobjektklassen werden im ResourceCenter zwei XML-Schemata für die Beschreibung von Kursen und von Lektionen verwendet.

Nach dem Aggregationsmodell von Abbildung 4.7 werden von der Beschreibung der Kurse die auf höchster Ebene liegenden Lektionen referenziert, aus denen sich der Kurs zusammensetzt. Untergeordnete Lektionen werden von den Lektionen referenziert, denen sie untergeordnet sind. Das im ResourceCenter verwendete XML-Schema unterstützt die Aggregation der Lektionen eines Kurses nach genau diesem Schema. Zusätzlich wird im ResourceCenter eine spezielle Form von Lektionen verwendet, die Elemente zur Wissensdiagnose enthalten. Sie werden „Testitems“ genannt und können ebenso wie Lektionen in Kurse eingebettet werden. Im Folgenden werden deswegen nur noch Lektionen betrachtet. Testitems enthalten im Vergleich zu Lektionen Erweiterungen, mit denen es möglich ist, in exportierten Kursen Punktestände über das in SCORM spezifizierte Run Time Environment (RTE) an das Lernsystem zu übertragen, das den Kurs abspielt.

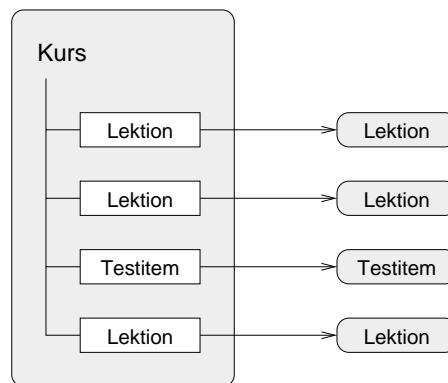


Abbildung 7.19: Aufbau von Kursen im ResourceCenter

Abbildung 7.19 zeigt den schematischen Aufbau von Kursen im ResourceCenter. Dabei stellen die Rechtecke mit den abgerundeten Ecken Lernobjekte dar. Rechtecke innerhalb der Rechtecke mit den abgerundeten Ecken stellen Elemente der aggregierten Lernobjekte dar, die mit dem Datenmodell beschrieben werden sollen. Das große grau hinterlegte Rechteck ist ein Kurs. In der Beschreibung des Kurses werden die Lektionen und Testitems, aus denen sich der Kurs zusammensetzt, referenziert. Dabei referenzieren die im ResourceCenter gespeicherten Kurse nur die auf höchster Ebene liegenden Lektionen und Testitems.

Abbildung 7.20 zeigt das XML-Schema für die Erstellung XML-basierter Kursbeschreibungen. Das Wurzelement der XML-Dateien zur Beschreibung von Kursen ist das Element „course“. Es nimmt eine beliebigen Anzahl von „lection“- und „testitem“-Elementen in beliebiger Reihenfolge auf. Diese entsprechen den Lektionen und Testitems des Kurses auf höchster Ebene. Jedes „lection“- oder „testitem“-

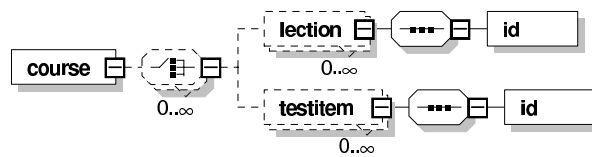


Abbildung 7.20: XML-Schema für Kurse im ResourceCenter

Element enthält einen Identifikator, mittels der die entsprechende Lektion oder das entsprechende Testitem referenziert wird.

Für die Beschreibung von Lektionen der Kurse im ResourceCenter ist ein weiteres XML-Schema entwickelt worden. Es beschreibt den inhaltlichen Aufbau der Lektionen und referenziert die untergeordneten Lektionen.

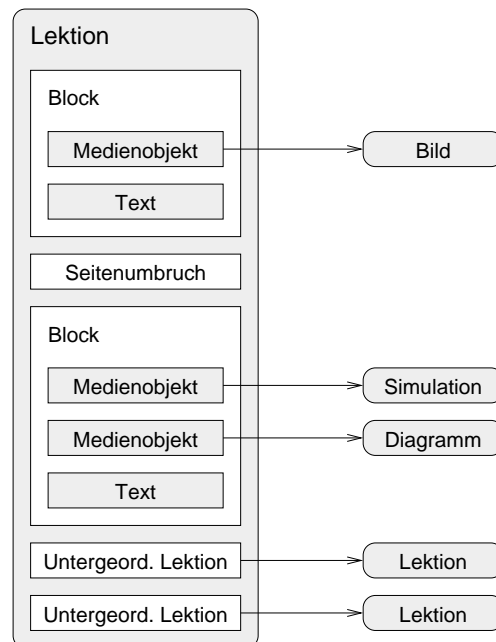


Abbildung 7.21: Aufbau von Lektionen im ResourceCenter

Abbildung 7.21 zeigt den schematischen Aufbau der Lektionen eines Kurses im ResourceCenter. Analog zu dem Aggregationsschema aus Abbildung 4.7 setzt sich eine Lektion aus einer Reihe von Blöcken zusammen, die zusätzlich durch Seitenumbrüche unterbrochen werden kann. Diese Seitenumbrüche werden je nach gewähltem Layout unter Umständen nicht berücksichtigt. Zum Beispiel werden Seitenumbrüche bei der Ausgabe einer Druckversion des Kurses aus technischen Gründen nicht berücksichtigt. Da die Blöcke und Textfragmente nicht als indizierbare Lernobjekte im Lernobjektarchiv gespeichert werden, schließt das XML-Schema ihre Beschreibung mit ein. Aus diesem Grund werden sie nicht von der Beschreibung einer Lektion referenziert, sondern in die Beschreibung der Lektion direkt eingebettet. Die

Blöcke setzen sich ihrerseits aus einer beliebigen Anzahl von Medienobjekten und bei Bedarf einem Text zusammen. Medienobjekte wie Bilder, Diagramme und Simulationen werden von den Lektionen als Lernobjektfragmente referenziert. Blöcke werden im Dokumentmodell dazu verwendet Medienobjekte und Text zu Sinneinheiten zusammenzustellen, vergleiche Abschnitt 4.2.4. Nach der inhaltlichen Beschreibung der Lektion erfolgt die Referenzierung der untergeordneten Lektionen und Testitems.

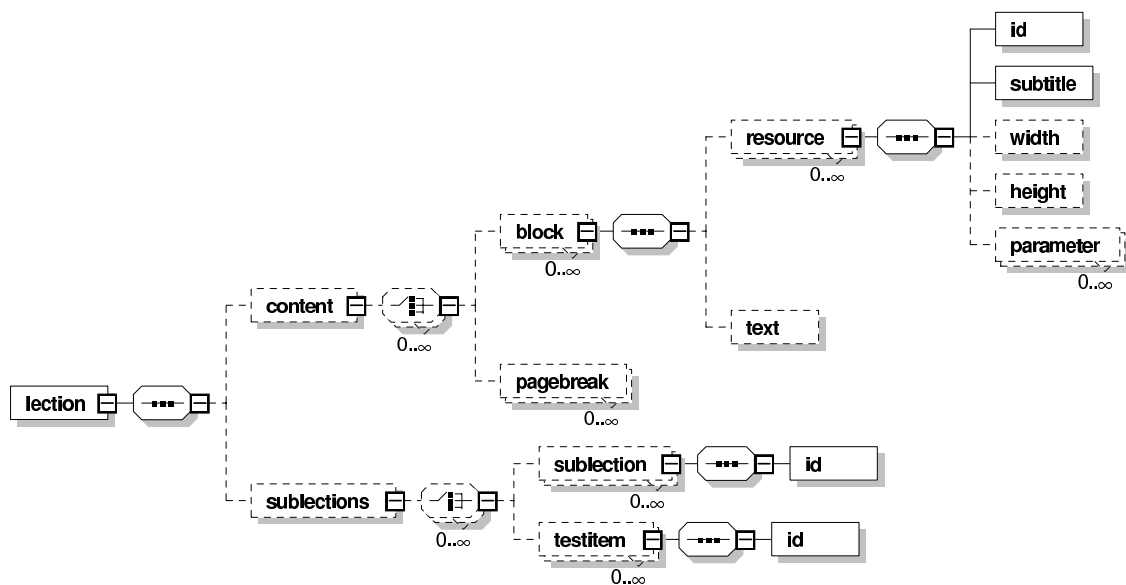


Abbildung 7.22: XML-Schema für Lektionen im ResourceCenter

Abbildung 7.22 visualisiert das XML-Schema für Lektionen im ResourceCenter. Das Wurzelement für Lektionen ist das Element „`lesson`“. Dieses kann die Elemente „`content`“ und „`sublections`“ enthalten. Dabei beinhaltet das Element „`content`“ die inhaltliche Beschreibung der Lektionen während das Element „`sublections`“ die untergeordneten Lektionen und Testitems referenziert. Das Element „`content`“ kann eine beliebige Menge von Blöcken enthalten, zwischen die jeweils ein Element „`pagebreak`“ eingefügt werden kann, um Seitenumbrüche zwischen Blöcken zu realisieren. Das Element zum Einfügen von Seitenumbrüchen enthält keine weiteren Elemente. Blöcke setzen sich dagegen aus einer beliebigen Anzahl von Medienobjekten und bei Bedarf aus einem Text zusammen. Medienobjekte werden mit Hilfe des Elements „`resource`“ eingefügt. Zur Referenzierung des entsprechenden Medienobjekts besitzt das Element „`resource`“ das Element „`id`“, das den Identifikator des entsprechenden Medienobjekts aufnimmt. Das Element „`subtitle`“ enthält den Untertitel der referenzierten Medienobjekte. Zusätzlich enthält das Element „`resource`“ die Elemente „`width`“, „`height`“ und „`parameter`“ die, wie in Abschnitt 7.2.1 beschrieben, optional verwendet werden können, um präsentationsbezogene Angaben in den Metadaten

zu überschreiben. Parallel zu den Medienobjekten kann ein Block ein Textobjekt enthalten. Der Text wird im Element „text“ abgelegt. Der Text wird mit einer stark eingeschränkten Menge von HTML-Tags ausgezeichnet, so dass übliche Web-Editoren im ResourceCenter zum Erstellen von Texten verwendet werden können, die auch von Computer-Laien bedient werden können, ohne dass dadurch der Nachteil von nicht mehr maschinell zu interpretierenden Texten entsteht. Das Element „sublections“ kann eine beliebige Anzahl der Elemente „sublection“ und „testitem“ enthalten. Mit ihrer Hilfe werden die untergeordneten Lektionen und Testitems referenziert. Jedes der beiden Elemente besitzt zu diesem Zweck jeweils ein Element „id“ das den Identifikator des entsprechenden Lernobjekts aufnimmt.

Die Umsetzung des Dokumentmodells im Datenmodell zur Beschreibung von Kursen und Lektionen im ResourceCenter ermöglicht die Wiederverwendung von Lernobjekten verschiedener Aggregationsniveaus auf verschiedenen Arten. Dabei kann fast jedes Lernobjekt, das für sich im Lernobjektarchiv abgelegt wird, durch Referenzierung in dem auf Aggregation basierenden Autorenprozess bei der Erstellung neuer Lernobjekte wiederverwendet werden. Dazu gehören Lektionen und Medienobjekte, nicht jedoch Texte. Textfragmente sind davon ausgenommen, weil sie bei einzelner Wiederverwendung in der Regel angepasst werden müssen, so dass eine Wiederverwendung durch Referenz nicht erforderlich wird. Prinzipiell können Textfragmente auch durch Referenz wiederverwendet werden, dies ist jedoch in der Implementierung des ResourceCenters nicht vorgesehen. Alle Teile der Kurse, die nicht als einzelne Lernobjekte im Lernobjektarchiv abgelegt werden, können durch Kopieren wiederverwendet werden.

7.3.2 Editorkomponenten der Autorenumgebung

In den folgenden Abschnitten werden die Editorkomponenten der Autorenumgebung [HHRS05, HRS05] des ResourceCenters beschrieben. Die Komponenten verwenden das im vorangehenden Abschnitt beschriebene Datenmodell für Kurse und Lektionen und nutzen die in Abschnitt 7.2 beschriebenen Funktionen des Lernobjektarchivs, um Lernobjekte beim Erstellen neuer Lernobjekte wiederzuverwenden oder erstellte Lernobjekte für die Wiederverwendung verfügbar zu machen. Die Autorenumgebung beinhaltet Editorkomponenten für jedes Lernobjekt der im Datenmodell definierten Aggregationsniveaus und für Texte. Medienobjekte sind davon ausgenommen, weil für deren Erstellung und Bearbeitung externe Werkzeuge verwendet werden. Im Folgenden werden die einzelnen Editorkomponenten der Autorenumgebung vorgestellt.

7.3.2.1 Kurseditor

Der Kurseditor ist die Editorkomponente der Autorenumgebung, mit deren Hilfe Lernobjekte mit dem Aggregationsniveau „Kurs“ erstellt werden können. Ein Lern-

objekt dieses Aggregationsniveaus ist das Element auf der höchsten Ebene eines Kurses. Der Kurseditor stellt den Einstiegspunkt zum Bearbeiten von Kursen dar.

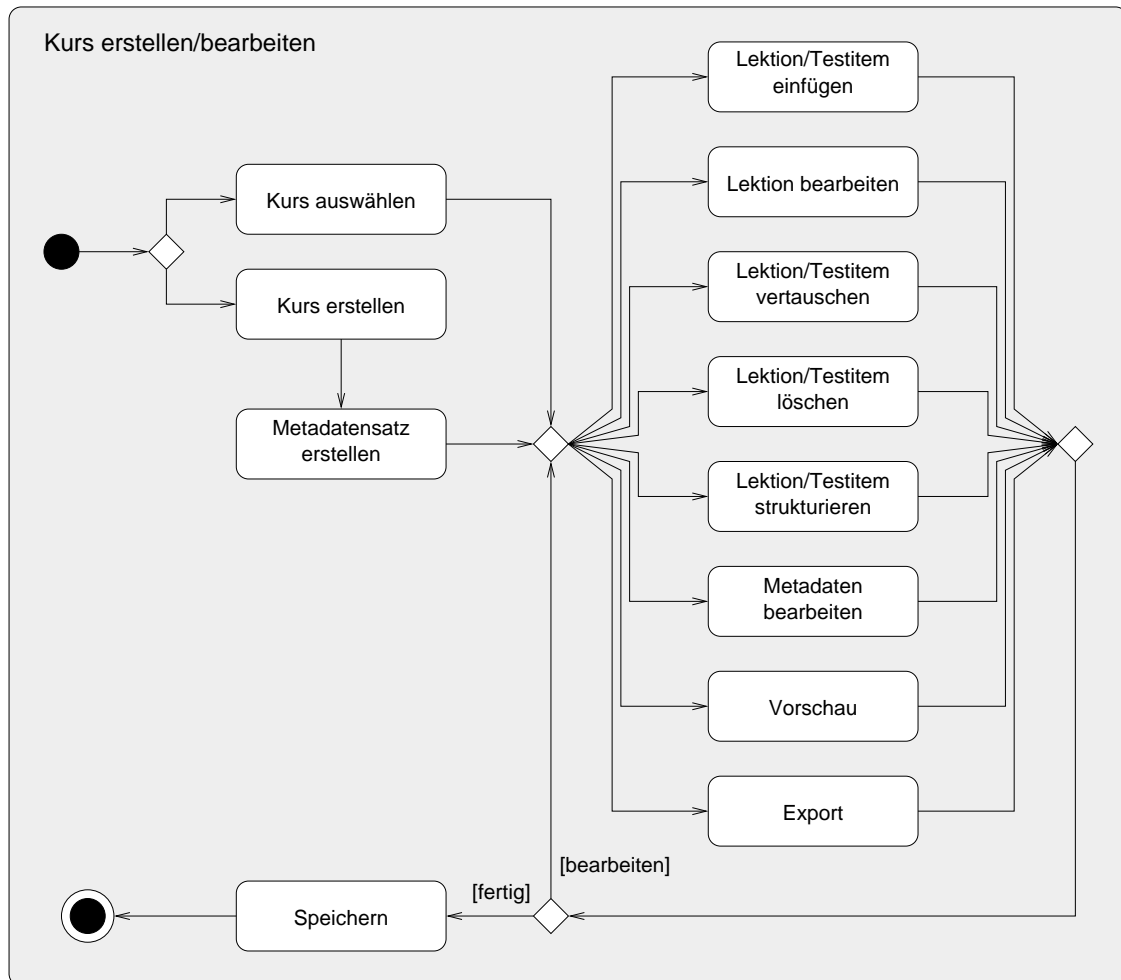


Abbildung 7.23: Autorenprozess des Kurseditors

Abbildung 7.23 zeigt das Aktivitätsdiagramm des Kurseditors. Es leitet sich aus dem Aktivitätsdiagramm des erweiterten, abstrakten Autorenprozesses aus Abbildung 4.9 ab. Das Erstellen von Kursen wird mit Hilfe der Funktion „Neuen Kurs erstellen“ und das Bearbeiten bestehender Kurse mit einem Mausklick auf den entsprechenden Kurs in der Liste der eigenen Kurse eingeleitet. Wenn ein neuer Kurs erstellt werden soll, dann wird ein zunächst noch leerer Kurs zusammen mit einem Metadatensatz angelegt, der noch keine Einträge zur Beschreibung des Inhalts des Kurses enthält. Anschließend gelangt der Autor zur Eingabemaske des Kurseditors. Diese bietet dem Autor verschiedene Aktionen an, die ausgeführt werden können, um den Kurs zu bearbeiten. Hierzu gehört das Einfügen von Lektionen oder Testitems. Dabei können entweder neue Lektionen erstellt oder bestehende,

im ResourceCenter gespeicherte, wiederverwendet werden. Die Lektionen des Kurses können bearbeitet werden, indem eine EditorKomponente zum Bearbeiten einer Lektion gestartet wird. Darüber hinaus können Lektionen und Testitems geordnet, strukturiert und gelöscht werden. Aus dem Kurseditor können zusätzlich auch Funktionen zum Bearbeiten der Kursmetadaten, eine Vorschaufunktion des Kurses und eine Funktion zum Exportieren der Kurse angesprochen werden. Dabei kann die Vorschaufunktion genutzt werden, um den Kurs so anzuzeigen, wie er von Lernern in der Lernplattform betrachtet werden kann. Hierfür werden die XML-Beschreibungen von Kursen und vor allem von Seiten in HTML mit Hilfe der XML Stylesheet Language (XSL) transformiert. Mittels der Exportfunktion können die im ResourceCenter erstellten Kurse exportiert werden, um sie beispielsweise in einem Lernsystem verwenden zu können. Für den Export der Kurse im ResourceCenter stehen zwei Ausgabeformate zur Verfügung. Die Kurse können als HTML-Seiten in einem Archiv oder als SCORM Content Package exportiert werden.

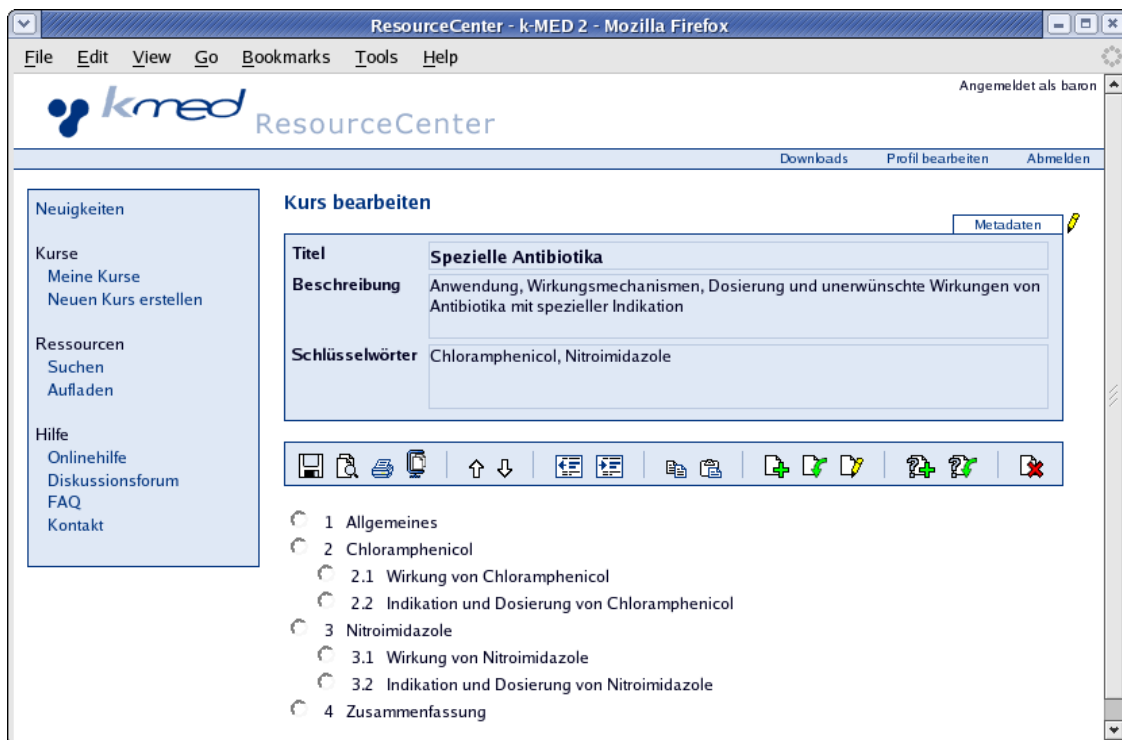


Abbildung 7.24: Kurseditor des ResourceCenters

Abbildung 7.24 zeigt die Oberfläche des Kurseditors des ResourceCenters. In dem oberen Bereich befindet sich das Formular des vereinfachten Metadateneditors, in dem die wichtigsten Metadaten, wie Titel, Beschreibung und Schlüsselwörter, bearbeitet werden können. Dabei wird der Titel, der im Metadatensatz eingetragen ist, als Titel für den Kurs verwendet. Durch Mausklick auf den Stift neben diesem For-

mular gelangt der Autor in den Metadateneditor, in dem alle übrigen Datenfelder bearbeitet werden können. Unter diesem Formular befindet sich die Werkzeugleiste, über die alle Funktionen zum Bearbeiten und Verwalten der Kurse verfügbar gemacht werden. Die Werkzeugleiste beinhaltet Funktionen zum Speichern, Anzeigen, Drucken und Exportieren von Kursen. Mit Hilfe weiterer Funktionen können Lektionen und Testitems geordnet, strukturiert, eingefügt, bearbeitet und gelöscht werden. Unterhalb der Werkzeugleiste befindet sich die Struktur des Kurses. Jeder Eintrag entspricht einer Lektion oder einem Testitem. Über Auswahlelemente kann ein Eintrag zur Bearbeitung selektiert werden.

7.3.2.2 Lektioneneditor

Der Lektioneneditor ist die Editorkomponente zur Bearbeitung von Lernobjekten des Aggregationsniveaus „Lektion“. Das Erstellen und Bearbeiten von Lektionen wird aus dem Kurseditor gestartet. Prinzipiell sollte das Erstellen und Bearbeiten von Lektionen auch außerhalb des Kurseditors angestoßen werden können, um die Erstellung von Lernobjekten nach dem Top-Down-Verfahren zu unterstützen, wie in Abschnitt 4.1.2 beschrieben.

Abbildung 7.25 zeigt das Aktivitätsdiagramm für das Erstellen und Bearbeiten von Lektionen im Lektioneneditor. Das Diagramm ist aus dem erweiterten, abstrakten Aktivitätsdiagramm des auf Aggregation basierenden Autorenprozesses aus Abbildung 4.9 abgeleitet worden. Für das Bearbeiten bestehender Lektionen muss zunächst eine Lektion ausgewählt werden. Das geschieht bereits im Kurseditor. Für den Fall, dass eine neue Lektion erstellt werden soll, wird zunächst eine leere Lektion zusammen mit einem Metadatensatz erstellt. Anschließend gelangt der Autor zu der Oberfläche des Lektioneneditors. Diese bietet den Autoren die Möglichkeit an Blöcke einzufügen, zu bearbeiten, zu ordnen und zu löschen. Über weitere Funktionen können Seitenumbrüche eingefügt oder gelöscht, die Metadaten der Lektion bearbeitet oder die Vorschau zur Anzeige der Lektion gestartet werden.

Abbildung 7.26 zeigt die Oberfläche des Lektioneneditors. Ebenso wie im Kurseditor befindet sich im oberen Bereich des Lektioneneditors ein Formular zur Bearbeitung der wichtigsten Metadaten. Durch einen Mausklick auf den Stift rechts neben dem Formular wird der Metadateneditor zur Bearbeitung des Metadatensatzes der Lektion gestartet. Direkt unter dem Formular zur Bearbeitung der Metadaten befindet sich die Werkzeugleiste des Lektioneneditors. Sie stellt Funktionen zum Speichern und Anzeigen der Lektion und zum Verlassen des Lektioneneditors bereit. Unterhalb der Werkzeugleiste werden die Blöcke der Lektion angezeigt. Jeder Block wird in einem separaten, farbig hinterlegten Bereich angezeigt. Ober- und unterhalb der Blöcke befinden sich Funktionen zum Einfügen und Ordnen von Blöcken sowie zum Einfügen von Seitenumbrüchen. Mit Hilfe der Symbole rechts neben den Blöcken können diese entweder gelöscht oder in die Zwischenablage kopiert werden.

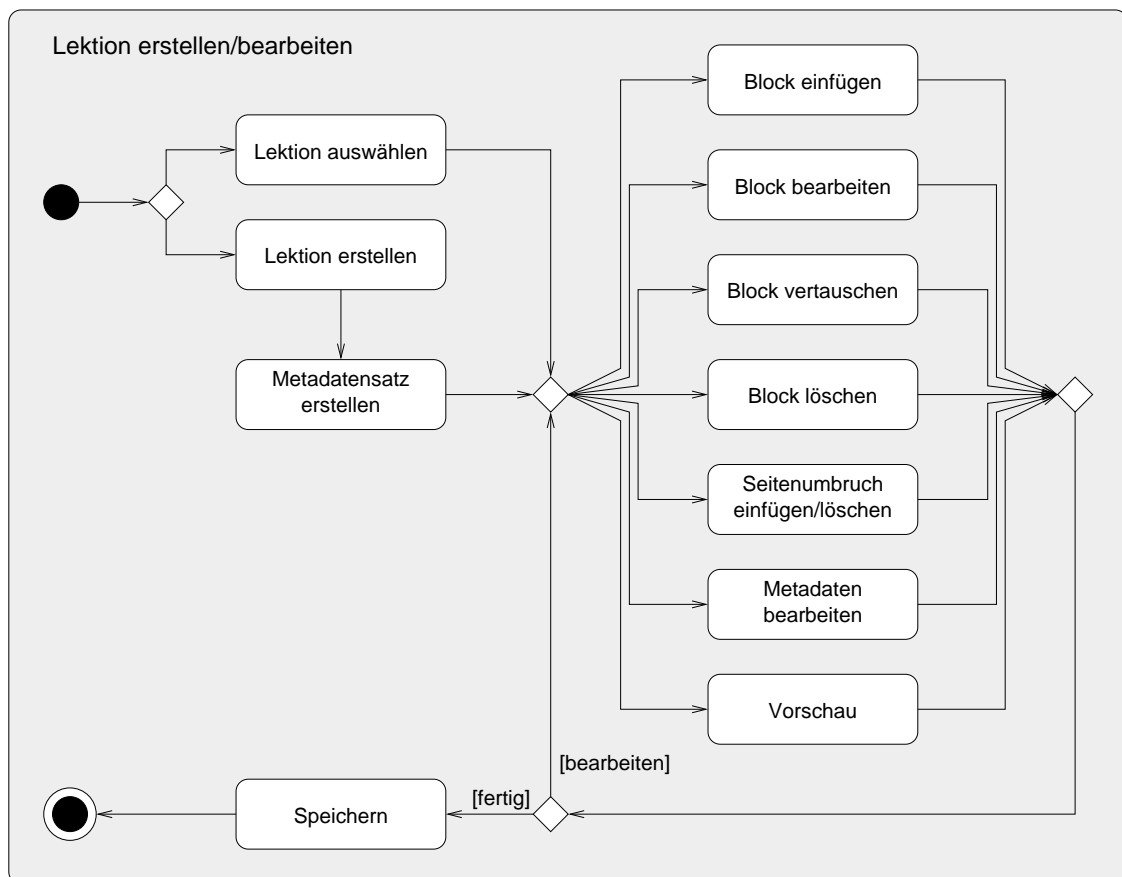


Abbildung 7.25: Autorenprozess des Lektioneneditors

ResourceCenter - k-MED 2 - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

kmed ResourceCenter

Angemeldet als baron

Downloads Profil bearbeiten Abmelden

Neuigkeiten

Kurse
Meine Kurse
Neuen Kurs erstellen

Ressourcen
Suchen
Aufladen

Hilfe
Onlinehilfe
Diskussionsforum
FAQ
Kontakt

Lektion bearbeiten

Titel Indikation und Dosierung von Chloramphenicol

Beschreibung Indikation und Dosierung von Chloramphenicol

Schlüsselwörter Chloramphenicol

Medienobjekt(e) zum Text

Applikation von ...

Textart: normal

Indikationen:

- Schwere bzw. lebensgefährliche Salmonellosen (Typhus, Paratyphus, Salmonellen-Meningitis)
- lebensbedrohliche Augeninfektionen, falls die risikoärmeren Antibiotika (v.a. Chinolone) unwirksam oder kontraindiziert sind
- u.U. bei Hirnabszeß, Rickettsiose (Fleckfieber), Brucellose.

Dosierung:

Bei Erwachsenen: 1,5 - 4 g/d, in 3 bis 4 Gaben (stets Blutbildkontrolle!); bei Kindern und Säuglingen: 50-80 mg/kg/d (Blutbild- und möglichst Blutspiegelkontrolle!); bei Neugeborenen nur bis höchstens 50 mg/kg/d. (Vorsicht vor Akkumulation des Pharmakons in der unreifen Leber und einem daraus

Abbildung 7.26: Lektioneneditor des ResourceCenters

7.3.2.3 Blockeditor

Der Blockeditor ist die Editorkomponente der Autorenumgebung zum Bearbeiten der aggregierten Lernobjekte mit dem kleinsten Aggregationsniveau, den „Blöcken“. Da Blöcke im ResourceCenter keine eigenständigen Lernobjekte und Teile von Lektionen sind, ist der Blockeditor im Lektioneneditor integriert. In der Oberfläche des Lektioneneditors ist für jeden Block ein Blockeditor eingebettet.

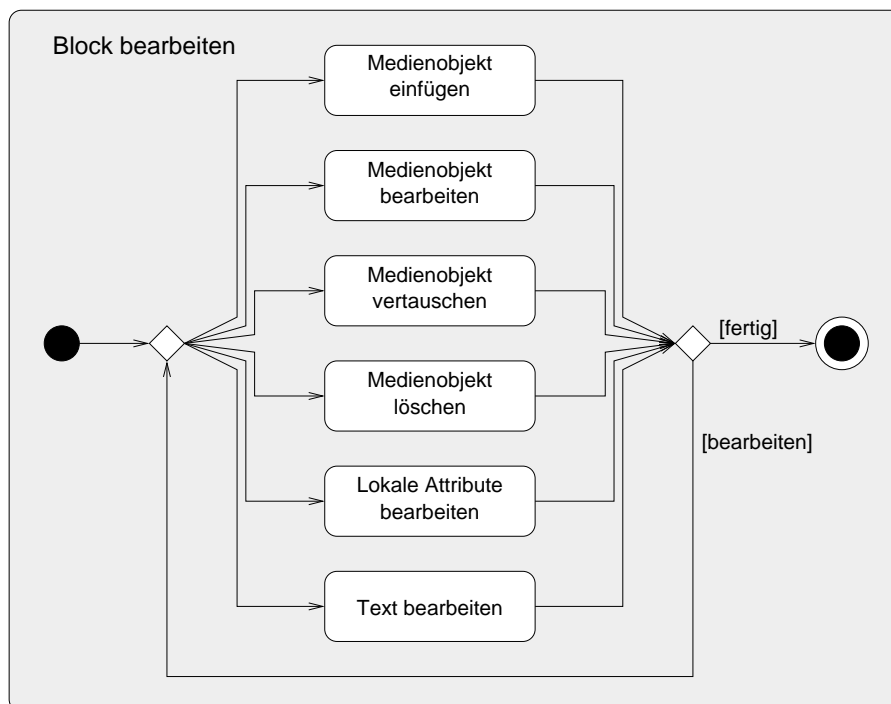


Abbildung 7.27: Autorenprozess des Blockeditors

Abbildung 7.27 zeigt das Aktivitätsdiagramm des Blockeditors. Mit Hilfe des Blockeditors können die Medienobjekte und der Text eines Blocks bearbeitet und allgemein verwaltet werden. Der Blockeditor stellt Funktionen zum Einfügen, Bearbeiten, Ordnen und Löschen von Medienobjekten bereit. Außerdem können mit Hilfe des Blockeditors lokale Attribute, wie Untertitel der Medienobjekte bearbeitet werden. Für die Erstellung und Bearbeitung von Text enthält der Blockeditor einen vom Funktionsumfang deutlich reduzierten HTML-Editor, so dass Autoren ohne spezielle Kenntnisse Textfragmente erstellen können, ohne dass dabei nicht mehr maschinell interpretierbarer ausgezeichnete Text entsteht. Das Bearbeiten von Medienobjekten wird nur für Medienobjekte unterstützt, für die im ResourceCenter Editoren integriert sind. In der derzeitigen Implementierung wird im ResourceCenter nur das Bearbeiten von Tabellen als spezielles Medienobjekt unterstützt. Auch hierfür wird der schon im Blockeditor verwendete HTML-Editor verwendet. Der

Blockeditor stellt im Vergleich zum Kurs- und Lektioneditor keine Funktionen für das Speichern der Blöcke oder Beschreiben der Blöcke mit Metadaten zur Verfügung. Der Grund dafür ist, dass Blöcke zusammen mit der kompletten Lektion über den Lektioneditor gespeichert werden. Eine Metadatenbeschreibung jedes einzelnen Blocks ist nicht vorgesehen, weil Blöcke im ResourceCenter nicht direkt indiziert werden können.

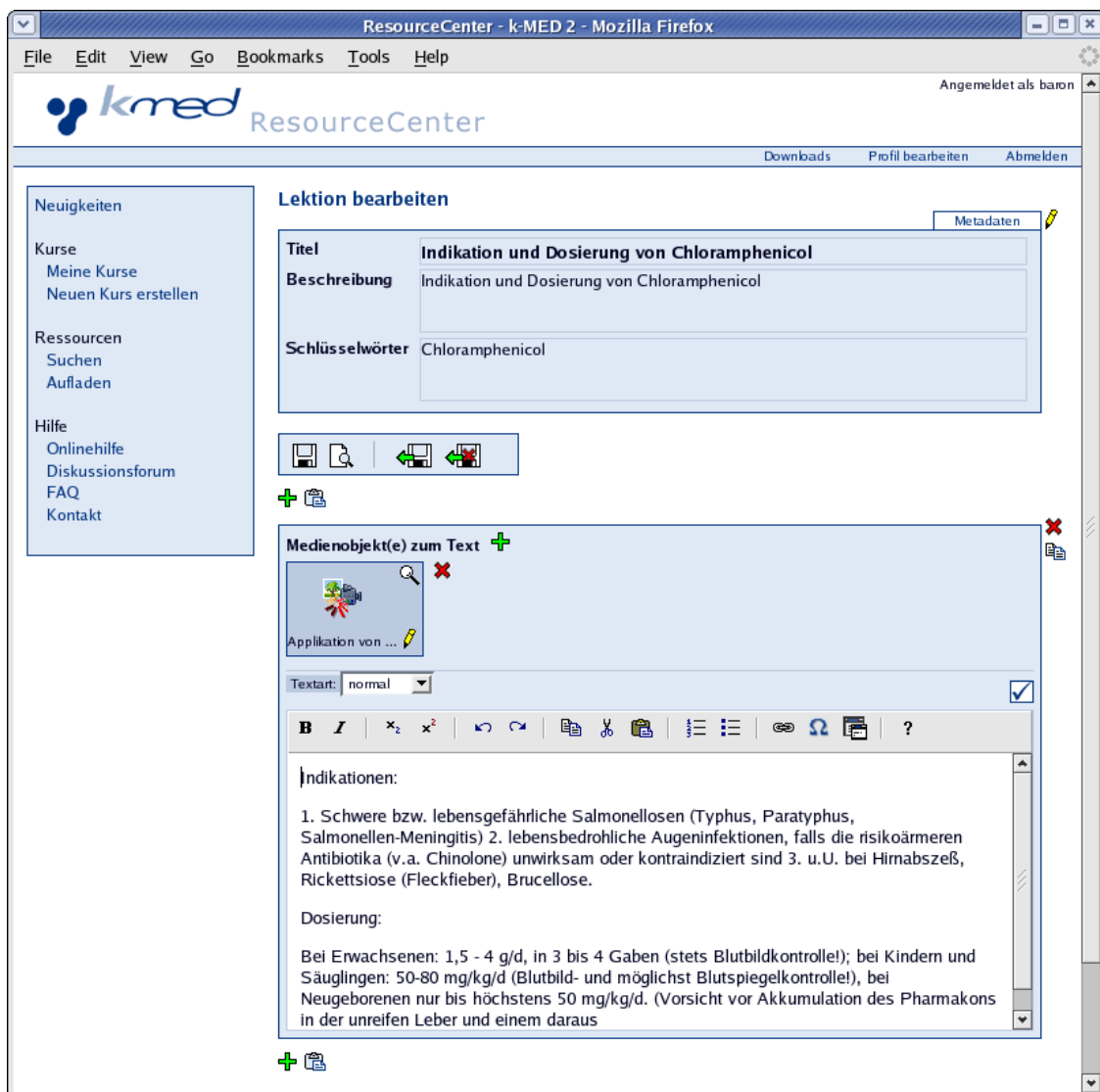


Abbildung 7.28: In den Lektioneditor eingebetteter Blockeditor

Abbildung 7.28 zeigt die Oberfläche des Lektioneditors mit einem integrierten Blockeditor. Im oberen Bereich des Blockeditors können die Medienobjekte verwaltet werden. Hier kann eine beliebige Anzahl von Medienobjekten eingefügt und

bezüglich ihrer Reihenfolge geordnet werden. In Abbildung 7.28 befindet sich ein Bild im Flash-Format in dem Block. Mit dem Lupensymbol kann das Bild betrachtet werden. Das Stiftsymbol kann dazu verwendet werden, den Untertitel des Bildes zu bearbeiten, der ursprünglich aus den Metadaten des Bildes kopiert wurde. Im unteren Bereich des Blockeditors befindet sich der Bereich zum Erstellen und Bearbeiten des Textfragments des Blocks. Abbildung 7.28 zeigt das Textfragment des Blocks im aktivierten HTML-Editor. Mit Hilfe des HTML-Editors kann der Text in gewissem Ausmaß ausgezeichnet werden. Mit Hilfe weiterer Funktionen des HTML-Editors ist es möglich Popup-Fenster einzufügen oder Verweise in den Text einzubetten. Der HTML-Editor wird nur zum Bearbeiten von Textfragmenten aktiviert und danach wieder geschlossen. Mit Hilfe des Auswahlelementes oberhalb des HTML-Editors kann die didaktische Bedeutung des Textfragments zum Beispiel als normaler Lehrtext oder Merksatz eingestellt werden. Je nach ausgewähltem Typ werden die Textfragmente in verschiedenen Formaten in den exportierten WBTs eingebettet, um so eine visuelle Kodierung der Absatztypen zu erzielen.

7.3.3 Einfügen von Lernobjekten in der Autorenumgebung

Das Einfügen von neuen oder existierenden Lernobjektfragmenten wird in allen zuvor vorgestellten Editorkomponenten ermöglicht. Lernobjekte können aus drei verschiedenen Quellen eingefügt werden: leere Fragmente über Funktionen der Editoren der Autorenumgebung, bestehende Lektionen, Testitems und Medienobjekte aus dem Lernobjektarchiv, Testitems und Medienobjekte alternativ auch vom lokalen Laufwerk des Arbeitsplatzrechners eines Autors. Da diese Quellen jeweils Lernobjekte mit verschiedenen Aggregationsniveaus liefern, wird das Einfügen von Lernobjektfragmenten zunächst abstrakt beschrieben und anschließend die Anwendung in der Autorenumgebung des ResourceCenters vorgestellt.

7.3.3.1 Lernobjekt aus der Autorenumgebung einfügen

Bei dem Einfügen von Lernobjekten über Funktionen der Editoren der Autorenumgebung werden neue leere Lernobjektfragmente erstellt. Dabei dient im Prinzip die Autorenumgebung als Quelle für die Aggregation neuer Lernobjektfragmente.

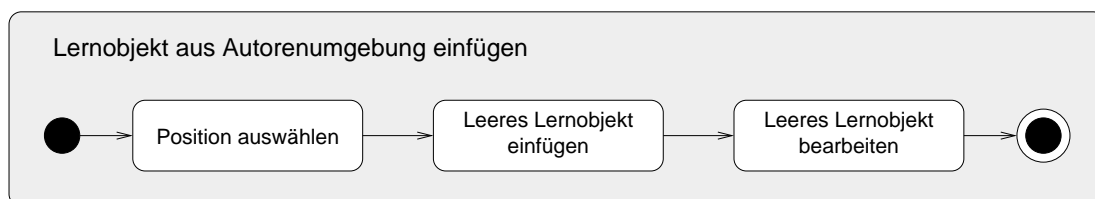


Abbildung 7.29: Lernobjekt aus der Autorenumgebung einfügen

Abbildung 7.29 zeigt das Aktivitätsdiagramm für das Einfügen von Lernobjekten aus der Autorenumgebung. Danach sind zum Einfügen eines Lernobjekts aus der Autorenumgebung drei Aktivitäten notwendig. Zunächst muss bei der ersten Aktivität die Position ausgewählt werden, an die das neue Lernobjekt eingefügt werden soll. Anschließend wird an der gewählten Position durch die Autorenumgebung ein leeres Lernobjekt erstellt. Diese wird dann in der letzten Aktivität bearbeitet. Dabei bleibt die Editorkomponente aus der der neue Autorenprozess gestartet wurde bei der Aktivität „Lernobjektfragment einfügen“ solange aktiv, bis die Bearbeitung des neuen Lernobjektfragments beendet wurde. Im Grunde handelt es sich dabei um eine Rekursion im Autorenprozess, die aufgrund der möglichen Rekursion im Dokumentmodell erforderlich wird.

Die Autorenumgebung des ResourceCenters kann nur dann als Quelle neuer Lernobjektfragmente dienen, wenn sie eine Editorkomponente für das Aggregationsniveau und den Typ des zu erstellenden Lernobjekts zur Verfügung stellt. Für den gegenwärtigen Stand der Implementierung kann der Lektioneditor im Kurseditor, der Blockeditor im Lektioneditor und der Tabelleneditor im Blockeditor als Quelle für neue Lernobjekte genutzt werden.

7.3.3.2 Lernobjekt aus dem Lernobjektarchiv einfügen

Bei dem Einfügen von Lernobjekten aus dem Lernobjektarchiv handelt es sich in der Regel um das Wiederverwenden von Lernobjekten, die bereits im Lernobjektarchiv abgelegt sind. Der Vorgang beim Einfügen von Lernobjekten aus dem Lernobjektarchiv entspricht im Wesentlichen dem Abrufen verfügbarer Lernobjekte, wie bereits im Abschnitt 7.2.2.2 beschrieben.

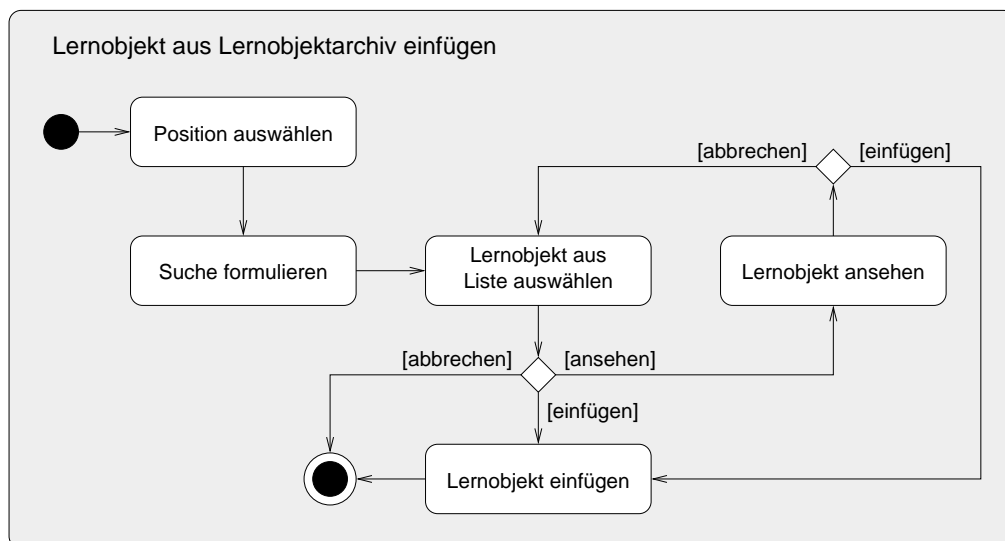


Abbildung 7.30: Lernobjekt aus dem Lernobjektarchiv einfügen

Abbildung 7.30 zeigt das Aktivitätsdiagramm zum Einfügen von Lernobjekten aus dem Lernobjektarchiv. Analog zu dem Einfügen von Lernobjekten aus der Autorenumgebung muss zunächst die Position gewählt werden, an die das Lernobjekt eingefügt werden soll. Danach gelangen die Autoren zum Suchformular, um die Suche für das gesuchte Lernobjekt zu formulieren und zu starten. Anschließend werden die Lernobjekte, die die formulierten Kriterien erfüllen in der Ergebnisliste aufgelistet. Auf Basis eines Auszugs der Metadaten kann jetzt ein Lernobjekt durch Anklicken ausgewählt werden, um es an entsprechender Position in den Kurs einzufügen. Alternativ kann das gewählte Lernobjekt jedoch auch zunächst betrachtet werden. Ist das Lernobjekt geeignet, kann es eingefügt werden, falls nicht, gelangen die Autoren wieder zur Ergebnisliste. Von dort kann das Einfügen eines Lernobjekt jedoch auch abgebrochen werden, wenn sich kein geeignetes Lernobjekt finden lässt.

Alle Lernobjekte, die im Lernobjektarchiv direkt indizierbar sind, außer Kurse selbst, können im ResourceCenter aus dem Lernobjektarchiv als Fragment in ein neues Lernobjekt eingefügt werden. Dabei handelt es sich um Lektionen, Testitems und alle Medienobjekte mit Ausnahme von Text. Damit wird das Einfügen von Lektionen und Testitems im Kurseditor und das Einfügen von Medienobjekten im Blockeditor möglich.

7.3.3.3 Lernobjekt von einem lokalen Laufwerk einfügen

Bei dem Einfügen von Lernobjekten von lokalen Laufwerken werden Lernobjekte von außerhalb als Fragment in das entsprechende Lernobjekt eingefügt. Da die Lernobjekte dabei in dem Lernobjektarchiv abgelegt werden, entspricht dieser Vorgang dem Ablegen von Lernobjekten, wie in Abschnitt 7.2.2.1 beschrieben.

Abbildung 7.31 zeigt das Aktivitätsdiagramm zum Einfügen eines Lernobjekts von einem lokalen Laufwerk. Auch bei diesem Vorgang wird zunächst im ersten Schritt die Position gewählt, an der das entsprechende Lernobjekt eingefügt werden soll. Anschließend gelangen die Autoren zum Aufladeformular. Hier selektieren sie die entsprechende Datei des Lernobjekts und laden diese in das ResourceCenter auf. Im nächsten Schritt werden präsentationsspezifische Metadaten manuell erfasst. Können alle notwendigen Parameter automatisch bestimmt werden, wird dieser Schritt übersprungen. Anschließend werden in dem vereinfachten Eingabefeld des Metadaten-Wizards die nicht automatisch zu bestimmenden Metadaten manuell erfasst und ein Metadatenatz für das aufgeladene Lernobjekt erstellt. Dieser kann anschließend im Metadateneditor bei Bedarf bearbeitet werden. Im Anschluss wird das aufgeladene Lernobjekt an der gewählten Position eingefügt.

Das Einfügen von Lernobjekten von lokalen Laufwerken der Autoren dient in erster Linie dazu, Lernobjekte, die mit Hilfe von externen Werkzeugen erstellt wurden, als Fragment eines Lernobjekts einzufügen. Im ResourceCenter werden externe Werkzeuge zum Erstellen von Testitems und Medienobjekten außer Text verwendet. Somit kann das Einfügen von Lernobjekten von einem lokalem Laufwerk für

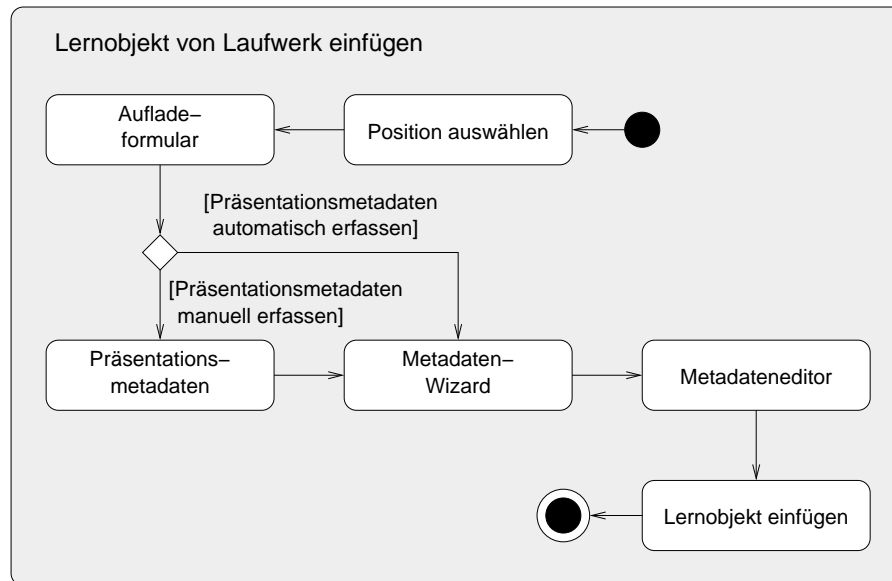


Abbildung 7.31: Lernobjekt von einem lokalen Laufwerk einfügen

das Einfügen von Testitern im Kurseditor oder das Einfügen von Medienobjekten im Blockeditor angewendet werden.

7.3.4 Zusammenspiel der Editorkomponenten der Autorenumgebung

In diesem Abschnitt erfolgt die Betrachtung des Zusammenspiels der zuvor beschriebenen Editorkomponenten der Autorenumgebung des ResourceCenters unter Einbeziehung des Browsers als Klient auf den lokalen Arbeitsplatzrechnern der Autoren. Als Anwendung wird hierfür die Erstellung beziehungsweise Aggregation eines neuen Kurses in der Autorenumgebung betrachtet. Dabei werden sowohl bestehende als auch neu erstellte Lernobjekte in dem Kurs aggregiert. Für die Darstellung des Zusammenwirkens der verschiedenen Systemkomponenten werden Sequenzdiagramme verwendet. Diese beschreiben die Kommunikation während der Kurserstellung zwischen den Systemkomponenten „Browser“, „Autorenumgebung“, „Lernobjektarchiv“ sowie den zur Laufzeit instanziierten Editorkomponenten, wie Kurseditor und Lektioneditor. Zur Reduzierung der Komplexität der Diagramme werden bei der Betrachtung verschiedene Einschränkungen gemacht. Hierzu wird die Erstellung von Metadaten nach dem in Kapitel 5 beschriebenen Verfahren lediglich angedeutet. Außerdem sind die Nachrichten, die zur Kommunikation zwischen Browser und allen anderen Objekten ausgetauscht werden, stark abstrahiert. Alle Nachrichten, die vom Browser verschickt werden, gehen von Aktionen der Autoren aus. Zur Verbesserung der Orientierung enthalten die Diagramme die

Zeitmarken mit den Buchstaben „a“ bis „f“. In dem Beispiel werden Lernobjekte entsprechend des Klassifikationsschemas aus Abschnitt 4.2.4 und dem Datenmodell des ResourceCenters, das in Abschnitt 7.3.1 beschrieben wurde, klassifiziert und als eigenständige Lernobjekte beziehungsweise Lernobjektfragmente im Lernobjektarchiv gespeichert.

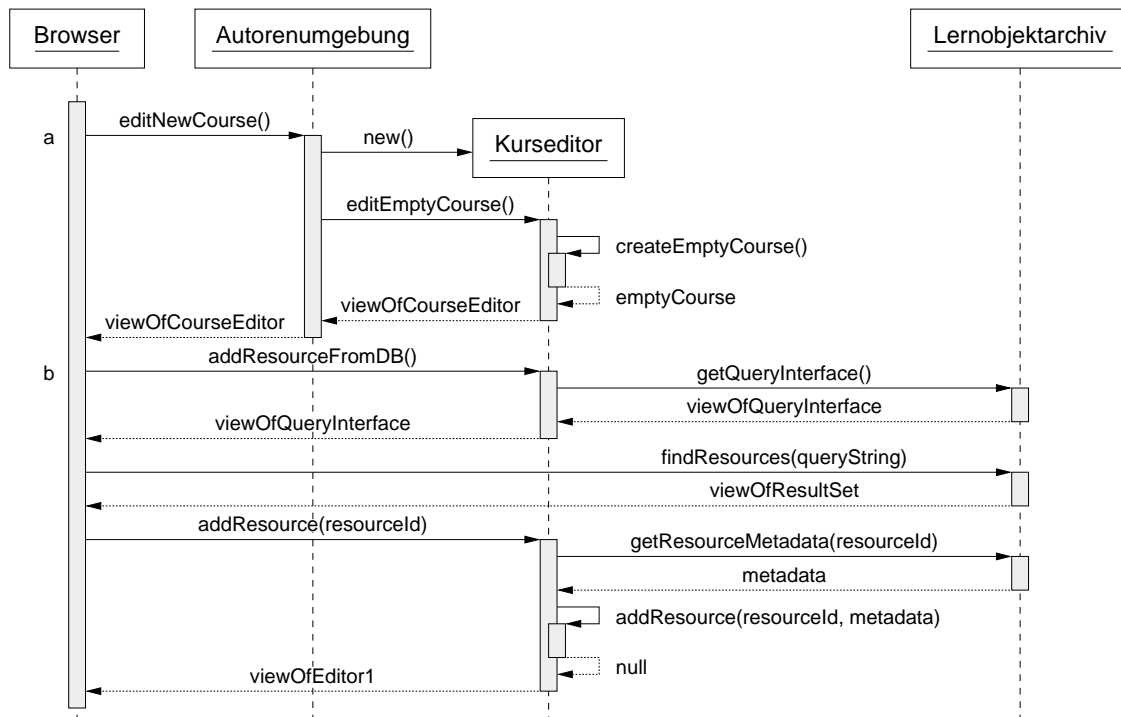


Abbildung 7.32: Erstellen eines neuen Kurses

Das Sequenzdiagramm in Abbildung 7.32 zeigt das Erstellen eines Kurses, in den nach der Erstellung eine bestehende Lektion aus dem Lernobjektarchiv eingefügt wird. Als Ausgangspunkt existieren die Komponenten Browser, der Klient auf dem Arbeitsplatzrechner des Autors, sowie Autorenumgebung und Lernobjektarchiv des ResourceCenters.

Ausgehend von einer Aktion eines Autors soll zum Zeitpunkt „a“ ein leerer Kurs für die Aggregation von Lektionen erstellt werden. Hierfür schickt der Browser die Nachricht „createNewCourse()“ an die Autorenumgebung. Diese instanziiert daraufhin die Editorkomponente „Kurseditor“. Anschließend wird es mit der Nachricht „editEmptyCourse()“ für die Bearbeitung eines leeren Kurses initialisiert. Der leere Kurs wird dabei in Selbstdelegation erstellt. Als Antwort liefert der Kurseditor „viewOfCourseEditor“ zurück, die die Ansicht des Kurseditors zur Bearbeitung des Kurses darstellt. Die Autorenumgebung reicht diese Antwort an den Browser weiter.

Zum Zeitpunkt „b“ soll eine bestehende Lektion in den Kurs eingefügt werden. Hierfür sendet der Browser die Nachricht „addResourceFromDB()“ an den Kurseditor. Dieser holt sich daraufhin die Ansicht für die Suchmaske vom Lernobjektarchiv. Hierfür schickt er die Nachricht „getQueryInterface()“ an das Lernobjektarchiv und erhält dafür die Antwort „viewOfQueryInterface“. Diese reicht der Kurseditor an den Browser weiter. Dieser zeigt danach die Ansicht des Suchformulars an, mit dessen Hilfe Autoren mittels geeigneter Anfragen das Lernobjektarchiv nach bestehenden Lernobjekten durchsuchen können. Zur Suche nach einer geeigneten Lektion schickt der Browser die Nachricht „findResources(queryString)“ mit dem entsprechenden „queryString“ an das Lernobjektarchiv. Die Antwort des Lernobjektarchivs enthält die Ansicht der Lernobjekte, die auf die gestellte Suchanfrage passen. Aus der Liste der gefundenen Lernobjekte wählt der Autor jetzt eine Lektion aus, die in den Kurs eingefügt werden soll. Direkt nach der Auswahl der Lektion, schickt der Browser die Nachricht „addResource(resourceId)“ an den Kurseditor, um die entsprechende Lektion in den Kurs einzufügen. Mit der Nachricht „getResourceMetadata(resourceId)“ fragt der Kurseditor die Metadaten der Lektion ab, die in der Antwort „metadata“ vom Lernobjektarchiv enthalten sind. Anschließend delegiert der Kurseditor das Hinzufügen der Lektion durch die Nachricht „addResource(resourceId, metadata)“ an sich selbst. Hierbei werden die Metadaten benötigt, um den Titel der eingefügten Lektion anzeigen zu können. Nachdem die Lektion dem Kurs hinzugefügt worden ist, antwortet der Kurseditor dem Browser mit der aktualisierten Ansicht der Eingabemaske des Kurseditors.

Nachdem als Teil des neu erstellten Kurses eine bestehende Lektion aus dem Lernobjektarchiv hinzugefügt wurde, soll dem Kurs nun eine weitere, neu erstellte Lektion hinzugefügt werden. Abbildung 7.33 zeigt das Sequenzdiagramm zum Einfügen und Erstellen der neuen Lektion.

Zum Zeitpunkt „c“ soll dem Kurs eine neue Lektion hinzugefügt werden. Zu diesem Zweck wird von dem Browser die Nachricht „addLecture“ zum Kurseditor geschickt. Dieser instanziiert daraufhin die Komponente „LectureEditor“. Der LectureEditor wird mit der Nachricht „EditEmptyLecture()“ für die Bearbeitung einer neuen, leeren Lektion initialisiert. Die Erstellung einer leeren Lektion delegiert der LectureEditor durch die Nachricht „createEmptyLecture()“ an sich selbst. Das Ergebnis ist eine Lektion mit einem leeren Block. Nach der Erstellung der leeren Lektion gibt der LectureEditor die Antwort „viewOfLectureEditor“, die die Eingabemaske des LectureEditors zur Bearbeitung der leeren Lektion darstellt, an den Kurseditor zurück. Diese reicht der Kurseditor an den Browser weiter. Ab diesem Zeitpunkt wird die Bearbeitung des Kurses durch den Kurseditor unterbrochen. Statt des Kurseditors ist jetzt der LectureEditor sichtbar. Der Zustand des Kurseditors bleibt während der Bearbeitung der Lektion erhalten.

Jetzt soll der leere Block der Lektion bearbeitet und somit mit Inhalt gefüllt werden. Da Blöcke keine eigenständigen Lernobjekte sind, sondern direkt in den Lektionen im Datenmodell des ResourceCenters eingebettet werden, wird die Be-

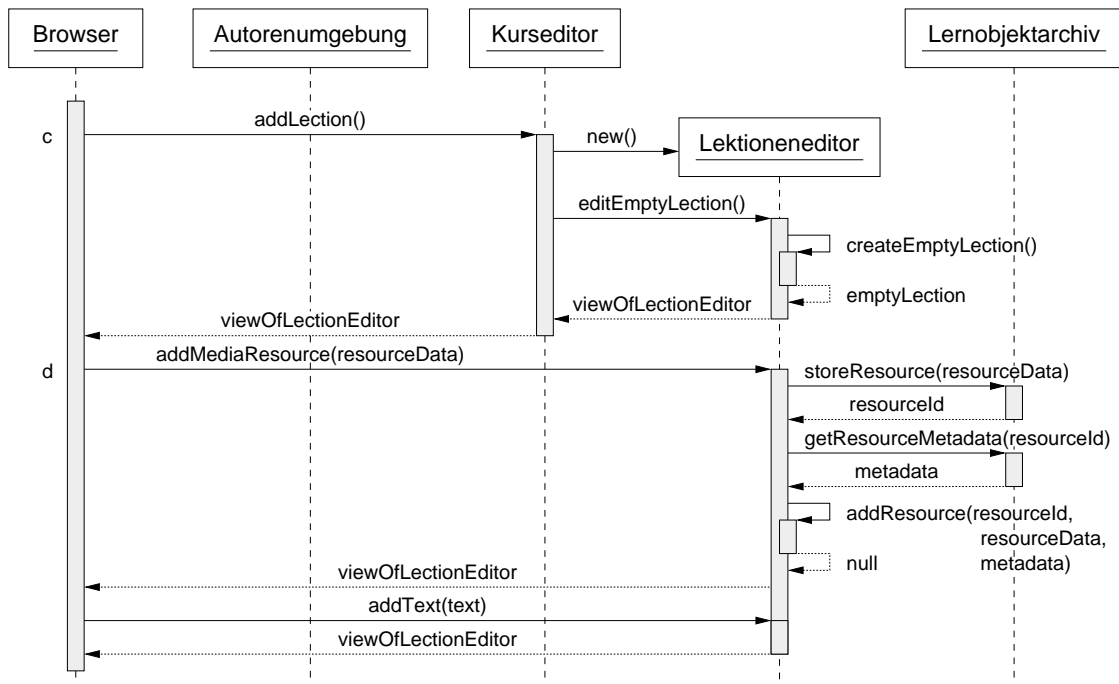


Abbildung 7.33: Erstellen einer neuen Lektion

arbeitung der Blöcke durch die Lektioneneditoren übernommen, so dass die zur Bearbeitung notwendigen Nachrichten auch an den Lektioneneditor gerichtet werden. Zum Zeitpunkt „d“ soll ein Bild in den leeren Block der Lektion eingefügt werden. Hierfür lädt der Autor das Bild, das er zuvor mit einem entsprechenden Bildbearbeitungsprogramm erstellt hat, in die Autorenumgebung. Der Browser schickt zu diesem Zweck die Nachricht „addMediaResource(resourceData)“ an den Lektioneneditor. Der Lektioneneditor legt das aufgeladene Bild gleich mit der Nachricht „storeResource(resourceData)“ im Lernobjektarchiv ab. An dieser Stelle erfolgt auch die Generierung der Metadaten mit Hilfe des Metadaten-Wizards. Für die übersichtlichere Gestaltung ist die Metadatengenerierung jedoch nicht im Diagramm enthalten. Die Antwort des Lernobjektarchivs enthält den Identifikator, unter dem das Bild künftig im Lernobjektarchiv zu finden ist. Danach fordert der Lektioneneditor die beim Aufladen des Bildes generierten Metadaten mit der Nachricht „getResourceMetadata(resourceId)“ beim Lernobjektarchiv an. Mit der Nachricht „addResource(resourceId, resourceData, metadata)“ delegiert der Lektioneneditor das Hinzufügen des Bildes schließlich an sich selbst. Als Antwort schickt der Lektioneneditor eine aktualisierte Ansicht seiner Eingabemaske an den Browser. In einem zweiten Schritt soll das Bild mit einem Text beschrieben werden. Hierfür wird der Block, in den das Bild eingefügt wurde, um diesen Text ergänzt. Der Text wird direkt im Browser erfasst. Wie in Abbildung 7.28 gezeigt, können die Texte direkt in der aktivierten Eingabemaske des Blocks verfasst werden. Danach werden

sie mit der Nachricht „addText(text)“ an den Lektioneneditor übertragen. Dieser speichert den Text direkt im Datenmodell des Blocks und schickt als Antwort die aktualisierte Ansicht des Blocks zum Browser zurück.

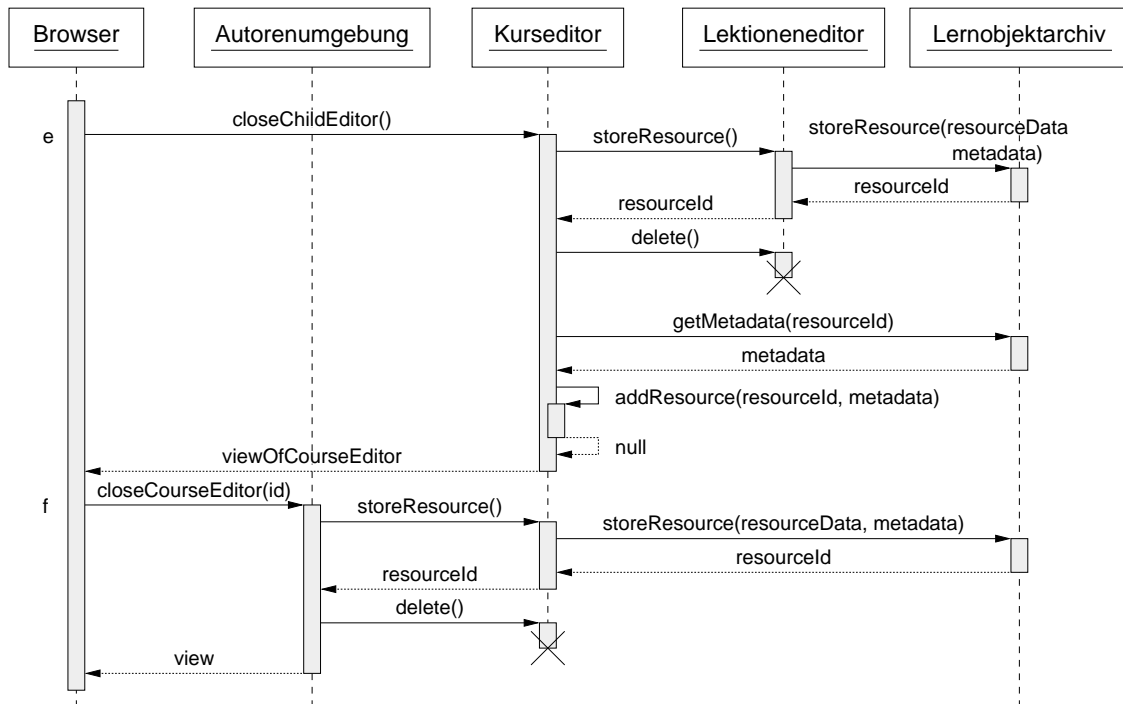


Abbildung 7.34: Speichern der Lektion und des Kurses

Die Bearbeitung der Lektion ist nun abgeschlossen und zum Zeitpunkt „e“ soll der Lektioneneditor geschlossen werden (vergleiche Abbildung 7.34). Hierfür schickt der Browser die Nachricht „closeChildEditor()“ an den Kurseditor. Dieser weist den Lektioneneditor mit der Nachricht „storeResource()“ an, die bearbeitete Lektion im Lernobjektarchiv zu speichern. Dieser schickt deshalb die Nachricht „storeResource(resourceData, metadata)“ zur Speicherung der Lektion an das Lernobjektarchiv. In der Antwort teilt das Lernobjektarchiv den Identifikator „resourceId“ mit, unter dessen Verwendung künftig auf die neu erstellte Lektion zugegriffen werden kann. Diese reicht der Lektioneneditor an den Kurseditor weiter. Mit der Nachricht „delete()“ wird jetzt die Lebensdauer der Editorkomponente des Lektioneneditors beendet. Damit die neu erstellte Lektion dem Kurs hinzugefügt werden kann, werden deren Metdaten benötigt. Diese werden mit der Nachricht „getMetadata(resourceId)“ vom Lernobjektarchiv abgefragt. Das Hinzufügen der Lektion delegiert der Kurseditor mit der Nachricht „addResource(resourceId, metadata)“ an sich selbst. Anschließend schickt der Kurseditor eine aktualisierte Ansicht seiner Eingabemaske an den Browser als Antwort zurück.

Zum Zeitpunkt „f“ soll auch der Kurseditor geschlossen werden. Der Browser

schickt deswegen der Autorenumgebung die Nachricht „closeChildEditor()“. Ebenso wie beim Beenden des Lektioneditors werden jetzt die Daten des Kurseditors im Lernobjektarchiv gespeichert. Das erfolgt mit der Nachricht „storeResource()“, die die Autorenumgebung an den Kurseditor schickt. Dieser speichert den Kurs mit der Nachricht „storeResource(resourceId, metadata)“ im Lernobjektarchiv. Das Lernobjektarchiv antwortet wieder mit dem Identifikator „resourceId“, mit dessen Hilfe auf den neu erstellten Kurs zugegriffen werden kann. In diesem Kontext wird der Identifikator jedoch nicht weiter genutzt. Danach beendet die Autorenumgebung die Lebensdauer des Kurseditors mit der Nachricht „delete()“. Als Antwort schickt die Autorenumgebung die Ansicht der Autorenumgebung an den Browser.

Als Ergebnis der Betrachtung des Zusammenspiels der verschiedenen Editorkomponenten ist zu nennen, dass das Lernobjektarchiv bei der Arbeit mit der Autorenumgebung eine zentrale Rolle spielt. Es dient als Quelle für bereits existierende Lernobjekte, die zusammen mit anderen bereits bestehenden Lernobjekten und neu erstellten Lernobjekten aggregiert werden können. Die durch Aggregation neu zusammengestellten Lernobjekte selbst werden wieder im Lernobjektarchiv abgespeichert, so dass diese für die Aggregation von Lernobjekten auf höherer Ebene bereitstehen und wiederverwendet werden können.

7.4 Zusammenfassung

In diesem Kapitel wurde zunächst eine geeignete Systemarchitektur für die Implementierung entwickelt, die es einer Gruppe von Autoren ermöglicht, Lernobjekte beim Erstellen neuer Lernobjekte in einer auf Aggregation basierenden Autorenumgebung wiederzuverwenden. Bei der gewählten Systemarchitektur sind durch das zentral gelegene Lernobjektarchiv mit angebundener Autorenumgebung allen Autoren sämtliche verfügbaren Lernobjekte zugänglich. Durch die Entwicklung der zentralen Autorenumgebung als Web-Applikation wird auf den lokalen Arbeitsplätzen lediglich die Installation eines Web-Browsers sowie die Installation aller benötigten Werkzeuge zur lokalen Erstellung von Medienobjekten benötigt.

Das ResourceCenter stellt die Implementierung der in dieser Arbeit entwickelten Konzepte für die Wiederverwendung von Lernobjekten in einer auf Aggregation basierenden Autorenumgebung dar. Die zwei wichtigsten Systemkomponenten des ResourceCenters, das Lernobjektarchiv und die Autorenumgebung, sind vorgestellt worden. Das Lernobjektarchiv beinhaltet alle Funktionen zum Ablegen und Abrufen und allgemein dem Verwalten von Lernobjekten. Als erweiterte Funktion des Lernobjektarchivs ist der Metadaten-Wizard zu sehen, mit dessen Hilfe die Erstellung von Metadaten für die abzulegenden Lernobjekte deutlich vereinfacht wird. Von diesen Funktionen des Lernobjektarchivs macht die auf Aggregation basierende Autorenumgebung des ResourceCenters immer dann Gebrauch, wenn bereits bestehende Lernobjekte beim Erstellen neuer Lernobjekte wiederverwendet werden

sollen. Hierfür bieten die drei ineinander geschachtelten Editorkomponenten der Autorenumgebung zum Erstellen von Kursen, Lektionen und Blöcken die Möglichkeiten Lernobjekte einzufügen, die über das Lernobjektarchiv verfügbar sind. Aber auch das Einfügen von Lernobjekten von lokalen Laufwerken der Arbeitsstationen der Autoren wird von der Autorenumgebung unterstützt, so dass vor allem für die Erstellung von Medienobjekten externe Werkzeuge eingesetzt werden können.

Das ResourceCenters wird in verschiedenen Bereichen produktiv eingesetzt. Insbesondere dient das ResourceCenter als Lernobjektarchiv und Autorenumgebung in einem großen Verbundprojekt [Szi05]. Auf Basis der Nutzung in diesem Projekt wurde eine Evaluation des ResourceCenters durchgeführt, die im nächsten Kapitel beschrieben wird.

8 Evaluation

Dieses Kapitel beschreibt die Evaluation verschiedener Konzepte, die in der Implementierung des ResourceCenters, das in Kapitel 7 beschrieben wurde, umgesetzt wurden. Dabei basiert die Evaluation auf Erfahrungen und Daten, die während des Betriebs des ResourceCenters im Projekt k-MED [15] gemacht, beziehungsweise gesammelt wurden. Zunächst wird der Kontext, in dem die Evaluation des ResourceCenters erfolgte, näher betrachtet. Danach wird die Evaluation des Metadaten-Wizards beschrieben, mittels der überprüft werden soll, ob die durch die Einführung des Metadaten-Wizards erhofften Verbesserungen bei der Erstellung von Metadaten eingetroffen sind. Anschließend erfolgt eine quantitative Betrachtung der mit Hilfe des ResourceCenters erstellten und wiederverwendeten Lernobjekte, welche eine Beurteilung des in dieser Arbeit entwickelten Konzepts eines auf Aggregation basierenden Autorenprozesses erlaubt.

8.1 Kontext der Evaluation

Das in Kapitel 7 beschriebene ResourceCenter wird zur Erstellung von WBTs verwendet. Dies geschieht im Rahmen des Projektes k-MED (Knowledge Based Multimedia Medical Education) [15]. Die erstellten WBTs werden von den Projektpartnern ergänzend zu ihrer Vorlesung im Medizinstudium angeboten. Langfristig ist der Aufbau eines internetbasierten Lehr- und Lernangebots für die medizinische Aus- und Weiterbildung an hessischen Universitäten das Ziel des Projektes k-MED.

In k-MED liegt die inhaltliche Konzentration auf Fächern der Vorklinik und auf theoretisch orientierten Fächern des klinischen Studienabschnittes. Es werden WBTs für die Fachgebiete Anästhesie und Intensivtherapie, Arbeits- und Sozialmedizin, Biochemie, Humangenetik, Infektiologie, Krankenhaushygiene, Nuklearmedizin, Pharmakologie, Physiologie, Zytologie und Histologie entwickelt. Damit soll eine möglichst umfangreiche thematische Abdeckung erreicht werden, um die erstellten WBTs in einem Großteil der vorklinischen, klinisch-theoretischen sowie in ausgewählten klinisch-praktischen Fächern des humanmedizinischen Studiums curricular einbinden zu können. Derzeit werden die WBTs von mehr als 1200 Studierenden der beteiligten medizinischen Fakultäten und von über 180 Gästen aus dem Bereich niedergelassener Ärzte und der Pharmaindustrie genutzt.

Genutzt wird das ResourceCenter von 13 aktiven Fachautoren der oben genannten 11 Fachgebiete. Zu diesen Fachautoren gehören in erster Linie Professoren und

Doktoren mit Lehrauftrag. In einigen Fällen gehören zu den Fachautoren Doktoranden, die im Rahmen ihrer Promotion im Auftrag eines Dozenten WBTs erstellen. Nur einer der aktiven Autoren ist explizit für die Erstellung von Lernobjekten angestellt worden und hat im Vergleich zu den anderen Autoren keinen medizinischen oder biochemischen Hintergrund. Auf dem Gebiet der elektronischen Informationsverarbeitung gehen die Kenntnisse der Autoren sehr weit auseinander. Alle Autoren können mit gängigen Textverarbeitungssystemen umgehen. Vereinzelt liegen sogar Kenntnisse über die Erstellung von HTML-Seiten mit aktiven Komponenten wie Java-Script vor. Über grundlegende technologische Ansätze und Vorgehensweisen zur Wiederverwendung von Lernobjekten, wie beispielsweise die Modularisierung von Lernobjekten und deren Beschreibung mit einem genormten Metadatenchema, liegen jedoch keine Erfahrungen vor. Im Rahmen des Projektes wurden die Autoren in der Erstellung von HTML-Seiten und ihren darin enthaltenen Medienobjekte geschult.

Die Funktionen des ResourceCenters sind in verschiedenen Zeitabschnitten fertiggestellt und für die Benutzung freigegeben worden. Die schrittweise Freigabe der Funktionen des ResourceCenters erlaubte die Evaluation der Teilkomponente Metadaten-Wizard unabhängig vom Lernobjektarchiv und von der Autorenumgebung, in die der Metadaten-Wizard jetzt eingebunden ist. Das ermöglichte die nachfolgend dargestellt Evaluation des Metadaten-Wizards.

8.2 Erstellung von Metadaten

Dieser Abschnitt fasst die Evaluation des Metadaten-Wizards zusammen. Zunächst wird hierfür die historische Entwicklung der Metadatenerstellung im Projekt k-MED beschrieben. Danach erfolgt die eigentliche Evaluation des Metadaten-Wizards.

Zu Beginn des Projekts k-MED wurden die erstellten Lernobjekte mit einem web-basierten Metadateneditor erstellt und lokal gespeichert. Die Erstellung der Metadaten war nicht in den Erstellungsprozess der Lernobjekte selbst integriert. Dieses Werkzeug war ein auf Formularen basierender Metadateneditor, wie er in Abschnitt 5.1.1 beschrieben wurde. Mit seiner Hilfe konnten Metadatensätze geladen, bearbeitet und gespeichert werden. Die Erstellung neuer Metadatensätze basierte auf einer vollständig leeren Eingabemaske. Das heißt, dass bei diesem Verfahren keine Hilfen, wie beispielsweise die Verwendung von Metadaten-schablonen, zum Einsatz kamen. Da keiner der im vorangehenden Abschnitt beschriebenen aktiven Autoren Erfahrungen mit der Erstellung von Metadaten hatte, wurden diese Projektteilnehmer in einer Schulung mit dem Konzept und den Hintergründen zur Erstellung von Metadaten vertraut gemacht. Hierbei wurden sie auch in das Metadatenchema und die Bedienung des Metadateneditors eingewiesen.

Trotzdem erwies sich die Metadatenerstellung mit dem Metadateneditor als problematisch. Schon nach kurzem Betrieb des Metadateneditors zeigte sich, dass die-

ser nicht von den Autoren angenommen wurde. In einem Zeitraum von 101 Tagen wurden lediglich 27 Metadatenätze von den in 8.1 beschriebenen Autoren erstellt. Aus zahlreichen Rückfragen seitens der Autoren ergab sich, dass das Metadaten-schema trotz eingeschränkter Menge obligatorischer Datenfelder zu komplex und damit auch die Eingabemaske des Metadateneditors zu komplex war. Die Analyse der erstellten Metadatenätze ergab weiterhin, dass sie nicht alle als obligatorisch definierten Datenfelder enthalten haben, also unvollständig waren.

Aus diesen Gründen und zur Erleichterung der Metadatenerstellung wurde der Metadateneditor um die weitestgehend automatisierte Metadatengenerierung mit Hilfe des Metadaten-Wizards nach den Konzepten, die in Kapitel 5 beschrieben wurden, erweitert. Mit Einführung des Metadaten-Wizards wurde die Metadatenerstellung weitestgehend automatisiert. Dabei wurden soweit wie möglich Informationen aus dem Benutzerprofil der Autoren und dem zu beschreibenden Lernobjekt gezogen und ein Großteil der auszufüllenden Metadatenfelder automatisch bestimmt. Metadaten, die trotzdem manuell erfasst werden mussten, wurden in einer einfachen Eingabemaske mit vier Eingabefeldern abgefragt. Zusammen aus den manuell abgefragten und den automatisch generierten Metadaten wurde ein Metadatenatz erstellt, der alle obligatorischen Metadaten enthielt. Dieser konnte anschließend im bekannten Metadateneditor nachträglich manuell bearbeitet werden. Bevor der Metadatenatz gespeichert wurde, wurde er auf Vollständigkeit geprüft, um sicherzustellen, dass bei der manuellen Bearbeitung keine Datenfelder gelöscht wurden.

Ziel der Evaluation des Metadaten-Wizards ist die Überprüfung, ob die, durch die Einführung des Metadaten-Wizard erhofften, Verbesserungen bei der Erstellung von Metadaten eingetroffen sind. Dabei gilt es, folgende Hypothesen zu überprüfen:

H1 Der Metadaten-Wizard erleichtert die Metadatenerstellung.

H2 Durch die Integration des Metadaten-Wizards in den Metadatenerstellungsprozess wird die Akzeptanz der Metadatenerstellung gesteigert.

Für die Überprüfung von Hypothese H1 muss die Arbeitserleichterung festgestellt werden, die durch die Verwendung des Metadaten-Wizards entsteht. Dies soll an Hand einer quantitativen Betrachtung durch Auswertung der Log-Dateien des Metadatenerstellungswerkzeugs erfolgen. Für die Ermittlung der Arbeitserleichterung ist die jeweilige Differenz des Aufwands für die Erstellung von Metadatenätzen, die ohne und mit Unterstützung des Metadaten-Wizards gemessen werden kann, auszuwerten. Objektiv kann eine Arbeitserleichterung jedoch schlecht bewertet werden. Aus diesem Grund wird angenommen, dass die Arbeitserleichterung proportional zur Zeitersparnis ist, so dass für die weiteren Betrachtungen die Dauer für die Erstellung von Metadatenätzen verglichen werden kann.

Die Messung der benötigten Dauer zur Erstellung eines Metadatenatzes wurde über die Log-Dateien des Metadatenerstellungswerkzeugs durchgeführt. Dabei

beginnt die Zeitmessung für jeden Metadatensatz mit dem Eintreffen der Anforderung, einen neuen Metadatensatz zu erstellen und endet mit dem Eintrag in der Logdatei, dass der Metadatensatz gespeichert ist.

Für den Vergleich sollten die Zeiten herangezogen werden, die bei der Metadatenerstellung in der Betriebsphase ohne Metadaten-Wizard gemessen wurden. Diese sollten dann mit den gemessenen Werten aus der Betriebsphase mit Unterstützung durch den Metadaten-Wizard verglichen werden. In den beiden Betriebsphasen haben dieselben Autoren, wie in Abschnitt 8.1 beschrieben, Metadaten erstellt. Daraus ergibt sich der folgende Ablauf: Vor der Durchführung der Messung wurden die Autoren zunächst geschult. Im Anschluss an die Schulung begann die erste Betriebsphase mit einer Dauer von 101 Tagen, in der die Autoren mit einem einfachen formularbasierten Editor Metadaten erstellten. Danach wurde der Metadaten-Wizard installiert und die zweite Phase mit einer Dauer von 185 Tagen begann.

Aufgrund dessen, dass in den beiden betrachteten Phasen jeweils dieselben Autoren jedoch in temporaler Sequenz Metadaten erstellt haben, weist die Untersuchung einen methodischen Fehler auf, da die Autoren in der zweiten Phase bereits die Erfahrungen, die sie in der ersten Phase gesammelt hatten, anwenden konnten. Ein signifikanter Unterschied zwischen den Zeiten für die Erstellung von Metadaten ohne und mit Hilfe des Metadaten-Wizards liefert aus diesem Grund lediglich einen Hinweis zur Bestätigung der Hypothese.

Tabelle 8.1: Mit dem ResourceCenter erstellte Metadatensätze

Metadaten-Wizard	Zeitraum	Metadaten-sätze	Dauer [sec] Mittelwert	Standard-abweichung
ohne	101 Tage	27	208	313
mit	185 Tage	670	201	237

Tabelle 8.1 zeigt die Messwerte der beiden Phasen im Vergleich. In der zweiten Betriebsphase mit einer Dauer von 185 Tagen wurden insgesamt 670 Metadatensätze erstellt. Die Differenz der Mittelwerte der gemessenen Zeiten für die Erstellung von Metadaten ohne und mit Hilfe des Metadateneditors unterscheiden sich um 7 Sekunden beziehungsweise 3,37% pro Metadatensatz. Damit liegt die durchschnittliche Zeitersparnis für die Erstellung von Metadaten mit Hilfe des Metadaten-Wizards bei 7 Sekunden pro Metadatensatz. Das entspricht keiner signifikanten Zeitersparnis durch die Verwendung des Metadaten-Wizards, so dass die Betrachtung keinen Hinweis zur Bestätigung der Hypothese H1 liefert.

Eine Sichtung der Metadatensätze, die in der ersten Phase erstellt worden sind, liefert jedoch den Grund für die geringe Differenz der Mittelwerte: Die wenigsten Metadatensätze beinhalten Werte für die vollständige Menge der obligatorischen

Datenfelder. Das ist möglich, weil in der ersten Phase die Metadatenansätze nicht auf Vollständigkeit überprüft wurden. Durch das Erstellen unvollständiger Metadatenansätze, wurde so ein Zeitvorteil erzielt, der die Messung verfälscht. Einen Hinweis hierfür liefert die Standardabweichung der Dauer für die Metadatenerstellung. Sie ist in der ersten Phase größer, weil sich der geleistete Aufwand für die Erstellung von Metadaten in Abhängigkeit der Vollständigkeit der Metadatenansätze zwischen den einzelnen Autoren stark unterscheiden. Durch die Verwendung des Metadaten-Wizards wird jedoch die Erstellung SCORM-konformer Metadatenansätze, die die obligatorischen Datenfelder enthalten, sichergestellt. Um so beachtlicher ist, dass durch die Verwendung des Metadaten-Wizards die durchschnittliche Dauer für das Erstellen von Metadatenansätzen reduziert wurde, obwohl mehr Metadaten erfasst wurden als von den Autoren in der ersten Phase eingegeben wurden. Daraus lässt sich ableiten, dass durch die Verwendung des Metadaten-Wizards unter der Annahme, dass sämtliche obligatorische Metadaten erfasst werden, eine deutliche Erleichterung bei der Erstellung von Metadaten erzielt werden kann.

Für die Überprüfung der Hypothese H2 wird auf die Rückmeldung seitens der Autoren und die Messergebnisse der beiden Betriebsphasen des Metadatenerstellungswerkzeugs zurückgegriffen. Die Hypothese kann bei überwiegend positiver Rückmeldung und deutlichem Anstieg erstellter Metadatenansätze in der zweiten Phase bestätigt werden.

Nach Einführung des Metadaten-Wizards reduzierte sich der Aufwand für die Unterstützung der Autoren bei der Erstellung der Metadaten. Hinzu kamen positive Rückmeldungen seitens der Autoren, die eine Zufriedenheit beziehungsweise verbesserte Akzeptanz gegenüber der Metadatenerstellung erkennen ließen. Auch die Anzahl der in der zweiten Phase erstellten Metadatenansätze im Vergleich zur ersten Phase ist ein Beleg für die gestiegene Akzeptanz bei den Autoren gegenüber der Metadatenerstellung. In der zweiten Phase wurden von den Autoren im Vergleich zur ersten Phase ungefähr 12,5-mal mehr Metadatenansätze erstellt. Aufgrund dieser Reaktionen beziehungsweise der Anzahl der erstellten Metadatenansätze kann Hypothese H2 aufrecht erhalten werden.

Zusammenfassend ergibt die Untersuchung, dass durch die Integration des Metadaten-Wizards in den Metadatenerstellungsprozess die Erstellung von Metadaten vereinfacht werden konnte. Gemessen an der Anzahl der erfassten Metadaten verbesserte sich durch die Verwendung des Metadaten-Wizards die Qualität der Metadatenansätze, ohne dass hierfür mehr Aufwand seitens der Autoren betrieben werden musste.

8.3 **Verwendbarkeit der Autorenumgebung**

Ziel der in diesem Abschnitt erläuterten Untersuchung ist die Überprüfung der Eignung des ResourceCenters und der zugrunde liegenden Konzepte (vergleiche

Abschnitt 4.3) für die tägliche Erstellung von WBTs. Folgende Hypothese wird aufgestellt.

H3 Die Autorenumgebung des ResourceCenters ist tauglich für den praktischen Einsatz.

Als Indikator für die Tauglichkeit der auf Aggregation basierenden Autorenumgebung für den praktischen Einsatz, werden die Anzahl der mit Hilfe der Autorenumgebung erstellten und im Lernobjektarchiv gespeicherten Lernobjekte und Rückmeldungen seitens der Autoren herangezogen. Bei überwiegend positiver Rückmeldung von den Autoren und vergleichbarer oder höherer Anzahl von Lernobjekten, die mit Hilfe der Autorenumgebung erstellt und im Lernobjektarchiv abgelegt wurden, kann die Hypothese bestätigt werden.

Seit Bereitstellung der Autorenumgebung haben die in Abschnitt 8.1 beschriebenen 13 Autoren in einem Zeitraum von neun Monaten 4473 Lernobjekte im ResourceCenter abgelegt beziehungsweise mit Hilfe der integrierten Autorenumgebung erstellt.

Tabelle 8.2: Erstellte und gespeicherte Lernobjekte nach Typ

Typ	Anzahl	Anteil
Kurse	205	4,58%
Lektionen	1760	39,35%
Tabellen	91	2,03%
Flash	1211	27,08%
Bilder	536	11,98%
Molekülanimationen	2	0,04%
Testitems	668	14,94%
Gesamt	4473	100,00%

Tabelle 8.2 zeigt eine Aufschlüsselung der im ResourceCenter gespeicherten Lernobjekte nach Typ. Von den insgesamt 205 Kursen sind im Mittel circa 16,8 Kurse von jedem Autor erstellt worden. Die Kurse setzen sich durchschnittlich aus 8,6 Lektionen und 3,3 Testitems zusammen. Insgesamt wurden 1749 Lernobjekte vom Aggregationsniveau Rohmaterial, wie Bilder, Flash und Molekülanimationen in diese Lektionen eingebunden. Das entspricht durchschnittlich einem Bild, Flash oder Molekülanimationen in jeder Lektion. Unter Berücksichtigung der erstellten Tabellen, die ebenfalls einzeln wiederverwendet werden können, setzt sich ein Kurs

durchschnittlich aus 20,8 Lernobjekten zusammen. Das heißt, dass durchschnittlich 21,8 Metadatenätze zur Erstellung eines Kurses angelegt werden müssen.

Bezüglich der Menge im Lernobjektarchiv abgelegter Lernobjekte ist der Vergleich mit dem *Knowledge Pool System* (KPS) des Ariadne-Projekts interessant. Nach [NTD03] wurden in einem Zeitraum von sieben Jahren im Vergleich nur 3900 Lernobjekte abgelegt und mit Metadaten beschrieben. Die Zahlen sind jedoch nicht direkt vergleichbar, weil im Ariadne-KPS ca. 80% aller gespeicherten Lernobjekte ein Aggregationsniveau von Kursen haben. Im Vergleich dazu sind es im ResourceCenter nur 4,6%. Je höher das Aggregationsniveau der Lernobjekte ist, desto höher wird die durchschnittliche Zeit für die Erstellung der Lernobjekte liegen. Trotzdem ist die Anzahl der im k-MED-Projekt erstellten Lernobjekte inklusive der Metadatenbeschreibungen in Bezug auf die unterschiedlichen Betriebszeiten von Ariadne und dem ResourceCenter beachtlich.

Auf Basis der Zahlen der erstellten Lernobjekte insbesondere im Vergleich mit Ariadne und den überwiegend positiven Rückmeldungen seitens der Autoren kann Hypothese H3 bestätigt werden. Damit ist die Eignung für die tägliche Erstellung von WBTs des auf Aggregation basierenden Autorenprozesses, der in der Autorenumgebung des ResourceCenter umgesetzt wurde, belegt.

8.4 Wiederverwendung im ResourceCenter

In diesem Abschnitt erfolgt die Betrachtung der Wiederverwendung von Lernobjekten im ResourceCenter, die in der neunmonatigen Betriebsphase, die im vorangehenden Abschnitt beschrieben wurde, erstellt wurden. Da für einen Vergleich der Wiederverwendung von Lernobjekten keine Werte vorliegen, wird in diesem Abschnitt auf eine systematische Evaluation verzichtet. Als wiederverwendete Lernobjekte werden bei dieser Erhebung Lernobjekte gezählt, die mindestens zweimal als Teil eines anderen Lernobjekts referenziert werden. Das heißt es werden nur Lernobjekte gezählt, die mindestens zweimal verwendet worden sind.

Tabelle 8.3 zeigt die Anzahl der wiederverwendeten Lernobjekte aufgeschlüsselt nach Typ. Angesichts der großen Zahl gespeicherter Lernobjekte im ResourceCenter überrascht die geringe Zahl wiederverwendeter Lernobjekte. Nur sechs der 13 Autoren nutzen die Möglichkeit in der Autorenumgebung, Lernobjekte wiederzuverwenden. Dabei verwendet nur einer dieser Autoren Lernobjekte anderer Autoren wieder. Die geringe Wiederverwendung von Lernobjekten anderer Autoren ist im Projekt k-MED jedoch nicht verwunderlich, weil bei der Auswahl der Autoren auf eine möglichst breite Fächerkombination geachtet worden ist, so dass es für jedes Fachgebiet in der Regel nur einen Autor gibt. Ein weiterer Grund für die geringe Wiederverwendungsrate ist, dass die kritische Masse, ab der ein signifikanter Anstieg der Wiederverwendung von Lernobjekten zu erwarten ist, noch nicht erreicht ist. Forte et al. [FHW⁺99] beziffern die Zeit, in der die kritische Masse an Lernob-

Tabelle 8.3: Wiederverwendete Lernobjekte nach Typ

Typ	Anzahl	Anteil
Lektionen	6	26,09%
Tabellen	1	4,34%
Flash	10	43,48%
Bilder	6	26,09%
Molekülanimationen	0	0,00%
Testitems	0	0,00%
Gesamt	23	100,00%

jekten im Ariadne-KPS erreicht ist, mit einigen Jahren. Damit die kritische Masse wiederverwendbarer Lernobjekte früher erreicht wird, werden in aktuellen Arbeiten Lösungsansätze für den Zusammenschluss verschiedener Lernobjektarchive erarbeitet [TOD05].

Die Aufschlüsselung der wiederverwendeten Lernobjekte nach Typ zeigt aber auch, dass sowohl Lernobjekte mit niedrigem, als auch mit hohem Aggregationsniveau wiederverwendet werden. Dabei ist deutlich eine Tendenz zur Wiederverwendung von Lernobjekten mit niedrigem Aggregationsniveau mit 74% zu erkennen. Ein Grund hierfür ist die noch nicht implementierte Versionierung von Lernobjekten im ResourceCenter. Während Medienobjekte in der Regel im finalen Zustand in die Kurse importiert werden, befinden sich Lektionen oft über einen längeren Zeitraum in Bearbeitung. Ohne Versionierung ist deren Wiederverwendung nur mit Einschränkungen möglich (vergleiche Kapitel 6).

Eine Aufschlüsselung der wiederverwendeten Lernobjekte nach der Anzahl der Wiederverwendungen zeigt das Balkendiagramm in Abbildung 8.1. An dem Balkendiagramm kann abgelesen werden dass ein Großteil der Lernobjekte nur ein Mal wiederverwendet wird. Die Lernobjekte, die mehr als ein Mal wiederverwendet wurden sind speziell für die Wiederverwendung erstellt worden. Dabei handelt es sich um Flash-Dateien, die mittels Parameter in verschiedene Zustände versetzt werden können.

Insgesamt zeigt die Auswertung, dass die Wiederverwendung von Lernobjekten mit den im ResourceCenter umgesetzten Konzepten umgesetzt werden kann, jedoch von den Autoren aufgrund der fehlenden kritischen Masse wiederverwendbarer Lernobjekte und noch nicht implementierter Versionierung nicht ausschöpfend genutzt wird.

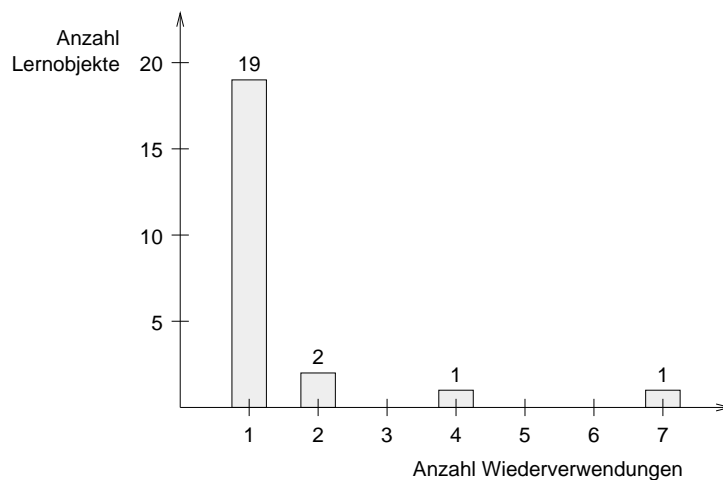


Abbildung 8.1: Wiederverwendung von Lernobjekten

8.5 Zusammenfassung

Allgemein zeigt die Evaluation, dass die in der vorliegenden Arbeit entwickelten Konzepte und deren Umsetzung im ResourceCenter für die tagtägliche Entwicklung von WBTs zur Ergänzung der Präsenzlehre geeignet sind. Allein die Anzahl der im k-MED-Projekt wiederverwendeten Lernobjekte überzeugt nicht. Hauptsächlich ist das mit der breiten Fächerauswahl in k-MED zu begründen, bei der es kaum Überschneidungen gibt und so die Wiederverwendung von Lernobjekten mangels thematischer Überlappung nur durch Vernetzung der Themengebiete erreicht werden kann. Ein anderer Grund hierfür ist das Fehlen der kritischen Masse wiederverwendbarer Lernobjekte.

Die Zeiten für das Erstellen von Metadaten mit Hilfe des einfachen formularbasierten Metadateneditors im Vergleich mit dem Metadaten-Wizard zeigen, dass die Verwendung des Metadaten-Wizards nur eine geringe Zeitersparnis bringt, die jedoch nicht signifikant ist. Das liegt daran, dass bei der rein manuellen Erfassung die Metadaten nicht vollständig erstellt wurden. Unter zusätzlicher Betrachtung der Vollständigkeit der erstellten Metadaten, kann der Metadaten-Wizard deshalb als erfolgreich angesehen werden, da durch seine Anwendung bei kürzerer Bearbeitungszeit vollständige Metadatensätze erstellt werden konnten. Bemerkenswert ist auch die nach der Installation des Metadaten-Wizards gemessene Steigerung der Arbeitsleistung, die durch eine größere Akzeptanz der Metadatenerstellung bei den Autoren durch die Verwendung des Metadaten-Wizards zu erklären ist.

Von der allgemein gestiegenen Akzeptanz der Erstellung von Metadaten mit Hilfe des Metadaten-Wizards profitiert auch die Autorenumgebung des ResourceCenters, da von den Kursen bis zu den Rohmaterialien alle Lernobjekte mit Metadaten beschrieben werden müssen, wenn sie in wiederverwendbarer Form zur Verfügung

gestellt werden sollen. Das belegt die Menge der Kurse, Lektionen und Medienobjekte, die von der Gruppe der aktiven Autoren im k-MED-Projekt erstellt und mit Metadaten beschrieben worden sind. Dabei wurden in einem Zeitraum von acht Monaten 4473 Metadatenätze angelegt. Das übersteigt die Menge der pro Zeiteinheit erstellten Metadatenätze im Vergleich zu anderen Projekten bei weitem.

Aufgrund des nicht fachspezifischen Dokumentmodells ist das ResourceCenter nicht beschränkt auf medizinische Fachinhalte. Das ResourceCenter wird in anderen Projekten bereits erfolgreich für die Erstellung von WBTs in der Linguistik und Informatik eingesetzt, wenn auch in bislang geringerem Umfang.

9 Zusammenfassung und Ausblick

9.1 Zusammenfassung

Die Motivation der vorliegenden Arbeit liegt im Wunsch nach einer Erhöhung des Return of Investments von Lernobjekten. Dies wird im besonderen Hinblick auf die Wiederverwendung bestehender Lernobjekte und deren Fragmente untersucht. Das Ziel der vorliegenden Dissertationsschrift war es deshalb, einen geeigneten Autorenprozess zu entwickeln, innerhalb dessen die Wiederverwendung von Lernobjekten und ihrer Fragmente unterstützt wird. Dabei sollte in dieser Arbeit nicht allein eine Analyse und Optimierung dieses Autorenprozesses erfolgen, sondern es sollten die entwickelten Konzepte durch Implementierung und Evaluation des praktischen Einsatzes überprüft werden.

Der dieser Arbeit zugrunde liegende Ansatz sieht die Wiederverwendung von Lernobjekten und ihrer Fragmente sowohl in unveränderter als auch in angepasster Form während der Erstellung neuer Lernobjekte in einer Autorenumgebung vor. Dazu ist es notwendig, dass eine partielle Wiederverwendung von Lernobjekten, wie beispielsweise einer Einleitung eines Themas oder eines Anwendungsbeispiels, und deren Anpassung an den neuen Lernkontext ermöglicht wird. Die Wiederverwendung bereits bestehender Lernobjekte schließt dabei sowohl eigene als auch Lernobjekte fremder Autoren ein. Um juristische Implikationen der Wiederverwendung ausklammern zu können, wurde die Modellannahme getroffen, dass sämtliche verarbeiteten Inhalte einer auf digitale Inhalte bezogenen Lizenz nach dem Muster der GNU General Public License (GPL) der Free Software Foundation [11] unterliegen.

Weiterhin besteht der Ansatz darin, den konzeptionierten *Autorenprozess* durch eine weitestgehend automatisierte Unterstützung der Wiederverwendung von Lernobjekten zu optimieren. Im Mittelpunkt standen hierbei gleichermaßen bestehende Lernobjekte als auch mit Hilfe der Autorenumgebung neu erstellte Lernobjekte. Insbesondere sollte die Modularisierung der Lernobjekte durch ein modulares Dokumentmodell unterstützt werden. Außerdem sollte eine weitestgehend automatisierte Erfassung von Metadaten unterstützt werden, die sich für die Generierung von Metadaten schon während der Erstellung neuer Lernobjekte eignet, aber auch für Lernobjekte genutzt werden kann, die mit externen Werkzeugen erstellt worden sind. Für die Wiederverwendung von Lernobjekten in angepasster Form sollte die Versionierung der Lernobjekte und ihrer Fragmente im Autorenprozess unter-

stützt werden. Hierfür sollte ein für das Dokumentmodell und den Autorenprozess geeignetes Versionierungskonzept integriert werden.

Zunächst wurde als Beitrag der vorliegenden Arbeit ein abstraktes Prozessmodell für einen auf Aggregation von Lernobjekten basierenden Autorenprozess entwickelt. Dieses Prozessmodell dient als Grundlage für die Entwicklung von EditorKomponenten für Lernobjekte verschiedener Aggregationsniveaus und beinhaltet sowohl die Wiederverwendung bereits bestehender als auch die Erstellung neuer Lernobjekte. Diese konzeptionelle Integrationsleistung im Prozessmodell ist neuartig. Das abstrakte Modell des Autorenprozesses macht keine Einschränkungen bezüglich der Quellen, verschiedener Lernobjektarchive oder der Dateisysteme, aus denen Lernobjekte zur Wiederverwendung stammen können.

Grundlage für den vorgestellten, neuartigen Autorenprozess ist die Möglichkeit, Lernobjekte in modularisierter Form zur Verfügung zu stellen oder sie in eine modularisierte Form zu überführen. Aus diesem Grund ist im Rahmen dieser Arbeit ein innovatives, *generisches Dokumentmodell* entwickelt worden, das eine modularisierte Speicherung von Lernobjekten und Lernobjektfragmenten erlaubt und somit den Anforderungen eines auf Aggregation basierenden Autorenprozesses genügt.

Qualitativ hochwertige Metadaten spielen eine Schlüsselrolle bei der Wiederverwendung von Lernobjekten, da sie in der Regel für die Indizierung und das Finden wiederverwendbarer Lernobjekte herangezogen werden. Deshalb wurde in der vorliegenden Arbeit ein neues Verfahren für die erleichterte, *semiautomatische Generierung* von Metadaten entwickelt. Dieses Verfahren stellt eine Rekombination verschiedener Methoden der Metadatengenerierung dar, die hinsichtlich der Zuverlässigkeit der generierten Metadaten und der Reduzierung des Arbeitsaufwandes bei der lernstandardkonformen Erstellung von Metadaten im Autorenprozess optimiert wurde.

Die *Versionierung* von Lernobjekten ist die Voraussetzung für die gleichzeitige Verwendung von Lernobjekten in verschiedenen Versionen. Aus diesem Grund wurde auf Basis des Dokumentmodells ein Versionierungskonzept entworfen, das sich von bestehenden Ansätzen insbesondere dadurch unterscheidet, dass alle für die Versionierung notwendigen Daten in den Metadaten der Lernobjekte gespeichert werden und dass es die Verwendung in einem auf Aggregation basierenden Autorenprozess berücksichtigt.

Das *ResourceCenter* ist die Implementierung des in dieser Arbeit untersuchten Autorenprozesses. Im ResourceCenter werden die in dieser Arbeit entwickelten Konzepte für die Wiederverwendung von Lernobjekten in einer auf Aggregation basierenden Autorenumgebung zusammengefügt. Diese Autorenumgebung besteht im Wesentlichen aus drei ineinander geschachtelten EditorKomponenten, die vom abstrakten Prozessmodell eines auf Aggregation basierenden Autorenprozess abgeleitet sind.

Das ResourceCenter wird produktiv in verschiedenen Bereichen eingesetzt. Im Rahmen des Projektes k-MED wird das ResourceCenter für die Erstellung von

Kursen verwendet, die die Präsenzlehre der vorklinischen Fächer des Medizinstudiums an hessischen Universitäten ergänzen sollen. Auf Basis dieser Nutzung konnten die im ResourceCenter umgesetzten Konzepte der vorliegenden Arbeit untersucht werden.

Die Menge der erstellten Kurse (WBTs) zeigt, dass das ResourceCenter im produktiven Betrieb für die Erstellung von WBTs geeignet ist. Aufgrund des nicht fachspezifischen Dokumentmodells ist das ResourceCenter nicht beschränkt auf medizinische Fachinhalte. Das ResourceCenter wird in anderen Projekten bereits erfolgreich für die Erstellung von WBTs in der Linguistik und Informatik eingesetzt.

Im Vergleich mit verwandten Arbeiten beschreibt die vorliegende Arbeit erstmals ein abstraktes Modell eines auf Aggregation basierenden Autorenprozesses mit dessen Hilfe Lernobjekte mit einem breiten Spektrum von Aggregationsniveaus kombiniert werden können. Als Basis für den Autorenprozess wurde ein Dokumentmodell entwickelt, das die Aggregation und zugleich die Modularisierung der Lernobjekte beibehält. Der wichtigste Unterschied dieses Dokumentmodells zu bisherigen Arbeiten ist die direkte Indizierbarkeit von Lernobjekten vieler verschiedener, insbesondere kleiner, Aggregationsniveaus. Auch die Unterstützung der Versionierung von Lernobjekten und aller ihrer Fragmente unterscheidet das vorgestellte Dokumentmodell von den meisten verwandten Dokumentmodellen. Mit dem Metadaten-Wizard wird ein Verfahren zur semiautomatischen Generierung von Metadaten vorgestellt, das im Vergleich zu vielen anderen Arbeiten keine Insellösung darstellt, sondern in den auf Aggregation basierenden Autorenprozess integriert worden ist, damit so die Erstellung von Metadaten schon bei der Erstellung von Inhalten weitestgehend automatisch erfolgen kann.

Auf Aggregation basierende Autorenprozesse sind geeignet, Entwicklungskosten für Selbstlernmaterialien zu reduzieren. Die vorliegende Arbeit hat einen wichtigen Beitrag geliefert, der Vision von Autorenumgebungen der neuen Generation mit Autorenprozessen, die auf Aggregation basieren, einen Schritt näher zu kommen.

9.2 Ausblick

Die vorliegende Arbeit liefert einen signifikanten Beitrag im Bereich der Wiederverwendung von Lernobjekten und deren Fragmente in einem auf Aggregation basierenden Autorenprozess. Die im Rahmen der Arbeit entwickelten Konzepte und deren Anwendung werfen weiterführende Fragen auf. Dazu gehören aus technischer Sicht beispielsweise der Import und die damit einhergehende automatische Modularisierung bereits bestehender Lernobjekte, die in monolithischen Dokumentformaten vorliegen, um auch diese einem auf Aggregation basierenden Autorenprozess zuzuführen. Unter wirtschaftlichen Gesichtspunkten ist die Entwicklung geeigneter Lizenz- und Geschäftsmodelle für modulare Lerninhalte, die zudem für die Wiederverwendung durch die Autoren angepasst werden können, von Interesse.

Ebenfalls wertvoll wäre eine weitergehende Unterstützung der Autoren durch eine detailliertere Signalisierung der Wiederverwendungsprozesse in der Multiautoren-umgebung. Weitere Potenziale für eine Verbesserung des Autorenprozesses könnten sich zudem durch die Einbindung von Feedback-Prozessen seitens der Nutzer der Lernobjekte an die jeweiligen Autoren ergeben.

Zusammenfassend ist festzustellen, dass die entwickelten Konzepte und deren Implementierung im ResourceCenter bereits erfolgreich in verschiedenen Anwendungsgebieten eingesetzt werden konnten und auch für die Anwendung in anderen Gebieten geeignet sind.

Literaturverzeichnis

- [AB04] Lee F. Alderman und Chuck Barrit: *Creating a Reusable Learning Objects Strategy: Leveraging Information and Learning in a Knowledge Economy*. Jossey-Bass, San Francisco, 2004.
- [Adv01] Advanced Distributed Learning Initiative: *Sharable Content Object Reference Model (SCORM) Version 1.2*, 2001.
- [AH04] Mario Aehnelt und Sybille Hambach: *CSI-Studie - Informationen und Empfehlungen der Content Sharing Initiative zu Modularisierung von Lernmaterial und Learning Object Repositories*. Fraunhofer-Institut für Graphische Datenverarbeitung, 2004.
- [AIC04] AICC - Aviation Industry CBT Committee: *CMI Guidelines for Interoperability*, August 2004.
- [AKKRM03] Khaldoun Ateyeh, Michael Klein, Birgitta König-Ries und Jutta Mülle: *A Practical Strategy for the Modularization of Courseware*. In: *Tagungsband zur 2. Konferenz Professionelles Wissensmanagement Erfahrungen und Visionen*, 2003.
- [AM02] Khaldoun Ateyeh und Jutta Mülle: *Making Courseware Reusable*. In: *Proceedings of World Congress on Networked Learning in a Global Environment, Challenges and Solutions for Virtual Education*, 2002.
- [AML00] Khaldoun Ateyeh, Jutta Mülle und Peter Lockemann: *Modular Development of Multimedia Courseware*. In: *Proceedings of First International Conference on Web Information Systems Engineering (WISE'00) - Volume 2*, Hong Kong, China, Juni 2000.
- [BA04] Chuck Barritt und Lee F. Alderman: *Creating a Reusable Learning Objects Strategy: Leveraging Information and Learning in a Knowledge Economy*. Pfeiffer, San Francisco, 2004.
- [BCV03] C. Brooks, J. Cooke und J. Vassileva: *Versioning of Learning Objects*. In: *Proceedings of International Conference on Advanced Learning Technologies*, Athen, Griechenland, Juli 2003.

- [BDK02] Y. Bourda, B. Doan und W. Kekhia: *A Semi-automatic Tool for the Indexation of Learning Objects*. In: *Proceedings of the World Conference on Educational Multimedia, Hypermedia and Telecommunications*, Seiten 190 – 191, 2002.
- [BK99] Susanne Boll und Wolfgang Klas: *ZYX - A Semantic Model for Multimedia Documents and Presentations*. In: *Proceedings of the IFIP TC2/WG2.6 Eighth Working Conference on Database Semantics- Semantic Issues in Multimedia Systems table of contents*, Seiten 189 – 209. Kluwer, B.V., 1999.
- [Bor99] Christine L. Borgman: *What are digital libraries? Competing visions*. *Information Processing and Management*, (35):227 – 243, 1999.
- [Boy02] Tom Boyle: *Design Principles for Authoring Dynamic, Reusable Learning Objects*. In: A. Williamson, C. Gunn, A. Young und T. Clear (Herausgeber): *Winds of Change in the Sea of Learning: Proceedings of the 19th Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education*, Seiten 57 – 64. Institute of Technology Auckland, 2002.
- [BRG⁺04] Udo Bleimann, Horst Röder, Ulrich Gojny, Ingo Stengel, Angela Leichtweiß und Gerhard Knorz: *Von ELAT zu Atlantis University. Querschnitt (Beiträge aus Forschung und Entwicklung der FH Darmstadt)*, (18):8 – 15, 2004.
- [BYRN99] R.A. Baeza-Yates und B. Ribeiro-Neto: *Modern Information Retrieval*. ACM Press and Addison-Wesley, Reading, Ma., 1999.
- [CD03] Kris Cardinaels und Erik Duval: *Composite Learning Objects: Exposing the Components*. In: *Proceedings of the 3rd Annual ARIADNE Conference*, 2003.
- [CDH⁺04] T. Carlton, B. DeNise, J. Hyde, W. McDonald, A. Montgomery, W. Shook, S. Smith, G. Tourigny, G. Uhrich und N. Cramer: *Reusability Analysis of Reusable Objects*. AICC Management and Processes Subcommittee, 2004.
- [CH01] Jörg Caumanns und Silvia Hollfelder: *Web-basierte Repositorien zur Speicherung, Verwaltung und Wiederverwendung multimedialer Lernfragmente*. In: Ralph Schmidt (Herausgeber): *Tagungsband Information Research & Content Management - Orientierung, Ordnung und Organisation im Wissensmarkt. 23. Online Tagung der DGI (Deutsche Gesellschaft für Informationswissenschaft und Informationspraxis)*, ISBN 3-925474-43-9, Seiten 130 – 140, Frankfurt a.M., Mai 2001.

- [Cis03] Cisco Systems Inc.: *Reusable Learning Object Strategy: Designing and Developing Learning Objects for Multiple Learning Approaches*, 2003.
- [CMD05] Kris Cardinaels, Michael Meire und Erik Duval: *Automating Metadata Generation: the Simple Indexing Interface*. In: *Proceedings of 14th International World Wide Web Conference*, Chiba, Japan, 2005.
- [CTZ00] S. Chien, V. Tsotras und C. Zaniolo: *A Comparative Study of Version Management Schemes for XML Documents*. Technischer Bericht 8/2000, TimeCenter, 2000.
- [DC003] *ISO TC 46/SC 4 N515 Information and documentation - The Dublin Core metadata element set*, 2003.
- [DFC⁺01] Erik Duval, Eddy Forte, Kris Cardinaels, Bart Verhoeven, Rafael Van Durm, Koen Hendrikx, Maria Wentland Forte, Norbert Ebel, Maciej Macowicz, Ken Warkentyne und Florence Haenni: *The Ariadne Knowledge Pool System*. *Communications of the ACM*, 44(5):73 – 78, 2001.
- [DH98] F. Dawson und T. Howes: *vCard MIME Directory Profile*. Internet Engineering Task Force, Reston, USA, September 1998.
- [DH03] Erik Duval und Wayne Hodgins: *A LOM Research Agenda*. In: G. Hencsey, B. White, Y. Chen, L. Kovacs und S. Lawrence (Herausgeber): *Proceedings of the twelfth international conference on World Wide Web*, Seiten 1 – 9, 2003.
- [Dob99] Matthias Dobratz, Susanne und Schulz: *Dissertationen in SGM-L/XML - Erprobung eines Publikationskonzeptes*. In: *Konferenzband der ISKO99*, 1999.
- [Dow01] Stephen Downes: *Learning Objects: Resources for Distance Education Worldwide*. *International Review of Research in Open and Distance Learning*, Juli 2001. ISSN: 1492-3831.
- [Eul01] Dieter Euler: *Handbuch E-Learning*, Kapitel Selbstgesteuertes Lernen mit Multimedia und Telekommunikation gestalten, Seiten 1 – 19. Deutscher Wirtschaftsdienst, Köln, 2001.
- [Faa04] Andreas Faatz: *Ein Verfahren zur Anreicherung fachgebietsspezifischer Ontologien durch Begriffsvorschläge*. Dissertation, Technische Universität Darmstadt, November 2004.

- [FHW⁺99] Eddy Forte, F. Haenni, K. Warkentyne, Erik Duval, Kris Cardinaels, E. Vervaet, K. Hendrikkx, Maia Wentland-Forte und F. Simillion: *Semantic and Pedagogic Interoperability Mechanisms in the Ariadne Educational Repository*. ACM SIGMOD Record, 28(1):20 – 25, März 1999.
- [FMMW05] Emmanuel Fernandes, Hend Madhour, Sami Miniaoui und Maia Wentland: *Phoenix: An XML-Based Tool for E-Learning Environment*. In: *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications (EDMEDIA) 2005*, 2005.
- [Fre02] Burkhard Freitag: *LMML - Eine XML - Sprachfamilie für eLearning Content*. In: Norbert Jesse Sigrid E. Schubert, Bernd Reusch (Herausgeber): *Informatik bewegt: Informatik 2002 - 32. Jahrestagung der Gesellschaft für Informatik e.v. (GI)*, Dortmund, Germany, 2002.
- [GA03] Wolfgang Gerteis und Michael Altenhofen: *E-Learning-Services im Spannungsfeld von Pädagogik, Ökonomie und Technologie*, Kapitel Von der Didaktik zur Technik: Dynamische Kursverläufe, Seiten 95 – 130. W. Bertelsmann Verlag, Bielefeld, 2003.
- [Gap93] D. Kaye Gapen: *The virtual library: knowledge, society, and the librarian*, Kapitel 1, Seiten 1 – 14. Meckler Corporation, 1993, ISBN 0-88736-860-3.
- [GAP02] Ines Grützner, Niniek Angkasaputra und Dietmar Pfahl: *A Systematic Approach to Produce Small Courseware Modules for Combined Learning and Knowledge Management Environments*. In: *Proceedings of 14th International Conference on Software Engineering and Knowledge Engineering*, Ischia, Italia, 2002.
- [GCDRL03] Jane Greenberg, Abe Crystal, W. Davenport Robertson und Ellen Leadem: *Iterative Design of Metadata Creation Tools for Resource Authors*. In: S. Sutton, J. Greenberg und J. Tennis (Herausgeber): *Proceedings of the 2003 Dublin Core Conference: Supporting Communities of Discourse and Practice – Metadata Research and Applications*, 2003.
- [Ger03] Wolfgang Gerteis: *E-Learning-Services im Spannungsfeld von Pädagogik, Ökonomie und Technologie*, Kapitel L3-Lebenslanges Lernen - Weiterbildung als Grundbedürfnis, Seiten 21 – 47. W. Bertelsmann Verlag, 2003.

- [GFT98] Bilge Günsel, A. Müfit Ferman und A. Murat Tekalp: *Temporal video segmentation using unsupervised clustering and semantic object tracking*. SPIE Journal of Electronic Imaging, 7(3):592 – 604, 1998.
- [GPPR01] Jane Greenberg, Maria Cristina Pattuelli, Bijan Parsia und W. Davenport Robertson: *Author-generated Dublin Core Metadata for Web Resources: A Baseline Study in an Organization*. Journal of Digital Information (JoDI), 2(78), November 2001.
- [GSM⁺03] Eicke Godehardt, Klara Schroeder, Jennifer Mockenhaupt, Thomas Trebing, Guido Rößling und Jana Trnková: *Ein System zur Unterstützung verteilter Autorengruppen innerhalb einer E-Learning Umgebung*. In: *Tagungsband zum Workshop Neue Medien in der Aus- und Weiterbildung von Bauingenieuren und Architekten*, Seiten 1 – 12, Oktober 2003.
- [Haa96] Anja Haake: *Versionenunterstützung für strukturierte Hypertextdokumente im elektronischen Publizieren*. Dissertation, Technische Universität Darmstadt, 1996.
- [Haa02] Jörg M. Haake: *Cooperative Learning of Process Knowledge: Making an Argument for Cooperative Hypermedia-based Learning Environments*. International Journal of Continuing Engineering Education and Life-Long Learning, 12(5/6):448 – 471, 2002.
- [HFM⁺01] Stefan Hoermann, Andreas Faatz, Oliver Merkel, Ansgar Hugo und Ralf Steinmetz: *Ein Kurseditor für modularisierte Lernressourcen auf der Basis von Learning Objects Metadata zur Erstellung von adaptierbaren Kursen*. In: *LLWA 01 - Tagungsband der GI-Workshopwoche „Lernen-Lehren-Wissen-Adaptivität“*, Seiten 315–323. Ralf Klinkenberg, Stefan Rueping, Andreas Fick, Nicola Henze, Christian Herzog, Ralf Molitor, Olaf Schroeder, Oktober 2001. (Research Report #763).
- [HFSS04] Stefan Hoermann, Andreas Faatz, Cornelia Seeberg und Ralf Steinmetz: *ResourceCenter: Eine digitale Bibliothek für Lernressourcen mit Open Source Lizenz*. In: M. Fischer H. Handels R. Klar J. Leven F. Puppe S. Pöppel, J. Bernauer und K. Spitzer (Herausgeber): *Rechnergestützte Lehr und Lernsysteme in der Medizin*, Seiten 113–124, März 2004.
- [HHRS05] Stefan Hoermann, Tomas Hildebrandt, Christoph Rensing und Ralf Steinmetz: *ResourceCenter - A Digital Learning Object Repository with an Integrated Authoring Tool Set*. In: *Proceedings of ED-MEDIA World Conference on Educational Multimedia, Hypermedia & Telecommunications*, Montreal, 2005.

- [HJ03] Stefan Hoermann und Ronny John: *Evaluation von XML-Datenbanken für LOM-Datensätze*. Technischer Bericht TR-KOM-2003-08, KOM Multimedia Communications Lab, Mai 2003.
- [HJSS03] Stefan Hoermann, Ronny John, Cornelia Seeberg und Ralf Steinmetz: *Evaluation von Datenbanken zur Speicherung von strukturierten Metadaten am Beispiel LOM*. In: *Proceedings der DeLFI 2003 - 1. Fachtagung „e-Learning“ der Gesellschaft für Informatik*, Seiten 215 – 224, September 2003.
- [HKL⁺00] H. J. H. Hermans, E. J. R. Koper, A. Loeffen, J. M. Manderveld und E. M. Rusman: *Edubox-EML Reference Manual (Beta version)*. Open University of the Netherlands, Dezember 2000.
- [Hod02] H. Wayne Hodgins: *The Future of Learning Objects*. In: *Proceedings of the eTEE Conference*, Davos, Switzerland, August 2002.
- [Hor98] Robert E. Horn: *Instructional Development: State of the Art*, Kapitel Structured Writing as a Paradigm. Englewood Cliffs, NJ: Educational Technology Publications, 1998.
- [HRS05] Stefan Hoermann, Christoph Rensing und Ralf Steinmetz: *Wiederverwendung von Lernressourcen mittels Authoring by Aggregation im ResourceCenter*. In: *Tagungsband der 3. Deutschen e-Learning Fachtagung Informatik - DeLFI*, Seiten 153 – 164, Rostock, September 2005.
- [HS02a] Stefan Hoermann und Ralf Steinmetz: *k-MED: Kurse gestalten und adaptieren mit rhetorisch-didaktischen Relationen*. In: *Rechnergestützte Lehr- und Lernsysteme in der Medizin*. J. Bernauer and M. R. Fischer and F. J. Leven and F. Puppe and M. Weber, April 2002.
- [HS02b] Stefan Hoermann und Ralf Steinmetz: *Kurse erstellen mit Learning Object Metadata*. Technischer Bericht TR-2002-05, Multimedia Communications Lab - KOM, November 2002.
- [HSD98] T. Howes, M. Smith und F. Dawson: *A MIME Content-Type for Directory Information, RFC2425*. Internet Engineering Task Force, Reston, USA, September 1998.
- [HSDK⁺03a] Stefan Hoermann, Cornelia Seeberg, Luka Divac-Krnic, Oliver Merkel, Andreas Faatz und Ralf Steinmetz: *Building Structures of Reusable Educational Content Based on LOM*. In: *Proceedings of The 15th Conference On Advanced Information Systems Engineering workshop on Semantic Web for Web-based Learning*, Juni 2003.

- [HSDK⁺03b] Stefan Hoermann, Cornelia Seeberg, Luka Divac-Krnic, Oliver Merkel, Andreas Faatz und Ralf Steinmetz: *Representation of Knowledge as Support for Authors of Reusable Educational Content*. Poster Presentation at The Twelfth International World Wide Web Conference, Mai 2003.
- [HSGS03] Stefan Hoermann, Stefan Schneider, Ulrich Glowalla und Ralf Steinmetz: *Erstellung von SCORM-kompatiblen Kursen im Projekt k-MED*. In: *Rechnergestützte Lehr- und Lernsysteme in der Medizin*, Seiten 103–113. Shaker Verlag, Aachen, April 2003.
- [HSS05] Stefan Hoermann, Cornelia Seeberg und Ralf Steinmetz: *kmed: Knowledge - Based Multimedia Medical Education*, Kapitel E-Learning Repository - ResourceCenter, Seiten 92 – 97. htcc-Verlag, Februar 2005. ISBN 3-00-015378-0.
- [IGdLR⁺03] Leandro Soares Indrusiak, Manfred Glesner, Ricardo Augusto da Luz Reis, Giuliana Alcantara, Stefan Hoermann und Ralf Steinmetz: *Reducing Authoring Costs of Online Training in Microelectronics Design by Reusing Design Documentation Content*. In: *Proceedings of International Conference on Microelectronics Systems Education (MSE'03)*, Seiten 57 – 58, 2003.
- [IMS01a] IMS Global Learning Consortium, Inc.: *IMS Content Packaging Information Model*, 2001.
- [IMS01b] IMS Global Learning Consortium, Inc.: *IMS Learning Resource Meta-Data Information Model*, 2001.
- [IMS03a] IMS Global Learning Consortium, Inc.: *IMS Digital Repositories Interoperability - Core Functions Information Model*, 2003.
- [IMS03b] IMS Global Learning Consortium, Inc.: *IMS Learning Design Information Model*, Januar 2003.
- [Irv03] Kristina M. Irvin: *Comparing Information Retrieval Effectiveness of Different Metadata Generation Methods*. Master's paper for the m.s. in i.s. degree, April 2003.
- [ISO86] ISO: *Information processing - Text and office systems - Standard Generalized Markup Language (SGML)*, ISO 8879, 1 Auflage, 1986.
- [ISO96] ISO/IEC: *Information technology - Processing languages - Document Style Semantics and Specification Language (DSSSL)*, ISO/IEC 10179, 1 Auflage, 1996.

- [JJBW98] Charlotte Jenkins, Mike Jackson, Peter Burden und Jon Wallis: *Automatic classification of Web resources using Java and Dewey Decimal Classification*. Computer Networks and ISDN Systems, 30:646 – 648, 1998.
- [JJBW99] C. Jenkins, M. Jackson, P. Burdon und J. Wallis: *Automatic RDF metadata generation for resource discovery*. Computer Networks, (31):1305 – 1320, 1999.
- [KAKRM03] Michael Klein, Khaldoun Ateyeh, Birgitta König-Ries und Jutta Mülle: *Creating, Filling, and Using a Repository of Reusable Objects for Database Courses*. In: *Proceedings of the Workshop „Datenbanken und E-Learning“ within the GI-Fachtagung „Datenbanksysteme für Business, Technologie und Web (BTW 2003)“*, Leipzig, Germany, Februar 2003.
- [KAM04] S. Khaddaj, A. Adamu und M. Morad: *Object versioning and information management*. Information and Software Technology, 46:491 – 498, 2004.
- [Ker01] Michael Kerres: *Multimediale und telemediale Lernumgebungen: Konzeption und Entwicklung*. Oldenbourg, München, 2. Auflage, 2001.
- [KESS01] Stephan Kassarke, Abdulmotaleb El Saddik und Achim Steinacker: *Learning Objects Metadata in the Area of Operations Research*. In: *Proceedings of the EdMedia, Tampere Finland*, 2001.
- [KMT⁺05] Joris Klerkx, Michael Meire, Stefaan Ternier, Katrien Verbert und Erik Duval: *Information Visualisation: Towards an Extensible Framework for Accessing Learning Object Repositories*. In: *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications (EDMEDIA) 2005*, 2005.
- [KNN99] E. Kimber, S. Newcomb und P. Newcomb: *Version Management as Hypertext Application: Referent Tracking Documents*. In: *Proceedings of Markup Technologies*, Chicago, Illinois, 1999.
- [Kob85] Alfred Kobsa: *Benutzermodellierung in Dialogsystemen*. Berlin - New York: Springer, 1985.
- [Kun99] J. Kunze: *RFC 2731 - Encoding Dublin Core Metadata in HTML*. Dublin Core Metadata Initiative, Dezember 1999.
- [Lie01] Michael Liepert: *Rechte, Benutzerrollen und Inhaltsversionierung für verteilte Multimedia-Autorensysteme*. Dissertation, Technische Universität Darmstadt, 2001.

- [LOM02] *IEEE Standard for Learning Object Metadata.* Ieee standard 1484.12.1-2002, IEEE, November 2002.
- [LTV03] Ulrike Lucke, Djamshid Tavangarian und Denny Voigt: *Multidimensional Educational Multimedia with <ML>³.* In: Allison Rossett (Herausgeber): *Proceedings of E-Learn 2003, World Conference on E-Learning in Corporate, Government, Healthcare, & Higher Education*, November 2003.
- [Mag03] Amanda Magnussen: *Creating digital libraries: A model for digital library development.* In: *Proceedings of 10th Asia Pacific Special Health and Law Librarians Conference*, 2003.
- [MAI00] J. Mason, G. Adcock und A. Ip: *Modeling Information to Support Value Adding: EdNA Online.* WebNet Journal, 2(3):38 – 45, 2000.
- [Mas03] Elliot Masie: *Making Sense of Learning Specifications & Standards: A Decision Maker's Guide to their Adoption, 2nd Edition.* The MASIE Center's e-Learning Consortium, 2003.
- [MBMN96] J.S. Mitchell, J. Beall, W.E. Matthews und G.R. New (Herausgeber): *Dewey Decimal Classification Edition 21 (DDC 21).* Forest Press, New York, 1996.
- [Med03] Norbert Meder: *E-Learning-Services im Spannungsfeld von Pädagogik, Ökonomie und Technologie*, Kapitel L3 - ein didaktisches Modell als Impulsgeber, Seiten 49 – 82. W. Bertelsmann Verlag, Bielefeld, 2003.
- [ND02] Filip Neven und Erik Duval: *Reusable Learning Objects: a Survey of LOM-Based Repositories.* In: *Proceedings of the 10th ACM International Conference on Multimedia*, Seiten 291 – 294, 2002.
- [ND03] Jehad Najjar und Erik Duval: *Mapping between ARIADNE and LOM.* Technischer Bericht, Februar 2003.
- [NDS⁺03] Filip Neven, Erik Duval, Ternier Stefaan, Cardinaels Kris und Vandepitte Pieter: *An open and flexible indexation- and query tool for Ariadne.* In: D. Lassner und C. McNaught (Herausgeber): *Proceedings of the ED-MEDIA 2003 World Conference on Educational Multimedia, Hypermedia, and Telecommunications*, Seiten 107 – 114, 2003.
- [NHHM⁺04] H.M. Niegemann, S. Hessel, D. Hochscheid-Mauel, K. Aslanski, M. Deimann und G. Kreuzberger: *Kompendium E-Learning.* Springer, Berlin, 2004. ISBN: 3-540-43816-5.

- [NTD03] Jihad Najjar, Stefaan Ternier und Erik Duval: *The actual use of metadata in ARIADNE: an empirical analysis*. In: *Proceedings of the 3rd Annual ARIADNE Conference*, Seiten 1 – 6. ARIADNE Foundation, 2003.
- [Pol03] Pithamber R. Polsani: *Use and Abuse of Reusable Learning Objects*. *Journal of Digital Information*, 3(4), 2003.
- [Pri01] Michael Priestley: *DITA XML: A Reuse by Reference Architecture for Technical Documentation*. In: *Proceedings of the 19th annual international conference on Computer documentation table of contents, ISBN:1-58113-295-6*, Seiten 152 – 156, Sante Fe, New Mexico, USA, 2001.
- [Rab89] L. R. Rabiner: *A tutorial on Hidden Markov Models and selected applications in speech recognition*. In: *Proceedings of the IEEE*, Band 77, Seiten 257 – 285, 1989.
- [RBK96] Henry A. Rowley, Shumeet Baluja und Takeo Kanade: *Neural Network-Based Face Detection*. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 1996.
- [Ruy01] Kim E. Ruyle: *Meet Me in RIO: Implementing Reusable Information Objects*. In: *Proceedings of the 5th Annual KM World Conference*, 2001.
- [Sad01] Abdulmotaleb El Saddik: *Interactive Multimedia Learning*. Springer-Verlag, September 2001.
- [Sed98] Robert Sedgewick: *Algorithms in C++, Parts 1-4: Fundamentals, Data Structure, Sorting, Searching, 3rd Edition*. Addison Wesley Professional, 3rd Auflage, Juli 1998. ISBN: 0201350882.
- [See02] Cornelia Seeberg: *Life Long Learning - Modulare Wissensbasen für elektronische Lernumgebungen*. Springer-Verlag, Oktober 2002.
- [SFB99] Christian Süß, Burkhard Freitag und Peter Brössler: *Metamodeling for Web-Based Teachware Management*. In: P.P Chen, D.W. Embley, J. Kouloumdijan, S.W. Liddle und J.F. Roddick (Herausgeber): *Proceedings of International WWWC99 Workshop on World-Wide Web and Conceptual Modeling in conjunction with ER99*, LNCS 1727, Paris, France, 1999. Springer Verlag.
- [SM02] J. B. South und D. W. Monson: *The Instructional Use of Learning Objects*, Kapitel A university-wide system for creating, capturing,

- and delivering learning objects. Agency for Instructional Technology and the Association for Educational Communications and Technology, 2002, ISBN 0-7842-0892-1.
- [SMI05] *Synchronized Multimedia Integration Language (SMIL 2.0)*. W3c recommendation REC-SMIL2-20050107, W3C, Januar 2005.
- [SN02] Ralf Steinmetz und Klara Nahrstedt: *Multimedia Fundamentals Volume 1*. Prentice Hall, Januar 2002.
- [SN04a] Ralf Steinmetz und Klara Nahrstedt: *Multimedia Applications*. Springer, April 2004.
- [SN04b] Ralf Steinmetz und Klara Nahrstedt: *Multimedia Systems*. Springer, April 2004.
- [SSR+99] Cornelia Seeberg, Achim Steinacker, Klaus Reichenberger, Stephan Fischer und Ralf Steinmetz: *Individual Tables of Contents in Web-based Learning Systems*. In: *Proceedings of the 10th ACM Conference on Hypertext and Hypermedia*, Seiten 167–168, Februar 1999.
- [Ste02] Achim Steinacker: *Medienbausteine für web-basierte Lernsysteme*. Dissertation, Darmstadt University of Technology, Januar 2002.
- [SW04] Ralf Steinmetz und Klaus Wehrle: *Peer-to-Peer-Networking and -Computing*. Informatik Spektrum, Aktuelles Schlagwort, 27(1):51 – 54, 2004.
- [Szi05] Andreas Sziegleit (Herausgeber): *k-MED: Knowledge - Based Multimedia Medical Education*. Httc-Verlag, Februar 2005. ISBN 3-00-015378-0.
- [TA03] Wolfgang Theilmann und Michael Altenhofen: *E-Learning-Services im Spannungsfeld von Pädagogik, Ökonomie und Technologie - Lebenslanges Lernen im Bildungsnetzwerk der Zukunft*, Kapitel Flexible Wiederverwendung von Lernobjekten, Seiten 131 – 146. Bertelsmann, Bielefeld, 2003.
- [Tan03] Andrew S. Tanenbaum: *Computer Networks*. Prentice Hall, fourth Auflage, 2003. ISBN: 0-13-066102-3.
- [Tee01] Gunnar Teege: *TLMM: The Teaching and Learning Material Model*. Web, Juni 2001.

- [Tee02] Gunnar Teege: *Reuse of Teaching Materials in Targetteam*. In: *Proceedings of International Workshop on Interactive Computer aided Learning ICL 2002*, Villach, Austria, 2002.
- [TOD05] Stefaan Ternier, Daniel Olmedilla und Erik Duval: *Peer-to-Peer versus Federated Search: towards more Interoperable Learning Object Repositories*. In: *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications (EDMEDIA) 2005*, Seiten 1421 – 1428, 2005.
- [TP91] M.A. Turk und A.P. Pentland: *Face recognition using eigenfaces*. In: *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1991.
- [Tsi99] Dennis Tsichritzis: *Reengineering the University*. *Communications of the ACM*, 42(6):93 – 100, Juni 1999.
- [Urb03] Stefan Urbansky: *Aspekte zur Wiederverwendung von e-Learning-Content*. In: *Proceedings of the Workshop on e-Learning*, 2003.
- [VCVD⁺01] B. Verhoeven, Kris Cardinaels, R. Van Durm, Erik Duval und Henk Olivie: *Experiences with the ARIADNE Pedagogical Document Repository*. In: C. Montgomery und J. Viteli (Herausgeber): *Proceedings of ED-MEDIA 2001 World Conference on Educational Multimedia, Hypermedia & Telecommunications*, Seiten 1949 – 1954, 2001.
- [VD04] Katrien Verbert und Erik Duval: *Towards a global architecture for learning objects: a comparative analysis of learning object content models*. In: *Proceedings of the ED-MEDIA 2004 World Conference on Educational Multimedia, Hypermedia and Telecommunications*, Seiten 202 – 208. AACE, AACE, 2004.
- [VJD⁺05] Katrien Verbert, Jelena Jovanovic, Erik Duval, Dragan Gasevic und Michael Meire: *Ontology-based Learning Content Repurposing: the ALOCoM Framework*. In: *Proceedings of PROLEARN-iClass thematic workshop on „Learning Objects in Context“*, Leuven, Belgium, März 2005.
- [VKM⁺04] Katrien Verbert, Joris Klerkx, Michael Meire, Jehad Najjar und Erik Duval: *Towards a Global Component Architecture for Learning Objects: An Ontology Based Approach*. *Lecture Notes in Computer Science*, 3292, 2004.

-
- [Vol97] S. Volmer: *Tracing Images in Large Databases by Comparison of Wavelet Fingerprints*. In: *Proceedings of the 2nd International Conference on Visual Information Systems*, Seiten 16 – 172, La Jolla, CA, USA, 1997.
- [Wei00] J. Weinheimer: *How to Keep the Practice of Librarianship Relevant in the Age of the Internet*. Vine, Special issue on Metadata, part 1, (116):14 – 27, 2000.
- [Whi99] E. James Whitehead: *Versioning in Hypertext Systems*. 1999.
- [Wil02] David A. Wiley: *The Instructional Use of Learning Objects*, Kapitel Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy, Seiten 1 – 35. Agency for Instructional Technology and the Association for Educational Communications and Technology, 2002, ISBN 0-7842-0892-1.
- [WM05] Norman Walsh und Leonard Mueller: *DocBook: The Definitive Guide*. O'Reilly & Associates, Inc., 2005.
- [XML04] *Extensible Markup Language (XML) 1.0 (Third Edition)*. W3c recommendation REC-xml-20040204, W3C, Februar 2004.
- [XSL01] *Extensible Stylesheet Language (XSL)*. W3c recommendation REC-xsl-20011015, W3C, Oktober 2001.
- [YVT⁺04] M.J. Yoo, B. Vité, J. Trincherini, V. Badan und M. Wentland-Forte: *Phoenix: An authoring tool for hybrid e-learning documents*. In: *Proceedings of the ED-MEDIA 2004 World Conference on Educational Multimedia, Hypermedia and Telecommunications*, Lugano, Switzerland, 2004.
- [YW96] J. Yang und A. Waibel: *A real-time face tracker*. In: *Proceedings of 3rd IEEE Workshop on Applications of Computer Vision*, 1996.

Online Referenzen

- [1] *Advanced Distributed Learning*. <http://www.adlnet.org>, zuletzt geprüft am 21.06.2005.
- [2] *ARIADNE Foundation for the European Knowledge Pool*. <http://www.ariadne-eu.org>, zuletzt geprüft am 21.06.2005.
- [3] *Aviation Industry CBT Committee*. <http://www.aicc.org>, zuletzt geprüft am 21.06.2005.
- [4] *CanCore*. <http://www.cancore.ca>, zuletzt geprüft am 21.06.2005.
- [5] *CAREO - Campus Alberta Repository of Educational Objects*. <http://www.careo.org>, zuletzt geprüft am 21.06.2005.
- [6] *Darwin Information Typing Architecture*. <http://www.ibm.com/developerworks/xml/library/x-dita1>, zuletzt geprüft am 21.06.2005.
- [7] *DC dot Dublin Core Metadata Editor*. <http://www.ukoln.ac.uk/metadata/dcdot>, zuletzt geprüft am 21.06.2005.
- [8] *DSpace*. <http://www.dspace.org>, zuletzt geprüft am 21.06.2005.
- [9] *Dublin Core Metadata Initiative*. <http://dublincore.org>, zuletzt geprüft am 21.06.2005.
- [10] *EdNA Online - Educational Network Australia*. <http://www.edna.edu.au>, zuletzt geprüft am 21.06.2005.
- [11] *Free Software Foundation*. <http://www.fsf.org>, zuletzt geprüft am 21.06.2005.
- [12] *IEEE Learning Technology Standards Committee*. <http://ltsc.ieee.org>, zuletzt geprüft am 21.06.2005.
- [13] *ILIAS open source*. <http://www.ilias.de>, zuletzt geprüft am 21.06.2005.
- [14] *IMS Global Learning Consortium, Inc.* <http://www.imsglobal.org>, zuletzt geprüft am 21.06.2005.

-
- [15] *k-MED - Knowledge - Based Multimedia Medical Education*. <http://www.k-med.org>, zuletzt geprüft am 21.06.2005.
- [16] *Klarity*. <http://www.intology.com.au/20products/50Klarity>, zuletzt geprüft am 21.06.2005.
- [17] *Macromedia Director*. <http://www.macromedia.com/software/director>, zuletzt geprüft am 21.06.2005.
- [18] *MERLOT - Multimedia Educational Resource for Learning and Online Teaching*. <http://www.merlot.org>, zuletzt geprüft am 21.06.2005.
- [19] *Metabrowser*. <http://metabrowser.spirit.net.au>, zuletzt geprüft am 21.06.2005.
- [20] *MulitBook*. <http://www.multibook.de>, zuletzt geprüft am 21.06.2005.
- [21] *National Institute of Environmental Health Sciences (NIEHS)*. <http://www.niehs.nih.gov>, zuletzt geprüft am 21.06.2005.
- [22] *Oasis DocBook Technical Committee*. http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=docbook, zuletzt geprüft am 21.06.2005.
- [23] *Open Archives Initiative*. <http://www.openarchives.org>, zuletzt geprüft am 21.06.2005.
- [24] *RELOAD (Reusable eLearning Object Authoring & Delivery) Editor*. <http://www.reload.ac.uk/editor.html>, zuletzt geprüft am 21.06.2005.
- [25] *ResourceCenter at SourceForge*. <http://sourceforge.net/projects/resourcecenter>, zuletzt geprüft am 21.06.2005.
- [26] *SMETE Digital Library*. <http://www.smete.org>, zuletzt geprüft am 21.06.2005.
- [27] *The Dissertation Markup Language (DiML)*. <http://edoc.hu-berlin.de/diml>, zuletzt geprüft am 21.06.2005.
- [28] *Toolbook*. <http://www.toolbook.com>, zuletzt geprüft am 21.06.2005.
- [29] Lim Kin Chew: *Implementation Issues; Learning Standards Technical Committee, Metadata Forum*, Dezember 2004. <http://www.itsc.org.sg/events/metadata04/Kin%20Chew.pdf>, zuletzt geprüft am 21.06.2005.

Abkürzungsverzeichnis

ADL	Advanced Distributed Learning Initiative, Inc.
AICC	Aviation Industry CBT Committee
Ariadne	Alliance of Remote Instructional Authoring and Distribution Networks for Europe
CBT	Computer-Based Training
DC	Dublin Core
DITA	Darwin Information Typing Architecture
DSSSL	Document Style Semantics and Specification Language
EML	Education Markup Language
GPL	GNU Public License
HTML	Hyper Text Markup Language
IEEE	Institute of Electrical & Electronics Engineers
IMS	IMS Global Learning Consortium, Inc., bei Gründung in 1997: Instructional Management Systems (IMS) project
IMS CP	IMS Content Packaging
IMS LD	IMS Learning Design
IMS LRM	IMS Learning Resource Metadata
k-MED	Knowledge based medical education
KPS	Knowledge Pool System
L3	Life Long Learning
LO	Learning Object

LOM	Learning Object Metadata
LOR	Learning Object Repository
LTSC	Learning Technology Standards Committee
RIO	Reusable Information Object
RLO	Reusable Learning Object
RTE	SCORM Run Time Environment
SCORM	Sharable Content Object Reference Model
SGML	Standard Generalized Markup Language
SMIL	Synchronized Multimedia Integration Language
WBT	Web Based Training
WYSIWYG	What You See Is What You Get
XML	eXtensible Markup Language
XSL	eXtensible Stylesheet Language

Anhang

A XML-Schema des Datenmodells

In diesem Anhang werden die XML-Schema-Definitionen des Datenmodells abgedruckt, das in Abschnitt 7.3.1 beschrieben wurde.

Es folgt die XML-Schema-Definition für Kurse, die in Abbildung 7.20 visualisiert wurde:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">
  <xs:element name="course">
    <xs:complexType>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element name="lection" minOccurs="0" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="id"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="testitem" minOccurs="0" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="id"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Die XML-Schema-Definition für Lektionen ist aufgrund der Integration von Blöcken und der Möglichkeit, Lektionen rekursiv zu schachteln, etwas aufwändiger, vergleiche auch mit Abbildung 7.22:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">
  <xs:element name="lection">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="content" minOccurs="0">
          <xs:complexType>
            <xs:choice minOccurs="0" maxOccurs="unbounded">
              <xs:element name="block" minOccurs="0"
                maxOccurs="unbounded">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="resource" minOccurs="0"
                      maxOccurs="unbounded">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="id"/>
                          <xs:element name="subtitle"/>
                          <xs:element name="width" minOccurs="0"/>
                          <xs:element name="height" minOccurs="0"/>
                          <xs:element name="parameter" minOccurs="0"
                            maxOccurs="unbounded"/>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                    <xs:element name="text" minOccurs="0">
                      <xs:complexType/>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="pagebreak" minOccurs="0"
                maxOccurs="unbounded"/>
            </xs:choice>
          </xs:complexType>
        </xs:element>
        <xs:element name="sublections" minOccurs="0">
          <xs:complexType>
            <xs:choice minOccurs="0" maxOccurs="unbounded">
```

```
<xs:element name="sublection" minOccurs="0"
  maxOccurs="unbounded">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="id"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="testitem" minOccurs="0"
  maxOccurs="unbounded">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="id"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:choice>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```


B Publikationen des Verfassers

- [HRS05] Stefan Hoermann, Christoph Rensing und Ralf Steinmetz: *Wiederverwendung von Lernressourcen mittels Authoring by Aggregation im ResourceCenter*. In: *Tagungsband der 3. Deutschen e-Learning Fachtagung Informatik - DeLFI*, Seiten 153 – 164, Rostock, September 2005.
- [HHR05] Stefan Hoermann, Tomas Hildebrandt, Christoph Rensing und Ralf Steinmetz: *ResourceCenter - A Digital Learning Object Repository with an Integrated Authoring Tool Set*. In: *Proceedings of EDMEDIA World Conference on Educational Multimedia, Hypermedia & Telecommunications*, Montreal, 2005.
- [HSS05] Stefan Hoermann, Cornelia Seeberg und Ralf Steinmetz: *kmed: Knowledge - Based Multimedia Medical Education*, Kapitel E-Learning Repository - ResourceCenter, Seiten 92 – 97. htcc-Verlag, Februar 2005. ISBN 3-00-015378-0.
- [HFSS04] Stefan Hoermann, Andreas Faatz, Cornelia Seeberg und Ralf Steinmetz: *ResourceCenter: Eine digitale Bibliothek für Lernressourcen mit Open Source Lizenz*. In: M. Fischer H. Handels R. Klar J. Leven F. Puppe S. Pöppel, J. Bernauer und K. Spitzer (Herausgeber): *Rechnergestützte Lehr und Lernsysteme in der Medizin*, Seiten 113–124, März 2004.
- [HJ03] Stefan Hoermann und Ronny John: *Evaluation von XML-Datenbanken für LOM-Datensätze*. Technischer Bericht TR-KOM-2003-08, KOM Multimedia Communications Lab, Mai 2003.
- [HJSS03] Stefan Hoermann, Ronny John, Cornelia Seeberg und Ralf Steinmetz: *Evaluation von Datenbanken zur Speicherung von strukturierten Metadaten am Beispiel LOM*. In: *Proceedings der DeLFI 2003 - 1. Fachtagung „e-Learning“ der Gesellschaft für Informatik*, Seiten 215 – 224, September 2003.

- [HSDK⁺03a] Stefan Hoermann, Cornelia Seeberg, Luka Divac-Krnic, Oliver Merkel, Andreas Faatz und Ralf Steinmetz: *Building Structures of Reusable Educational Content Based on LOM*. In: *Proceedings of The 15th Conference On Advanced Information Systems Engineering workshop on Semantic Web for Web-based Learning*, Juni 2003.
- [HSDK⁺03b] Stefan Hoermann, Cornelia Seeberg, Luka Divac-Krnic, Oliver Merkel, Andreas Faatz und Ralf Steinmetz: *Representation of Knowledge as Support for Authors of Reusable Educational Content*. Poster Presentation at The Twelfth International World Wide Web Conference, Mai 2003.
- [HSGS03] Stefan Hoermann, Stefan Schneider, Ulrich Glowalla und Ralf Steinmetz: *Erstellung von SCORM-kompatiblen Kursen im Projekt k-MED*. In: *Rechnergestützte Lehr- und Lernsysteme in der Medizin*, Seiten 103–113. Shaker Verlag, Aachen, April 2003.
- [HS02b] Stefan Hoermann und Ralf Steinmetz: *Kurse erstellen mit Learning Object Metadata*. Technischer Bericht TR-2002-05, Multimedia Communications Lab - KOM, November 2002.
- [HS02a] Stefan Hoermann und Ralf Steinmetz: *k-MED: Kurse gestalten und adaptieren mit rhetorisch-didaktischen Relationen*. In: *Rechnergestützte Lehr- und Lernsysteme in der Medizin*. J. Bernauer and M. R. Fischer and F. J. Leven and F. Puppe and M. Weber, April 2002.
- [HFM⁺01] Stefan Hoermann, Andreas Faatz, Oliver Merkel, Ansgar Hugo und Ralf Steinmetz: *Ein Kurseditor für modularisierte Lernressourcen auf der Basis von Learning Objects Metadata zur Erstellung von adaptierbaren Kursen*. In: *LLWA 01 - Tagungsband der GI-Workshopwoche „Lernen-Lehren-Wissen-Adaptivität“*, Seiten 315–323. Ralf Klinkenberg, Stefan Rueping, Andreas Fick, Nicola Henze, Christian Herzog, Ralf Molitor, Olaf Schroeder, Oktober 2001. (Research Report #763).
- [RHS05] Christoph Rensing, Stefan Hoermann und Ralf Steinmetz: *Wiederverwendung und Autorenunterstützung im E-Learning auf Basis von Digitalen Repositories*. In: Werner Sesink, Max Mühlhäuser und Ralf Steinmetz (Herausgeber): *thema Forschung: E-Learning*, Seiten 84 – 91, Februar 2005. ISSN 1434-7768.
- [HH05] Jan Hansen und Stefan Hoermann: *eLearning und Datenschutz - auf der Suche nach einem Gleichgewicht*. In: *Tagungsband der GML - 3. Workshop „Grundfragen multimedialen Lehrens und Lernens“*, Cottbus, März 2005.

-
- [DKMF⁺03] Luka Divac-Krnic, Oliver Merkel, Andreas Faatz, Stefan Hoermann, Cornelia Seeberg und Ralf Steinmetz: *Prozess der Lernplattformauswahl in k-Med.* Technischer Bericht TR-KOM-2003-04, KOM Multimedia Communications Lab, Mai 2003.
- [MHS02] Oliver Merkel, Stefan Hoermann und Ralf Steinmetz: *kmed: Ein wissensbasiertes multimediales Lernsystem für die Medizin.* In: *Tagungsband 1: Learntec 2002, 10. Europäischer Kongress und Fachmesse für Bildungs- und Informationstechnologie*, Seiten 443–449. Uwe Beck and Winfried Sommer, Februar 2002.
- [FHSS01] Andreas Faatz, Stefan Hoermann, Cornelia Seeberg und Ralf Steinmetz: *Conceptual Enrichment of Ontologies by means of a generic and configurable approach.* In: *Proceedings of the ESLLI 2001 Workshop on Semantic Knowledge Acquisition and Categorisation*, August 2001.
- [SFS⁺01] Achim Steinacker, Andreas Faatz, Cornelia Seeberg, Ivica Rimac, Stefan Hoermann, Abdulmotaleb El Saddik und Ralf Steinmetz: *MediBook: Combining Semantic Networks with Metadata for Learning Resources to Build a Web Based Learning System.* In: *Proceedings of the EdMedia 2001*, Juni 2001.
- [FSH⁺00] Andreas Faatz, Abdulmotaleb El Saddik, Stefan Hoermann, Ivica Rimac, Cornelia Seeberg, Achim Steinacker und Ralf Steinmetz: *Multimedia und Wissen: Unser Weg zu einem produktiven Umgang mit Wissensdurst.* *thema Forschung*, 2000(2):26–33, November 2000.
- [SRH⁺00] Cornelia Seeberg, Ivica Rimac, Stefan Hoermann, Andreas Faatz, Achim Steinacker und Abdulmotaleb El Saddik und Ralf Steinmetz: *MediBook: Realisierung eines generischen Ansatzes für ein internetbasiertes Multimedia-Lernsystem am Beispiel Medizin.* In: *Tagungsband: Treffen der GI-Fachgruppe 1.1.3 Maschinelles Lernen (GMD Report 114)*, Seiten 96–105, September 2000.

C Lebenslauf des Verfassers

Name: Stefan Hörmann
Geburtstag: 6. Mai 1972
Geburtsort: Hoya an der Weser

Schulbildung

1979 – 1983 Hirschbach-Schule (Grundschule) in Reinheim
1983 – 1985 Justin-Wagner-Schule (Förderstufe) in Roßdorf
1985 – 1992 Viktoria Schule (Gymnasium) in Darmstadt
Abschluss: Abitur 06/1992

Grundwehrdienst

10/1992 – 09/1993 Ausbildung zum Fernmeldemechaniker und spätere Stationierung im 1./ AmphPiBtl. 330 Speyer

Studium

10/1993 – 07/2000 Studium Elektrotechnik, Technische Universität Darmstadt
Abschluss: Diplom-Ingenieur Elektrotechnik und Informationstechnik, 07/2000

Berufliche Tätigkeit

seit 08/2000 Wissenschaftlicher Mitarbeiter im Fachbereich Elektrotechnik und Informationstechnik der Technischen Universität Darmstadt

