POSTPRINT

XML-strukturierte Leaning Objects

Henning Lobin
Maik Stührenberg

Justus-Liebig-Universität Gießen
Arbeitsbereich Angewandte Sprachwissenschaft und Computerlinguistik

Zusammenfassung

Dieser Beitrag skizziert die Möglichkeiten, die die Extensible Markup Language (XML) im Umfeld von eLearning und Web Based Training (WBT) eröffnet. Bisherige eLearning-Angebote kranken an verschiedenen Problemen, die durch die Verwendung von XML-basierten *Learning Objects* vermieden werden können. Ausgehend vom aktuellen Stand im Projekt MiLCA – Medienintensive Lehrmodule in der Computerlinguistik-Ausbildung – soll zudem ein Ausblick auf zukünftige technische Möglichkeiten des Computer-gestützten Lernens gegeben werden.

Inhaltsverzeichnis

1	Einleitung	3
2	eLearning – der aktuelle Stand	4
3	Strukturierung der Daten mit Hilfe von XML	5
3.1	Eine kurze Einführung in XML	5
3.2	Der Lebenszyklus strukturierter Dokumente	8
3.3	Vorteile XML-strukturierter Learning Objects	9
4	Learning Objects in MiLCA	10
4.1	Das Projekt MiLCA	10
4.2	Die MiLCA-DTD	11
4.2.1	Metadaten	11
4.2.2	XML-Implementation	13
4.3	Import in die Lehr-/Lernplattform ILIAS	14
5	Die Zukunft – adaptives Lernen	15
	Literatur	17

1 Einleitung

Nicht erst seit den Ergebnissen der PISA-Studie sind in Industrie und Forschung verstärkte Aktivitäten im Bereich des eLearnings zu beobachten. Hinter dem Begriff eLearning steht die Idee, Inhalte statt durch einen Lehrer aus Fleisch und Blut durch einen Computer (Computer Based Training – CBT) bzw. Internetbrowser (Web Based Training – WBT) zu vermitteln. Hierbei muss zunächst die prinzipielle Unterscheidung getroffen werden, ob die digitalen Medien den menschlichen Lehrenden unterstützen oder ihn ersetzen sollen. CBT und WBT erlauben eine Loslösung des Lernens vom Klassenzimmer: Multimedial aufbereitetes und sequenziertes Lernmaterial, so genannte *Learning Objects*, sollen das Lernen nicht nur ortsunabhängig, sondern auch für den Lernenden durch interaktive Elemente interessanter machen. Experimente sollen nicht mehr im Labor, sondern zu Hause auf dem heimischen Rechner oder auf dem PDA oder Mobiltelefon durchgeführt werden können. Durch Hyperlinks miteinander verknüpfte *Learning Objects* erlauben das einfache Nachschlagen von Begriffen und bieten die permanente Möglichkeit des Wiederholens von Lerninhalten – zu jeder beliebigen Zeit und an jedem beliebigen Ort.

Lehrmanagementsystemen, so genannte Lehr- und Lernplattformen, sollen dem Benutzer die gewünschten Lerninhalte in einer für ihn speziell aufbereiteten Weise präsentieren.

Um diese Vision von eLearning zu verwirklichen, gibt es verschiedene staatliche und industrielle Initiativen. Die einzelnen Ansätze sind allerdings oft untereinander nicht kompatibel, was die Verwendung offener, standardisierter Formate und Systeme sinnvoll erscheinen lässt.

Im vorliegenden Beitrag wird das Thema der Entwicklung von *Learning Objects* im Zusammenhang mit Erkenntnissen aus dem Bereich der Texttechnologie betrachtet. Hintergrund dazu ist ein durch das Bundesministerium für Bildung und Forschung gefördertes Projekt, bei dem es um die Entwicklung einer Lehrplattform für die akademische Computerlinguistik-Ausbildung geht. Im Bereich der Texttechnologie sind in den letzten Jahren Methoden und Techniken entwickelt worden, die es erlauben, textbasierte Inhalte, die auch multimediale Elemente umfassen können, effizient und nachhaltig zu verarbeiten. Mit der texttechnologischen Aufbereitung von *Learning Objects* konvergieren zwei bisher getrennt voneinander erforschte Bereiche mit der Folge, dass Formen der Vermittlung von Inhalten entwickelt werden können, die unabhängig von spezifischen medialen und technischen Ausprägungen in der Lage sind, den menschlichen Lerner deutlicher als bisher in den Mittelpunkt zu stellen.

2 eLearning – der aktuelle Stand

Betrachtet man die sich zur Zeit im Einsatz befindenden eLearning-Systeme wie WebCT¹, Uni-Online² oder auch die Open Source Lösung ILIAS³, so kann festgehalten werden, dass die Datenstrukturen dieser Produkte untereinander inkompatibel sind und sich alle als eine Kombination aus einer HTML-basierten Oberfläche und einer zu Grunde liegenden Datenbank darstellen.

HTML ist als Format für inhärent gut strukturierbare Informationen wie eLearning-Inhalte nur schlecht geeignet, da es kaum Möglichkeiten zur semantischen Auszeichnung und nur wenig strukturell gliedernde Elemente bietet. Obwohl es die vorherrschende Markupsprache zur Anzeige textueller Inhalte ist, verwenden nahezu alle eLearning-Systeme zusätzlich proprietäre Dateitypen wie Macromedia Flash oder Shockwave, um animierte oder interaktive Inhalte darzustellen. Das Problem bei diesen Formaten ist offensichtlich: Mit der Wahl einer Software bindet man sich an dieses Produkt und liefert sich in Bezug auf zukünftige Entwicklungen der Willkür des Herstellers aus. Das gilt nicht nur für die Unterstützung weiterer Standards, sondern auch und vor allem für die konzeptionelle Weiterentwicklung der Lehrplattform. Selbst der Umstieg auf eine andere Lehrplattform kann durch die Verwendung proprietärer Formate nahezu unmöglich gemacht oder nur mit großem Aufwand vollzogen werden.

Offene Formate und Open Source-Lösungen dagegen erreichen sehr schnell eine große Verbreitung und sind für eine Vielzahl von Systemen verfügbar. Der Fortbestand solcher Softwarelösungen ist durch die offenen Quelltexte in sehr viel stärkerem Maße gesichert als bei Produkten einzelner Unternehmen. Selbst wenn die Entwicklung einer Software eingestellt würde, erlaubte die selbständige Bearbeitung der Quelltexte eine Weiterführung und den Ausbau des eigenen eLearning-Angebots. Die Erweiterung der Metadaten zum Zwecke der Beschreibung adaptiver *Learning Objects* wird als nächster logischer Schritt angesehen (vgl. [1] und [8]).

Aber auch auf der Lernerseite ergeben sich Nachteile durch die Benutzung proprietärer Systeme. Anwender, die aus unterschiedlichen Gründen die entsprechenden PlugIns für den multimedialen Content nicht installiert haben (oder für deren Systeme eine solche Erweiterung nicht verfügbar ist), werden von der Nutzung ausgeschlossen.

Allgemein kann festgehalten werden, dass die Verwendung proprietärer Systeme sowohl die Wiederverwendung als auch die Modularisierung von *Learning Objects* erschwert. Ein Mittel zur Vermeidung der geschilderten Probleme ist die Verwendung XML-strukturierter Daten (vgl. [7] bzgl. der Wiederverwendung und [10]).

¹ WebCT Homepage – <http://www.webct.com>

² Uni-Online Homepage – <http://uni-open-platform.fernuni-hagen.de>

³ ILIAS Homepage – <http://www.ilias.uni-koeln.de/ios>

3 Strukturierung der Daten mit Hilfe von XML

3.1 Eine kurze Einführung in XML

Ein zentrales Mittel bei der nachhaltigen Nutzung textbasierter Lehrmaterialien bildet die strukturelle Textauszeichnung. Unter Textauszeichnung wird die Einfügung von Markierungen in einen Text verstanden, durch die einzelne Textteile in spezieller Weise verarbeitet werden können (vgl. [9]). Derartige Markierungen, auch *Annotation* oder *Markup* genannt, setzen sich aus einem *Anfangs*- und gegebenenfalls einem *Endtag* zusammen. Insbesondere zu Darstellungszwecken werden schon seit den sechziger Jahren Textauszeichnungsverfahren angewendet. Aber erst mit der *Standard Generalized Markup Language* (SGML), einer standardisierten Metasprache zur Spezifikation von Textauszeichnungsvokabularien, stand ab 1986 ein Instrumentarium zur Verfügung, durch das die Textauszeichnung von bestimmten Anwendungssystemen unabhängig wurde und zugleich für unterschiedliche Anforderungen spezifisch zugeschnitten werden konnte. Die bekannteste Anwendung von SGML wurde in den neunziger Jahren HTML, die „Sprache des World Wide Web“. Eine vereinfachte Version von SGML, die *Extensible Markup Language* (XML), hat heute den ursprünglichen Standard weitgehend ersetzt.

Eine der grundlegenden Neuerungen im Bereich des Dokument-Markup, die durch SGML eingeführt wurde und auch die Grundlage von XML bildet, besteht darin, die durch Tags annotierten Textteile in einen hierarchischen Zusammenhang zu bringen, anstatt sie lediglich linear anzuordnen (vgl. [5]). Die Textteile befinden sich dadurch nicht nur in der natürlich immer noch vorhandenen sequenziellen Reihenfolge, sondern stehen zusätzlich auch noch in einer Teil-Ganzes-Beziehung zueinander. Ein Textelement „Kapitel“ kann danach z. B. aus den Textelementen „Überschrift“, drei Absatz- und einem Abbildungselement bestehen. Das Kapitelelement seinerseits konstituiert mit anderen Elementen ein Textelement „Buch“.

Informationselemente, die in dieser Weise verschachtelt sind, lassen sich grafisch und formal als Bäume darstellen. Die Interpretation von Textdokumenten als Bäume hat einige große Vorteile gegenüber einer linearen Textauszeichnung: Wenn in einem Baum einem Element eine Eigenschaft zugeordnet ist, so kann z. B. festgelegt werden, dass diese Eigenschaft auch allen diesem Element untergeordneten Elementen zukommt, sofern bei ihnen nicht explizit etwas anderes vermerkt ist. Dieses Prinzip der *Vererbung* von Eigenschaften ist also ein Mittel, Teilen eines Dokuments effizient Merkmale zuzuordnen. Weiterhin erlaubt die Baumstruktur eine recht präzise Navigation im Dokument. Ein Ausdruck wie „der zweite Absatz im nachfolgenden Kapitel“ kann leicht als ein Pfad durch den Dokumentbaum dargestellt werden.

Aus Sicht der maschinellen Verarbeitung von annotierten Dokumenten können wegen der Universalität baumartig strukturierter Information in der Informationstechnologie viele Verfahren aus anderen Bereichen übernommen werden. Bei der Überprüfung beispielsweise, ob ein XML-Dokument eine korrekt aufgebaute Baumstruktur aufweist, können Verfahren angewendet werden, die für die Analyse von Computerprogrammen entwickelt worden sind (Compilerbau). Aus Sicht der Linguistik ist jedoch

der wichtigste Vorteil der Verwendung von Baumstrukturen, dass man diese durch Grammatiken beschreiben kann und Grammatiktheorie und Grammatikformalismen in der Linguistik als gut untersuchte Gebiete gelten können. In XML ist nämlich nicht nur kodifiziert, wie die Baumstruktur eines Dokuments formal zu notieren ist, es beinhaltet auch eine Sprache zur Definition von Dokumentgrammatiken. Sehen wir uns an, was für Arten von Regeln in einer XML-Dokumentgrammatik verwendet werden.

Bei der Textstrukturierung können wir zwei verschiedene Ebenen unterscheiden, auf denen Informationseinheiten erscheinen können:

1. die Ebene der konkreten textuellen Daten
2. die Ebene der abstrakten Einheiten, die Daten Funktionen zuordnen oder gruppieren.

Auf der Ebene der Elemente können entsprechend zwei verschiedene Typen unterschieden werden:

1. *Daten-Elemente*: Elemente dieses Typs enthalten unmittelbar die konkreten Daten;
2. *Container-Elemente*: Elemente dieses Typs enthalten selbst wiederum Elemente, wobei die enthaltenen Elemente Daten- oder Container-Elemente sein können.

Daten- und Container-Elemente können auch als Mischform auftreten, jedoch wird gewöhnlich empfohlen, bei der Definition von Elementen eine eindeutige Zuordnung zu einer der beiden Gruppen vorzusehen.

Für Daten-Elemente muss selbstverständlich klar sein, um was für eine Art von Daten es sich handelt. Soweit nichts anderes vermerkt ist, wird davon ausgegangen, dass Daten-Elemente Zeichenfolgen enthalten, die aus den dafür zulässigen Zeichen gebildet sind. In XML sieht die Deklaration eines Daten-Elements folgendermaßen aus:

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

Diese Definition besagt, dass ein Element vom Typ `autor` Daten vom Typ `PCDATA` enthalten kann. Das Zeichen `#` signalisiert dabei, dass es sich bei `PCDATA` um ein vordefiniertes Schlüsselwort handelt. Elemente erhalten also einen Namen, durch den es möglich wird, die Information, die in ihnen enthalten ist, zu klassifizieren und zu beschreiben.

Container-Elemente enthalten im Gegensatz zu Daten-Elementen nicht unmittelbar Daten, sondern organisieren Elemente:

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

Der Elementtyp `buch` ist hier als die Abfolge von vier Daten-Elementen definiert. Die Festlegung, in welcher Weise ein Container-Element im Einzelnen andere Elemente organisiert, geschieht durch das Inhaltsmodell, dem rund geklammerten Teil

innerhalb der Element-Deklaration. Im Inhaltsmodell werden zum einen Angaben darüber gemacht, welchen Status die einzelnen Elementtypen haben, zum anderen werden die Elemente zueinander in Beziehung gesetzt. Hier weisen alle Elemente den Status „obligatorisch“ auf, da ihrem Namen kein weiteres Zeichen folgt. Insgesamt gibt es vier verschiedene Status von Elementen:

- **a**: das Element muss genau einmal auftreten (obligatorisch)
- **a?**: das Element kann einmal auftreten, kann aber auch ausgelassen werden (fakultativ)
- **a+**: das Element muss mindestens einmal, kann aber beliebig oft auftreten
- **a***: das Element kann einmal oder beliebig oft auftreten, kann aber auch ausgelassen werden

Wir können beispielsweise die folgenden Statusänderungen vornehmen:

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

Durch diese Deklaration ist es nun möglich, mehrere Autoren für ein Buch anzugeben und die Verlags- und Jahresangaben fortzulassen.

Für die Festlegung der Beziehungen der Elemente zueinander stehen zwei Konnektoren zur Verfügung:

- **a, b**: b folgt auf a
- **a | b**: entweder a oder b

Die hierarchische Anordnung der Elemente kann als das wichtigste Grundprinzip von XML verstanden werden.

In einer Dokumentgrammatik gibt es die Möglichkeit, bestimmte Arten von Informationen als *Attribute* Elementen zuzuordnen. Der Vorteil besteht darin, für solche Attribute die erlaubten Werte angeben zu können:

```
<!ATTLIST buch
  typ (monografie | sammelband | festschrift) "monografie"
  preis NMTOKEN #REQUIRED>
```

Bei dieser Deklaration wird für das Attribut **typ** eine Liste mit drei möglichen Werten angegeben und als Standardbelegung der Wert **monografie** vorgesehen. Das Attribut **preis** kann Ziffern- und Buchstabenfolgen enthalten, es ist als vom Benutzer obligatorisch mit einem Wert zu belegen spezifiziert.

Neben der Deklaration von Elementtypen und Attributen werden in einer Dokumentgrammatik auch einige andere Festlegungen getroffen, die die Struktur der Dokumente betreffen. Die Menge aller dieser Deklarationen bildet die *Dokumenttyp-Definition* (DTD), die im XML-Standard kodifizierte Version einer Dokumentgrammatik.

3.2 Der Lebenszyklus strukturierter Dokumente

Man kann die Verwendung strukturierter Dokumente als das Durchlaufen eines Lebenszyklus konzeptualisieren. Danach folgen auf die Phasen der Strukturierung und der Datenerfassung die der Bearbeitung und Konvertierung, bevor sich der Zyklus mit der Phase der Revisionierung schließt. In allen diesen Phasen spielt die DTD eine zentrale Rolle bei der Automatisierung der beteiligten Verarbeitungsprozesse. Die Datenerfassung geschieht z. B. in XML-Editoren, die die DTD nutzen, um den Benutzer bei der Datenerstellung zu leiten. Bei der Bearbeitung der Dokumente können durch die Festlegung aller möglichen Dokumentstrukturen in der DTD sehr komplexe Transformationen generisch definiert werden.

In der Phase der *Strukturierung* müssen die Daten analysiert und eine Dokumentgrammatik als DTD spezifiziert werden. Dieser Vorgang kann mit dem Programmieren verglichen werden, da ein ähnlicher Unterzyklus (Spezifikation, Implementation, Testen, Modifikation) zu durchlaufen ist wie beim *Software Engineering*.

Die *Datenerfassung* kann entweder die durch die DTD unterstützte Eingabe neuer Daten sein oder die teil- oder vollautomatische Konvertierung von Altdaten ins XML-Format.

In der Phase der *Bearbeitung* werden aus dem textuellen Datenbestand verschiedene Textversionen abgeleitet. Der Textbestand kann beispielsweise mehrere Sprachversionen in sich vereinen, möglicherweise auf der Ebene der kleinsten Texteinheiten parallelisiert. Die Aufgabe eines automatischen Verarbeitungsprozesses ist es dann, die verschiedenen einzelsprachlichen Versionen aus dem Textbestand herauszufiltern und dabei ggfs. noch weitere notwendige Umstellungs- oder Auswertungsprozesse durchzuführen.

In der Phase des *Viewing* werden die XML-Strukturen über sog. *Style Sheets* mit Darstellungsinformationen kombiniert, um sie in geeigneten Browsern anzeigen zu können. Zur Definition eines Views auf die Daten gehört auch die Bereitstellung zusätzlicher Metadatenbestände, die für die Navigation in den Textbeständen durch den Benutzer herangezogen werden. Derartige Navigatoren können automatisch erzeugt werden, sofern für die in Textbeständen vorkommenden Struktureinheiten (Überschriften, Warnhinweise usw.) ihre Verwendung für Navigationszwecke allgemein festgelegt wird.

Die *Konvertierung* der Textbestände in andere Zielformate ähnelt der Festlegung von Style Sheets zu Zwecken des Viewings. Der Unterschied besteht darin, dass die XML-strukturierten Textbestände in andere Auszeichnungsformate unwiderruflich überführt werden, um von dort aus mit anderen Verfahren weiterbearbeitet zu werden.

In der Phase der *Revisionierung* werden die in den Test- und Anwendungsläufen der DTD, der Bearbeitungs- und Viewing-Subsysteme gewonnenen Erfahrungen evaluiert, mit der Spezifikation neuer Anforderungen verbunden und bilden dann den Ausgangspunkt für einen neuen Lauf durch den Lebenszyklus.

3.3 Vorteile XML-strukturierter Learning Objects

Es ist bisher deutlich geworden, dass jede Element-Deklaration einer Regel entspricht, die angibt, aus welchen Informationseinheiten ein Element zusammengesetzt sein soll. Durch Element-Deklarationen werden also nicht Informationen direkt ausgedrückt, sondern sie bilden strukturelle Informationen. Der Sinn der Textauszeichnung kann darin bestehen,

- einen Textabschnitt eindeutig identifizierbar zu machen (z. B. eine Definition),
- eine Zuordnung zu einer Gruppe gleich zu behandelnder Textabschnitte vorzunehmen (z.B. Überschriften) oder
- weitere Informationen, sog. Metadaten, zur Verfügung zu stellen (z.B. Bearbeiter und Datum der letzten Änderung eines Textabschnitts).

Dementsprechend können sehr unterschiedliche Verarbeitungsverfahren von der Annotation Gebrauch machen:

- In einem Hypertext-System kann durch Links auf eindeutige Textstellen verwiesen werden – die Textauszeichnung wird also für Browsing-Zwecke verwendet;
- eine Gruppe von gleich ausgezeichneten Textabschnitten kann grafisch gleich behandelt werden – die Textauszeichnung bildet dabei die Grundlage für die visuelle Textgestaltung;
- Metadaten können genutzt werden, um in einem Textkorpus Textabschnitte mit bestimmten Eigenschaften aufzufinden – die Textauszeichnung steht hier im Dienste des *Information Retrieval*.

Im Falle eines *Learning Object* kann man demzufolge sagen, dass Regeln dessen konstitutive Teile so miteinander in Verbindung setzen, dass die Bedeutung des *Learning Object* transferierbar und rekonstruierbar wird.

Was kann man nun aber unter seiner Bedeutung verstehen? In die Entwicklung von Lehrmaterialien fließen Überlegungen zum didaktischen Aufbau, zur Begrifflichkeit, zur Verknüpfung mit anderen Lehrmodulen und zum erwarteten Vorwissen des Lernalers ein. In gut strukturierten *Learning Objects* sollten alle diese Überlegungen in expliziter Form aus der Annotation deutlich werden. Die DTD, die die möglichen Strukturen eines *Learning Object* beschreibt, beinhaltet somit abstraktes Strukturwissen zur Didaktik von Lehrmodulen, die Annotation eines *Learning Object* dessen konkrete Umsetzung.

In den einzelnen Phasen des Lebenszyklus von *Learning Objects* erweist sich die Verfügbarkeit expliziter didaktischer Annotation in verschiedener Weise als nützlich:

- *Produktion*: Während der inkrementellen Produktion eines *Learning Object* verfolgen Editoren den Aufbau des entstehenden Dokuments und gleichen ihn mit der Dokumentgrammatik ab. Es kann angegeben werden, welche Elemente an

einer bestimmten Stelle strukturell eingefügt werden können, und es können falsche Elementeneinfügungen von vornherein verhindert werden. Die didaktische Struktur und die geforderten Metadaten zur Nutzung und zu den Voraussetzungen des *Learning Object* werden also von vornherein in Hinsicht auf ihre Vollständigkeit überwacht.

- *Validierung*: Ein vorliegendes *Learning Object* wird dahingehend überprüft, ob seine Baumstruktur sich von den in der Grammatik enthaltenen Regeln ableiten lässt, die Baumstruktur des Dokuments also der Grammatik entsprechend aufgebaut ist. Nur wenn dieses sichergestellt ist, kann das einzelne *Learning Object* in ein größeres Geflecht von *Learning Objects* eingegliedert und in flexibler Weise im Rahmen eines Lehrsystems genutzt werden.
- *Retrieval*: Das Auffinden von Informationen in *Learning Object* kann durch die Dokumentgrammatiken unterstützt werden. Soll z.B. ein Textelement **Test** mit einem bestimmten Stichwort gesucht werden, kann die Suche auf solche Abschnitte des *Learning Object* beschränkt werden, in denen ein Element **Test** überhaupt nur vorkommen kann.

Weitere Verwendungsideen der Annotation XML-strukturierter *Learning Object* reichen über den gegenwärtigen Stand in bereits eingesetzten Lehrsystemen noch hinaus. Tendenzen einer mittelfristig zu erwartenden Ausweitung der Nutzung von *Learning Objects* werden in Abschnitt 5 diskutiert.

4 Learning Objects in MiLCA

4.1 Das Projekt MiLCA

Das Projekt „MiLCA – Medienintensive Lerninhalte in der Computerlinguistik-Ausbildung“ – ist ein vom BMBF gefördertes Verbundprojekt im Rahmen des Programms „Neue Medien in der Bildung“. Die fünf beteiligten Projektpartner (die Universitäten Bonn, Gießen, Osnabrück, Saarbrücken und Tübingen) haben sich folgende Ziele gesetzt:

- Erstellung von Lehrmodulen für Studiengänge der Computerlinguistik auf nationaler und internationaler Ebene;
- Erprobung der Lehrmodule in der Lehre am Standort des Produzenten, an den Standorten der Projektpartner und schließlich bei akademischen und industriellen Partnern außerhalb des Konsortiums;
- Integration aller Lehrmodule in eine Lernumgebung;
- Gründung einer Verwertungsgesellschaft für die Pflege und Verbreitung der Produkte über den Förderzeitraum hinaus;

- Erprobung und Implementierung von Metadaten-Standards für die Beschreibung von Lernobjekten.

Dabei sind die einzelnen Universitäten jeweils für eigenständige, in sich abgeschlossene Module und Lerninhalte zuständig. Zur Gliederung und Strukturierung der Inhalte wurde an der Universität Gießen eine XML-DTD entwickelt, die es zusätzlich ermöglicht, einzelne Aspekte des Erstellungsprozesses zu automatisieren und ein einheitliches Erscheinungsbild der Module sicher zu stellen.

4.2 Die MiLCA-DTD

Es existieren verschiedene Ansätze, Lerninhalte mit Hilfe von XML zu strukturieren. Ein viel versprechendes Beispiel dafür ist die komplexe *Educational Modelling Language* EML [6], die sich noch in der Entwicklung befindet.

Ein Großteil der bisherigen Arbeiten beschränkt sich darauf, Sets von Metadaten, also Informationen über den eigentlichen Inhalt, zu definieren und diese dann in XML zu implementieren. Ein prominentes Beispiel hierfür ist die „IMS Learning Resource Meta-data Specification“ [4].

Da die Verwendung von Lehrmodulen in hohem Maße abhängig ist von der Annotation der Inhalte mit Metadaten, war auch bei der Entwicklung einer eigenständigen DTD für MiLCA der erste Schritt die Sichtung der verschiedenen Metadaten-Standards.

Der Metadatensatz der *Learning Object Metadata Working Group* [3] des IEEE *Learning Technology Standards Committee (LTSC)*⁴ gilt als eine der wichtigsten Initiativen zu diesem Thema: *Learning Object Metadata (LOM)* ist seit Juni 2002 offizieller IEEE-Standard⁵. Die verschiedenen Working Drafts, die während der mehrjährigen Entwicklung frei einsehbar waren, werden in mehreren Initiativen eingesetzt. So bauen etwa die Implementierungen des IMS *Global Learning Consortium*⁶ auf LOM auf. Aus diesem Grund enthält auch die MiLCA-DTD den vollständigen Metadatensatz von LOM (in der Version WD 6.1). Diese Kompatibilität sichert die Nachhaltigkeit der mit der DTD annotierten Inhalte.

4.2.1 Metadaten

Der *Learning Object Metadata*-Standard (LOM) definiert neun Kategorien von Metadaten, die sich ihrerseits aus mehreren Einträgen zusammensetzen⁷:

1. *General*: Beinhaltet allgemeine Informationen über das *Learning Object* wie Titel, verwendete Sprachen, Beschreibung und Schlüsselwörter. Das wichtigste

⁴ Homepage des IEEE *Learning Technology Standards Committee* – <http://ltsc.ieee.org>

⁵ Das Institute of Electrical and Electronics Engineers (IEEE) ist eine weltweit tätige Organisation, die in allen technischen Bereichen tätig ist. Zu den vorrangigen Zielen gehört vor allem die Mitarbeit und Gestaltung internationaler Standards

⁶ IMS Global Learning Consortium – <http://www.imspj.org>

⁷ für nähere Angaben zu den Kategorien vgl. [3]

Attribut in dieser Kategorie ist der *Aggregation Level*, der Aussagen über die Komplexität eines *Learning Objects* macht.

Ein *Aggregation Level* von 1 beschreibt ein *Learning Object*, das nicht weiter unterteilbar (atomar) ist, zum Beispiel ein Bild.

Learning Objects des *Aggregation Level* 2 sind aufgebaut wie eine Textseite (mit evtl. eingebetteten Bildern).

Aggregation Level 3 bezeichnet eine Ansammlung von Level 2 *Learning Objects*, die einen Kurs repräsentieren, zum Beispiel in Form einer Übersichtsseite mit daran angeschlossenen Inhaltsseiten.

Ein *Learning Object* des *Aggregation Level* 4 schließlich ist ein Set von Kursen, das dem Erreichen eines Abschlusses dient.

2. *LifeCycle*: Enthält Angaben über die in Abschnitt 3.2 angesprochenen Lebenszyklusphasen sowie Mitwirkende.
3. *Meta-Metadata*: Hier werden Informationen über die Metadaten eingetragen, da Metadaten durchaus auch von anderen Personen als den Autoren des Lehrmoduls erstellt worden sein können.
4. *Technical*: Enthält technische Angaben wie das Format und die Größe des *Learning Objects*, aber auch Installations- und Bedienungsvoraussetzungen, die für die Nutzung erfüllt sein müssen.
5. *Educational*: Hier können Angaben über das empfohlene Alter des Lernenden, den Grad der Interaktivität und die semantische Dichte sowie den Schwierigkeitsgrad eingetragen werden.
6. *Rights*: Beinhaltet Informationen zu Urheberrechten und evtl. anfallenden Kosten.
7. *Relation*: Mit Hilfe der hier gespeicherten Informationen können Beziehungen zwischen einzelnen *Learning Objects* hergestellt werden. Dazu stehen verschiedene Arten von Relationen zur Verfügung, wie u. a. *is part of*, *is basis for* oder *is referenced by*.
8. *Annotation*: Enthält Anmerkungen zum *Learning Object*.
9. *Classification*: Hier kann das *Learning Object* mit Hilfe eines Taxonomie-Pfades und Schlüsselbegriffen im Gesamtkontext des jeweiligen Fachgebietes eingeordnet werden.

Mit Hilfe dieser Metadaten können *Learning Objects* in vielfacher Hinsicht beschrieben werden. Die in den Metadaten gespeicherten Informationen erlauben es nicht nur dem menschlichen Lehrenden, Inhalte nach bestimmten Kriterien zu selektieren, sie erleichtern auch eLearning-Plattformen die automatische Auswahl von *Learning Objects*.

4.2.2 XML-Implementation

Die DTD setzt sich wie folgt zusammen: Die Dokumentwurzel ist das Element `LearningObject`. Ein `LearningObject` enthält die folgenden Tochterelemente:

- `MetaData`
- `Content`
- `Test`
- `Bibliography`
- `Glossary`

Abgesehen von `MetaData` sind alle Tochterelemente optional, das heißt, auch ein `LearningObject`, das nur aus `MetaData` besteht, ist gültig (valide). Damit können auch nicht-textuelle oder sogar nicht-digitale *Learning Objects* beschrieben werden.

```
<!ELEMENT LearningObject (MetaData, Content?, Test?,  
                           Bibliography?, Glossary?)>
```

Auf den Inhalt des Elements `MetaData` soll an dieser Stelle nicht eingegangen werden, es enthält die Metadatenkategorien und -elemente, die in Abschnitt 4.2.1 dargestellt wurden. Als Grundregel für die Umsetzung galt dabei, dass alle Metadaten, die einen Wert aus einem vom Standard vorgegebenen Vokabular annehmen können, als Attribute realisiert sind.

Das Element `Content` nimmt entweder `Text` oder ein weiteres `LearningObject` auf:

```
<!ELEMENT Content (Text | LearningObject)+>
```

Durch den rekursiven Aufbau und die damit mögliche Verschachtelung können `LearningObjects` bereits auf dieser Ebene strukturiert werden. Eine weitere Möglichkeit der Strukturierung ist durch die Metadatenfelder `Relation` und `Classification` möglich. Diese Form der Sequenzierung auf niedriger Ebene erleichtert die Entwicklung eines adaptiven Lernmodells (vgl. auch [2]).

Das Element `Text` enthält nicht-strukturierende Überschriften (`Headline`) und Absätze (`Paragraph`). Es muss mindestens ein Absatz enthalten sein, Überschriften sind optional und können nur vor Absätzen stehen.

Innerhalb eines Absatzes sind verschiedene logische Textauszeichnungen möglich: Betonungen, Kommentare, Hervorhebungen, Zitate und Referenzen, Beispiele, Formeln, Listen, (lokale) Referenzen und (externe) Links, Tabellen, Glossareinträge und Übungsfragen. Dabei wird von den XML-eigenen Möglichkeiten der Referenzierung Gebrauch gemacht, um Beziehungen zwischen einzelnen Worten, Textteilen oder Literatureinträgen oder *Learning Objects* herzustellen.

Im optionalen Element `Test` können Wissensfragen an den Lernenden gerichtet werden. Bislang erschöpft sich dieses Element auf einfache Multiple-Choice-Aufgaben.

Eine Erweiterung unter Berücksichtigung der IMS *Question & Test Interoperability Specification*⁸ ist geplant. Durch die Möglichkeit der rekursiven Verschachtelung ist es möglich, Tests nur an ein eng abgegrenztes Themengebiet (in Form eines `LearningObject`s mit dem *Aggregation Level* 2) oder an eine vollständige Lerneinheit (in Form eines `LearningObject`s mit dem *Aggregation Level* 3) zu knüpfen.

Das Element `Bibliography` enthält Literaturangaben der Quellen, die in einem `LearningObject` verwendet wurden. Die Strukturierung dieses Elements wurde in großen Teilen von dem Projekt „BibTeXXML“ der ETH Zürich übernommen. Ziel dieses Projekts war die automatisierte Überführung von Literaturangaben, die im weit verbreiteten BibTeX Format vorliegen, in ein XML-Format. Da das Projekt nicht mehr fortgeführt wird, werden auch XSLT-Stylesheets aus dem Projekt „The BibTeX-XML-HTML Bibliography Project“⁹ an der Universität Tübingen genutzt. Die Verwendung der BibTeX-kompatiblen DTD und der XSLT-Stylesheets ermöglicht eine einfache Weiternutzung sowohl der in MiLCA verwendeten als auch schon bestehender Literaturangaben.

Für `Bibliography` gilt das schon für das Element `Test` Gesagte: es kann als globale Angabe oder themenspezifisch verwendet werden.

Das Glossar enthält einzelne Einträge (`GlossaryItem`), denen entweder eine Liste von Definitionen oder eine einzelne Definition zugeordnet werden kann. Da die Einträge per XML-Referenzen mit den Definitionen verbunden sind, können keine Glossareinträge ohne erklärende Definition vorkommen. Auch im Fließtext als Glossareintrag markierte Wörter müssen zunächst im Glossar mit einer Definition eingetragen werden. Glossardefinitionen können auch mehreren Glossareinträgen zugeordnet werden und sind nicht an einen festen Eintrag gebunden.

Zum aktuellen Zeitpunkt werden nur noch minimale Wartungsarbeiten an der DTD vorgenommen, da sie bei allen Projektpartnern bereits im produktiven Einsatz ist. Wesentliche Aufgaben für die zukünftige Weiterentwicklung sind die Anpassung an die kürzlich veröffentlichte finale Fassung des LOM-Metadatenstandards und die Erweiterung in Bezug auf didaktische Konzepte, wie in Abschnitt 5 beschrieben.

4.3 Import in die Lehr-/Lernplattform ILIAS

Die *Learning Objects* werden von den Autoren mit Hilfe eines beliebigen XML-Editors erstellt. Dazu steht eine große Auswahl an mächtigen Editoren sowohl im kommerziellen als auch im Open Source Bereich für nahezu alle Betriebssysteme bereit. Nachdem ein *Learning Object* erstellt und mit Hilfe eines XML-Parsers auf seine Gültigkeit (Validität) geprüft worden ist, folgt die Verarbeitung durch XSLT-Stylesheets. Diese erlauben eine automatische Überführung der Inhalte nach XHTML (mit eingebetteten MathML zur Darstellung von Formeln und anderen mathematischen Inhalten) für die Online-Darstellung und XSLFO mit anschließender Wandlung nach PDF als Format zum Offline-Lesen.

⁸ IMS Question & Test Interoperability Specification – <http://www.imsproject.org/question>

⁹ BibTeX-XML-HTML-Bibliography Project Homepage – <http://www.sfs.nphil.uni-tuebingen.de/~auth/xml2html>

Der Import in die im Projekt genutzte Open Source Lernplattform ILIAS erfolgt zur Zeit noch manuell, d.h. die XHTML Ausgabe muss noch von Hand in ILIAS-Lerneinheiten eingebettet und eventuell vorhandene lokale Referenzen angepasst werden. In einer der nächsten ILIAS-Versionen wird der automatische Im- und Export von *Learning Objects* möglich sein, da eine zusammen mit der ILIAS-Entwicklergruppe erarbeitete DTD in weiten Teilen auf der in MiLCA eingesetzten beruht. Geplant ist zunächst ein Import der XML-Datei mit anschließender Server-seitiger Transformation in die beiden Ausgabeformate XHTML und PDF. Diese Konvertierung muss nur bei Änderungen an der XML-Ausgangsdatei erneut erfolgen. Auf die gleiche Weise soll der Export von *Learning Objects* ermöglicht werden, um Lerninhalte in andere ILIAS-Installationen oder andere XML-taugliche Lernplattformen zu überführen.

5 Die Zukunft – adaptives Lernen

Der hier dargestellte Stand von eLearning-Systemen wird schon in naher Zukunft überholt sein. Aktuelle Forschungen im stetig an Bedeutung wachsenden Bereich der digital vermittelten Bildung und Weiterbildung zielen darauf ab, dass sich die virtuellen Lernumgebungen mehr und mehr den individuellen Bedürfnissen und Lernfortschritten ihrer Nutzer anpassen werden. Das Ziel, dem diese Bemühungen weltweit – aber vor allem in Europa – gelten, heißt adaptives Lernen.

Schon jetzt existieren Ansätze, didaktische Modelle und Konzepte in die Metadaten von *Learning Objects* zu integrieren¹⁰. Eine entsprechend angepasste eLearning-Plattform wäre mit einer Auswahl an *Learning Objects* eines beliebigen Inhalts in der Lage, je nach Vorliebe des Benutzers (oder Vorgabe des Lehrenden) den Unterrichtsgegenstand nach bestimmten didaktischen Ansätzen zu präsentieren. Dadurch könnten bei einzelnen didaktischen Methoden schwächere Lernende gezielt trainiert werden, neben der Vermittlung des Inhalts kann die Methode als gleichberechtigtes Lernziel in den Vordergrund treten.

Aber auch Adaptivität in einem zweiten Sinne wird die eLearning-Systeme der Zukunft auszeichnen: Je nach Kenntnisstand des Lernenden wird das System automatisch einen Lernpfad durch die *Learning Objects* wählen. Einsteiger können damit behutsam an ein Thema herangeführt werden, während fortgeschrittene Lerner automatisch Abschnitte überspringen, deren Inhalt sie schon kennen. Dadurch wird die Lernmotivation durchgängig auf einem hohen Niveau gehalten und jeder Einzelne kann gezielt gefordert und gefördert werden.

Ein dritter Punkt besteht schließlich darin, die Materialien selbst in unterschiedlicher medialer und präsentationeller Form verfügbar zu machen. XML-strukturierte *Learning Objects* verhalten sich neutral gegenüber ihrer medialen Aufbereitung. Somit ist es denkbar, aus ein und demselben Pool von *Learning Objects* nicht nur Web-basierte eLearning-Systeme zu unterstützen, sondern parallel dazu auch Print-

¹⁰ Die Educational Modelling Language (EML) bietet ein didaktisches Rollenmodell an. Darüber hinaus wird im Rahmen des Deutschen Instituts für Normung an entsprechenden Metadaten-Standards gearbeitet.

Versionen herzustellen oder für andere digitale Endgeräte optimierte elektronische Fassungen zu generieren. Zusammen mit den Überlegungen zur Benutzeradaptivität der Inhaltspräsentation lässt sich auch die Produktion gedruckter Lehrmaterialien realisieren, die auf die Bedürfnisse eines bestimmten Benutzers angepasst sind („Print-on-Demand“).

Am Ende der Entwicklung werden intelligente eLearning-Systeme stehen, die einen Benutzer unter Berücksichtigung seiner individuellen Lernfortschritte, aber auch seiner Lernfrequenz und persönlichen Gedächtnis- und Verständnisleistungen mit maßgeschneiderten Lehrmaterialien in unterschiedlicher medialer Ausprägung versorgen, die erzielten Lernfortschritte kontrollieren und in die Berechnung der nächsten Lernschritte einfließen lassen. Grundlage dafür bilden stark mit Metadaten angereicherte, systematisch didaktisch strukturierte *Learning Objects*, die system- und medienunabhängig zugänglich sind.

Literatur

- [1] Owen Conlan, Cord Hockemeyer, Paul Lefrere, Vincent Wade, and Dietrich Albert. Extending Educational Metadata Schemas to describe Adaptive Learning Resources. In Hugh Davies, Yellowlees Douglas, and David G. Durand, editors, *Hypertext '01: Proceedings of the twelfth ACM Conference on Hypertext and Hypermedia*, 2001.
- [2] Stephan Fischer. Course and Exercise Sequencing Using Metadata in Adaptive Hypermedia Learning Systems. *ACM Journal of Educational Resources in Computing (JERIC)*, 1(1), 2001.
- [3] IEEE P1484.12 Learning Object Metadata Working Group. *Draft Standard for Learning Object Metadata, v. 6.1*. Learning Technology Standards Committee (LTSC), 2001.
- [4] IMS Global Learning Consortium. *IMS Learning Resource Meta-data Specification*.
- [5] Henning Lobin. *Informationsmodellierung in XML und SGML*. 2001.
- [6] Open University of the Netherlands. *Educational Modelling Language*.
- [7] Abdulmotaleb El Saddik, Stephan Fischer, and Ralf Steinmetz Steinmetz. Reusability and Adaptability of Interactive Resources In Web-Based Educational Systems. *ACM Journal of Educational Resources in Computing (JERIC)*, 1(1), 2001.
- [8] Abdulmotaleb El Saddik, Amir Ghavam, Stephan Fischer, and Ralf Steinmetz. Metadata for Smart Multimedia Learning Objects.
- [9] C. M. Sperberg-McQueen, Claus Huitfeldt, and Allen Renear. Meaning and Interpretation of markup. *Markup Languages - Theory & Practice*, 2(3):215–234, 2000.
- [10] Simon Wiest and Andreas Zell. Improving Web Based Training using an XML content base. In *World Conference on Educational Multimedia, Hypermedia & Telecommunication (ED-MEDIA)*, pages 2045–2050, 2001.