

Wird erscheinen in: *Computerlinguistik – Neue Wege in der Lehre*,
Sonderheft von *Sprache und Datenverarbeitung*

eLearning und offene Standards:

Zum Einsatz XML-strukturierter Lernobjekte

Henning Lobin Maik Stührenberg Georg Rehm

Justus-Liebig-Universität Gießen
Institut für Germanistik
Angewandte Sprachwissenschaft und Computerlinguistik
Otto-Behagel-Strasse 10 D
35394 Gießen

{Henning.Lobin|Maik.Stuehrenberg|Georg.Rehm}@uni-giessen.de

Version vom 3. Juni 2003

Inhaltsverzeichnis

1	Einleitung	1
2	MiLCA – eLearning in der Computerlinguistik	1
3	eLearning – der aktuelle Stand aus technischer Sicht	2
4	Strukturierung von Lernobjekten mit Hilfe von XML	3
4.1	Eine kurze Einführung in XML	3
4.2	XML-strukturierte Lernobjekte	6
4.3	Die MiLCA-DTD	7
4.3.1	Der Learning-Object-Metadata-Standard	7
4.3.2	XML-Implementation	8
5	Einsatz der Lehr-/Lernplattform ILIAS	12
5.1	Funktionsmerkmale von ILIAS	12
5.2	Import von Lernobjekten in die Lernplattform	14
6	Die Zukunft – adaptives Lernen und intelligente eLearning-Systeme	14
	Literatur	16

1 Einleitung

Der Begriff eLearning verfolgt die Idee, Inhalte statt durch einen Lehrer aus Fleisch und Blut durch einen Computer (Computer Based Training, CBT) bzw. Webbrowser (Web Based Training, WBT) zu vermitteln. Hierbei muss zunächst die prinzipielle Unterscheidung getroffen werden, ob die digitalen Medien den menschlichen Lehrenden ersetzen oder ihn unterstützen sollen (Blended Learning, Sauter und Sauter, 2002). Mit dem Einsatz von CBT oder WBT geht eine Loslösung vom Klassenzimmer einher: Multimedial aufbereitetes und sequenziertes Lernmaterial, *Lernobjekte*, können die Inhalte nicht nur ortsunabhängig, sondern auch für den Lernenden durch interaktive Elemente transparenter und anschaulicher gestalten, Experimente müssen nicht mehr im Labor, sondern können auch auf dem heimischen Rechner oder dem PDA durchgeführt werden (allgemein hierzu Wendt, 2003). Durch Hyperlinks verknüpfte Lernobjekte erlauben das Nachschlagen von Begriffen und bieten die Möglichkeit des Wiederholens von Inhalten – zu jeder beliebigen Zeit, an jedem beliebigen Ort. Lehrplattformen sollen dem Benutzer die gewünschten Lerninhalte in einer speziell für ihn aufbereiteten Weise präsentieren. Um diese Vision zu realisieren, existieren verschiedene staatliche und industrielle eLearning-Initiativen. Die benutzten Ansätze und Technologien sind jedoch häufig untereinander nicht kompatibel, was die Verwendung standardisierter Formate und offener Systeme sinnvoll erscheinen lässt, um den kritischen Aspekt der Nachhaltigkeit gewährleisten zu können.

Der vorliegende Beitrag betrachtet die Entwicklung und den Einsatz von Lernobjekten aus texttechnologischer Perspektive (Lobin und Lemnitzer, 2003, Rehm, 2003). Der Artikel basiert auf Arbeiten, die im Rahmen eines durch das Bundesministerium für Bildung und Forschung (BMBF) geförderten Projekts durchgeführt werden, das die Entwicklung einer Lehrplattform für die Computerlinguistik-Ausbildung zum Ziel hat.¹ Im Bereich der Texttechnologie sind in den letzten Jahren Methoden entwickelt worden, die es erlauben, textbasierte Inhalte, die auch multimediale Elemente umfassen können, effizient und nachhaltig zu verarbeiten. Mit einer derartigen Aufbereitung von Lernobjekten 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 MiLCA – eLearning in der Computerlinguistik

Das Projekt MiLCA („Medienintensive Lehrmodule in der Computerlinguistik-Ausbildung“) ist ein vom BMBF im Rahmen des Programms „Neue Medien in der Bildung“ gefördertes Verbundprojekt, an dem die Universitäten Bonn, Gießen, Osnabrück, Saar-

¹ Die in diesem Beitrag dargestellten Arbeiten werden vom BMBF im Rahmen des Programms *Neue Medien in der Bildung* im Projekt MiLCA („Medienintensive Lehrmodule in der Computerlinguistik-Ausbildung“, Kennzeichen: PT-NMB – 08NM167 F) gefördert.

brücken und Tübingen als Projektpartner beteiligt sind.² Die Ziele des Projekts sind die Erstellung von Lehrmodulen für Computerlinguistik-Studiengänge auf nationaler und internationaler Ebene, der Einsatz und die Evaluation dieser Module in der Lehre an den Standorten der beteiligten Universitäten sowie bei industriellen und akademischen Projektpartnern im In- und Ausland, die Integration aller Lehrmodule in eine einheitliche Lernumgebung und schließlich die Gründung einer Verwertungsgesellschaft, um die Pflege und Verbreitung der Module über den Förderzeitraum hinaus gewährleisten zu können. Ein übergreifendes Ziel betrifft die Erprobung und Implementierung internationaler Metadaten-Standards zur Beschreibung von Lernobjekten.

Die beteiligten Universitäten sind jeweils für eigenständige, in sich abgeschlossene Module und Lerninhalte zuständig, die die folgenden Schwerpunkte verfolgen: Computerlinguistische Algorithmen, Semantik, gesprochene Sprache, Grammatikformalismen und Parsing, Texttechnologie, Information Retrieval, Computerlexikographie, Dialogsysteme und computerlinguistische Verfahren für Sprachlernsysteme. Eine zusätzliche Arbeitsgruppe kümmert sich um die Evaluation der Materialien im konkreten Lehrbetrieb und steht als Ansprechpartner für mediendidaktische Fragen zur Verfügung.

3 eLearning – der aktuelle Stand aus technischer Sicht

Betrachtet man zur Zeit im Einsatz befindliche eLearning-Systeme (Schulmeister, 2001), die auf den WBT-Markt ausgerichtet sind, so kann festgehalten werden, dass die Datenstrukturen dieser Produkte untereinander häufig inkompatibel sind und sich alle als eine Kombination aus einer HTML-basierten Oberfläche und einer zu Grunde liegenden Datenbank darstellen. HTML ist als Datenhaltungsformat für eLearning-Inhalte, die inhärent gut strukturierbare Informationen darstellen, nur bedingt geeignet, da es kaum Möglichkeiten zur semantischen Auszeichnung und nur wenige Elemente zur expliziten Strukturierung bietet. Obwohl es die vorherrschende Markup-Sprache zur Darstellung textueller Inhalte ist, verwenden nahezu alle eLearning-Systeme zusätzlich proprietäre Formate wie Macromedia Flash oder Shockwave, um Animationen oder interaktive Präsentationen darzustellen. Das hierdurch entstehende Problem 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 immensem Aufwand vollzogen werden.

Offene Formate und Open-Source-Lösungen dagegen erreichen sehr schnell eine große Verbreitung und sind in vielfältigen Ausprägungen 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

² Informationen zu MiLCA befinden sich unter <http://milca.sfs.uni-tuebingen.de>, der Projektträger stellt sich unter <http://www.gmd.de/PT-NMB/> vor.

die Pflege der Software und somit die Weiterführung und den Ausbau des eigenen eLearning-Angebots. Ein zentraler Aspekt in diesem Kontext betrifft die Erweiterung von Metadaten-Beschreibungen zum Zwecke der Beschreibung adaptiver Lernobjekte (vgl. Conlan et al., 2001 und Saddik et al., 2002).

Auch auf der Lernerseite ergeben sich Nachteile durch die Benutzung proprietärer Systeme und Formate. Anwender, die aus unterschiedlichen Gründen die notwendigen PlugIns für multimedialen Content nicht installiert haben (oder für deren Betriebssysteme eine solche Erweiterung nicht verfügbar ist), werden unmittelbar von deren Nutzung ausgeschlossen.

Allgemein kann festgehalten werden, dass die Verwendung proprietärer Ansätze sowohl die Wiederverwendung als auch die Modularisierung von Lernobjekten stark einschränkt. Ein Mittel zur Vermeidung der geschilderten Probleme ist der Einsatz XML-strukturierter Daten (vgl. Saddik et al., 2001 bzgl. der Wiederverwendung und Wiest und Zell, 2001).

4 Strukturierung von Lernobjekten mit Hilfe von XML

Im Folgenden gehen wir zunächst auf die Extensible Markup Language, XML, ein und stellen ihre wesentlichen syntaktischen Mechanismen zur Definition von Dokumentgrammatiken (DTDs) vor. Der sich anschließende Abschnitt 4.2 knüpft die Verbindung von XML zu Lernobjekten und thematisiert die Vorteile, die durch den Einsatz XML-basierter Formate zur Repräsentation von Lernobjekten entstehen. Abschnitt 4.3 geht auf die im Rahmen von MiLCA entworfene Markup-Sprache (in Form einer XML-DTD und einer XML-Schema-Beschreibung) ein, deren Mechanismen zur Notation von Metadaten sich an dem LOM-Standard orientieren, der in Abschnitt 4.3.1 vorgestellt wird. Abschnitt 4.3.2 enthält abschließend ein kommentiertes Beispiel eines nach der MiLCA-DTD annotierten Lernobjekts.

4.1 Eine kurze Einführung in XML

Das zentrale Mittel zur nachhaltigen Nutzung textbasierter Lehrmaterialien bildet die strukturelle Textauszeichnung. Hierunter wird die Einfügung von Markierungen in einen Text verstanden, die spezifische Eigenschaften einzelner Textteile explizit auszeichnen (Sperberg-McQueen et al., 2000). 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, ISO 8879, 1986), einer standardisierten Metasprache zur Definition von Markup-Sprachen, stand ab 1986 eine Methode zur Verfügung, durch die die Textauszeichnung von bestimmten Anwendungssystemen unabhängig wurde. Die bekannteste Anwendung von SGML wurde die Hypertext Markup Language (HTML, Raggett et al., 1999). Eine vereinfachte Version von SGML, die Extensible Markup Language (XML, Bray et al., 2000), hat SGML heute weitgehend ersetzt.

Eine der grundlegenden Neuerungen im Bereich des Dokument-Markup besteht darin, die durch Tags annotierten Textteile in einen hierarchischen Zusammenhang zu bringen, anstatt sie lediglich linear anzuordnen (Lobin, 2001). Die Textteile befinden sich dadurch nicht nur in der noch immer vorhandenen sequenziellen Reihenfolge, sondern stehen zusätzlich in einer Teil-Ganzes-Beziehung zueinander. Ein Textelement „Kapitel“ kann somit 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 annotierter Dokumente 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 Validierungs-Techniken angewendet werden, die aus dem Compilerbau stammen. 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 nicht nur kodifiziert, wie die Baumstruktur eines Dokuments formal zu notieren ist, es beinhaltet auch eine Sprache zur Definition von Dokumentgrammatiken, die als Dokumenttyp-Definitionen (DTDs) bezeichnet werden.

Bei der Textstrukturierung können zwei Ebenen unterschieden werden, auf denen Informationseinheiten erscheinen können. Hierbei handelt es sich einerseits um die Ebene der konkreten Daten, andererseits um die Ebene abstrakter Einheiten, die Daten Funktionen zuordnen und sie gruppieren. Auf der Ebene der Elemente kann man entsprechend zwei verschiedene Typen unterscheiden: Daten-Elemente enthalten unmittelbar die konkreten Daten und Container-Elemente enthalten selbst wiederum Elemente, wobei es sich um Daten- oder Container-Elemente handeln kann.³ Daten-Elemente enthalten üblicherweise Text in Form beliebiger Zeichenfolgen, in XML sieht die Deklaration eines solchen Elements innerhalb einer DTD wie folgt aus:

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

³ Daten- und Container-Elemente können auch in einer Mischform auftreten, jedoch wird gewöhnlich empfohlen, bei der Definition von Elementen eine eindeutige Zuordnung zu einer der beiden Gruppen vorzusehen.

Das Element `autor` darf also Daten vom Typ `PCDATA` enthalten kann. `PCDATA` ist ein vordefiniertes Schlüsselwort und erlaubt beliebige Zeichenketten. Container-Elemente organisieren Elemente:

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

Das Element `buch` ist hier als Abfolge von vier Daten-Elementen definiert. Die Festlegung, in welcher Weise ein Container-Element im Einzelnen andere Elemente organisiert, geschieht durch das Inhaltsmodell, den 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 ihren Namen kein weiteres Zeichen folgt. Es existieren drei weitere Statusangaben für Elemente: `a?` bedeutet, dass das Element `a` einmal auftreten kann, es kann aber auch ausgelassen werden. Bei `a+` muss `a` mindestens einmal, es kann jedoch beliebig oft auftreten. Benutzt man `a*`, so kann `a` einmal oder beliebig oft auftreten, kann aber auch ausgelassen werden. Wir können nun also die folgenden Statusänderungen vornehmen:

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

Durch diese Deklaration ist es 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. Durch `a, b` wird festgelegt, dass `b` auf `a` folgen muss, wohingegen `a|b` eine beliebige Reihenfolge ausdrückt.

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

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

Bei dieser Deklaration wird für das Attribut `typ`, das dem Element `buch` zugeordnet ist, eine Liste mit drei möglichen Werten angegeben und als Standardbelegung der Wert `monografie` vorgesehen. Das durch `REQUIRED` als obligatorisch spezifizierte Attribut `preis` kann Ziffern- und Buchstabenfolgen enthalten.

Man kann die Verwendung strukturierter Dokumente als das Durchlaufen eines Lebenszyklus konzeptualisieren (Lobin, 2000). Danach folgen auf die Phasen der *Strukturierung* und der *Datenerfassung* die der *Bearbeitung* und des *Viewings* bzw. der *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.

4.2 XML-strukturierte Lernobjekte

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 Metadaten – Informationen *über* die annotierten Informationen – 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 (*Browsing*), eine Gruppe von gleich ausgezeichneten Textabschnitten kann grafisch gleich behandelt werden (*visuelle Textgestaltung*) und Metadaten können auch genutzt werden, um in einem Korpus Textabschnitte mit bestimmten Eigenschaften aufzufinden (*Information Retrieval*). Im Falle eines Lernobjekts kann man demzufolge sagen, dass Regeln dessen konstitutive Teile so miteinander in Verbindung setzen, dass die Bedeutung des Lernobjekts 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 Lerners ein. In gut strukturierten Lernobjekten sollten diese Überlegungen in expliziter Form aus der Annotation deutlich werden. Die DTD, die die möglichen Strukturen eines Lernobjekts beschreibt, beinhaltet somit abstraktes Strukturwissen zur Didaktik von Lehrmodulen, die Annotation eines Lernobjekts dessen konkrete Umsetzung.

In den einzelnen Phasen des Lebenszyklus von Lernobjekten erweist sich die Verfügbarkeit einer expliziten Annotation in verschiedener Weise als nützlich:

- *Produktion*: Während der inkrementellen Produktion eines Lernobjekts verfolgen Editoren den Aufbau des entstehenden Dokuments und gleichen ihn mit der DTD ab. Es kann angegeben werden, welche Elemente an einer bestimmten Stelle strukturell eingefügt werden können, und unerlaubte Element Einfügungen können verhindert werden. Die didaktische Struktur und die geforderten Metadaten zur Nutzung und zu den Voraussetzungen des Lernobjekts werden also von vornherein in Hinsicht auf ihre Vollständigkeit überwacht.
- *Validierung*: Ein vorliegendes Lernobjekt wird dahingehend überprüft, ob sich seine Baumstruktur von den in der Grammatik enthaltenen Regeln ableiten lässt. Nur wenn dies sichergestellt ist, kann das einzelne Lernobjekt in ein größeres Geflecht von Lernobjekten eingegliedert und in flexibler Weise im Rahmen eines Lehrsystems genutzt werden.
- *Retrieval*: Das Auffinden von Informationen kann durch DTDs unterstützt werden. Soll z. B. ein Textelement `Test` mit einem bestimmten Stichwort gesucht

werden, kann die Suche auf solche Abschnitte des Lernobjekts beschränkt werden, in denen ein Element `Test` überhaupt nur vorkommen kann.

Weitere Verwendungsideen der Annotation XML-strukturierter Lernobjekte reichen über den gegenwärtigen Stand in bereits eingesetzten Lehrsystemen noch hinaus. Tendenzen einer mittelfristig zu erwartenden Ausweitung der Nutzung von Lernobjekten werden in Abschnitt 6 diskutiert.

4.3 Die MiLCA-DTD

Es existieren verschiedene Ansätze, Lerninhalte mit Hilfe von XML zu strukturieren, ein Beispiel ist die komplexe Educational Modelling Language (EML, <http://eml.ou.nl>). 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 des IMS Global Learning Consortiums (<http://www.imsproject.org/metadata/>).

Da die Verwendung von Lehrmodulen in hohem Maße abhängig ist von der Annotation mit Metadaten, war auch bei der Entwicklung einer eigenständigen Markup-Sprache für MiLCA der erste Schritt die Sichtung der verschiedenen Metadaten-Standards. Der Metadatensatz Learning Object Metadata des IEEE Learning Technology Standards Committee (LTSC, <http://ltsc.ieee.org>) gilt als wichtigste Initiative zu diesem Thema: Learning Object Metadata (LOM) ist seit Juni 2002 ein IEEE-Standard, der in unterschiedlicher Form in vielen Projekten eingesetzt wird, so bauen etwa die Implementierungen des IMS Global Learning Consortium (<http://www.imsproject.org>) auf LOM auf. Aus diesem Grund enthält auch die MiLCA-DTD den vollständigen LOM-Metadatensatz (Version 1.0 Final Draft), wodurch die Nachhaltigkeit der annotierten Inhalte gesichert wird.

4.3.1 Der Learning-Object-Metadata-Standard

Der Learning-Object-Metadata-Standard (LOM) definiert neun Kategorien von Metadaten, die sich ihrerseits aus mehreren Einträgen zusammensetzen (IEEE P1484.12 Learning Object Metadata Working Group, 2002):

1. *General*: Beinhaltet allgemeine Informationen über das Lernobjekt wie Titel, verwendete Sprachen, Beschreibung und Schlüsselwörter. Das wichtigste Attribut in dieser Kategorie ist der Aggregationsgrad (*Aggregation Level*), der Aussagen über die Komplexität eines Lernobjekts macht:

Der Aggregationsgrad 1 beschreibt ein atomares, nicht weiter unterteilbares Lernobjekt, z. B. eine Abbildung. Lernobjekte des Aggregationsgrades 2 sind aufgebaut wie eine Textseite (mit evtl. eingebetteten Bildern). Der Aggregationsgrad 3 bezeichnet eine Ansammlung von Level 2 Lernobjekten, die einen Kurs repräsentieren, z. B. in Form einer Übersichtsseite mit daran angeschlossenen In-

haltsseiten. Ein Lernobjekt des Aggregationsgrades 4 schließlich ist ein Set von Kursen, das dem Erreichen eines Abschlusses dient.

2. *LifeCycle*: Enthält Angaben über 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 Lernobjekts, 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, die semantische Dichte sowie den Schwierigkeitsgrad eingetragen werden.
6. *Rights*: Beinhaltet Informationen zu Urheberrechten und evtl. Kosten.
7. *Relation*: Mit Hilfe dieser Informationen können Beziehungen zwischen einzelnen Lernobjekten hergestellt werden. Dazu stehen verschiedene Arten von Relationen zur Verfügung, z. B. *is-part-of*, *is-basis-for* oder *is-referenced-by*.
8. *Annotation*: Enthält Anmerkungen zum Lernobjekt.
9. *Classification*: Hier kann das Lernobjekt mit Hilfe eines Taxonomie-Pfades und Schlagworten in den Kontext des jeweiligen Fachgebietes eingeordnet werden.

Diese Metadaten beschreiben Lernobjekte in vielfacher Hinsicht. Die in ihnen hinterlegten Informationen erlauben es nicht nur dem menschlichen Lehrenden, Inhalte nach bestimmten Kriterien zu selektieren, sie erleichtern auch eLearning-Plattformen die automatische Auswahl von Lernobjekten.

4.3.2 XML-Implementation

Die MiLCA-DTD ist wie folgt aufgebaut: Das Element `LearningObject` stellt die Dokumentwurzel dar. Ein `LearningObject` enthält die Tochterelemente `MetaData`, `Layout-Information`, `Content`, `Test`, `Bibliography` und `Glossary`. Abgesehen von `MetaData` sind alle Tochterelemente optional, d. h. auch ein Lernobjekt, das nur aus Metadaten besteht, ist gültig. Somit können nicht-textuelle oder sogar nicht-digitale Lernobjekte beschrieben werden.

```
<!ELEMENT LearningObject (MetaData, Lifecycle?, 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 4.3.1 dargestellt wurden. Alle Metadatenfelder, die einen Wert aus einem vom Standard vorgegebenen Vokabular annehmen können, sind als Attribute realisiert worden, um den Autoren die

Arbeit zu erleichtern und korrekte Attribut-Wert-Paare sicherstellen zu können. Wie aus den Abbildungen 1 und 2 ersichtlich ist, nehmen die Metadaten einen beachtlichen Teil eines Lernobjekts ein. Dabei ist zu beachten, dass das dargestellte Lernobjekt einen nahezu vollständigen Satz an Metadaten enthält, der sich für alle Lernobjekte des Aggregationsgrades 3 anbietet. Eingebettete Level 2 Lernobjekte können auf einen Großteil der Metadaten verzichten. Die Angaben im Feld `General` werden genutzt, um in der Ausgabe (siehe Abbildung 3) den Titel zu generieren. Denkbar ist auch die Ausgabe des Elementinhalts von `Description`, um eine kurze Inhaltsangabe des Lernobjekts zu präsentieren.

`LayoutInformation` ist ein leeres Element, das Angaben über ein zu verwendendes Cascading Stylesheet für die Online-Ausgabe enthält. Weitere Informationen können hier nach Bedarf hinzugefügt werden. Das annotierte Lernobjekt verweist in Zeile 90 auf eine CSS-Datei, die bei allen Projektpartnern eingesetzt wird, um eine einheitliche Darstellung der XHTML-Version sicherzustellen.

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

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

Durch den rekursiven Aufbau und die hierdurch ermöglichte Verschachtelung können Lernobjekte 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 Fischer, 2001).

Das Element `Text` enthält Absätze (`Paragraph`), die mit Hilfe des Attributs `Characteristic` genauer spezifiziert werden können (z. B. als Überschrift – wie der Absatz in Zeile 101 – oder Zusammenfassung) Ein weiteres (im Beispiel nicht verwendetes) Attribut kennzeichnet Absätze als optional. Somit können Lernobjekte bis zur Absatzebene variabel gestaltet werden. Innerhalb eines Absatzes sind verschiedene logische Textauszeichnungen möglich: Betonungen, Kommentare, Hervorhebungen, Zitate, Referenzen, (externe) Links, Beispiele, Formeln, Listen, Tabellen, Glossareinträge und Übungsfragen. Dabei wird von den XML-eigenen Möglichkeiten der Referenzierung Gebrauch gemacht, um Beziehungen zwischen einzelnen Wörtern, Textteilen, Literatureinträgen oder Lernobjekten herzustellen. Einige der genannten Elemente finden sich im Beispiel.

Im optionalen Element `Test` können Wissensfragen an den Lernenden gerichtet werden. Das Element ist zum jetzigen Zeitpunkt noch nicht vollständig ausdifferenziert. Eine Erweiterung unter Berücksichtigung der IMS Question & Test Interoperability Specification (<http://www.imspj.org/question/>) ist geplant. Durch die schon angesprochene rekursive Verschachtelung von Lernobjekten ist es möglich, Tests nur an ein eng begrenztes Themengebiet (in Form eines `LearningObjects` des Aggregationsgrades 2) oder an eine vollständige Lerneinheit (in Form eines `LearningObjects` mit dem Aggregationsgrad 3) zu knüpfen.

Das Element `Bibliography` enthält Literaturangaben, die in einem `LearningObject` verwendet wurden. Die Strukturierung dieses Elements wurde in großen Teilen von dem Projekt BibTeXML der ETH Zürich übernommen. Ziel dieses Projekts war die automatisierte Überführung von Literaturangaben, die im weit verbreiteten BibTeX

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE LearningObject SYSTEM "lo_final.dtd">
3 <LearningObject>
4   <MetaData>
5     <General Structure="Collection" AggregationLevel="2">
6       <Identifier Catalog="MilCA" Entry="A-5-3"/>
7       <Title Language="de">Einführung in die Hypertext Markup Language</Title>
8       <Language>de</Language>
9       <Description Language="de">
10        Gegenstand dieser Lerneinheit ist die Hypertext Markup Language
11        (HTML) und deren Neuformulierung XHTML.
12      </Description>
13      <Keyword Language="de">Hypertext</Keyword>
14      <Keyword Language="de">HTML</Keyword>
15    </General>
16    <Lifecycle Status="Draft">
17      <Version Language="de">0.3</Version>
18      <Contribute Role="Author">
19        <Entity>
20          <vCard>
21            <FN>Maik Stührenberg</FN>
22            <N><Family>Stührenberg</Family><Given>Maik</Given></N>
23            <ORG>
24              <Orgname>Justus-Liebig-Universität Gießen</Orgname>
25              <Orgunit>Angewandte Sprachwissenschaft und
26                Computerlinguistik</Orgunit>
27            </ORG>
28            <ADR TYPE="Work">
29              <Street>Otto-Behaghel-Str. 10 D</Street>
30              <Locality>Gießen</Locality>
31              <Pcode>35394</Pcode>
32              <Country>Germany</Country>
33            </ADR>
34            <TEL TYPE="Work">0641/99-29056</TEL>
35            <EMAIL TYPE="Internet">Maik.Stuehrenberg@Uni-Giessen.de</EMAIL>
36          </vCard>
37        </Entity>
38        <Date>2002-01-26</Date>
39      </Contribute>
40    </Lifecycle>
41    <Meta-Metadata MetadataScheme="LOM□□□1.0">
42      <Identifier Catalog="MilCA" Entry="M-A-5-3"/>
43      <Contribute Role="Author">
44        <Entity>Maik Stührenberg</Entity>
45        <Date>2002-01-26</Date>
46      </Contribute>
47    </Meta-Metadata>
48    <Technical Format="text-xml">
49      <Size>5</Size>
50      <Requirement>
51        <Type>
52          <Browser Name="Any"/>
53        </Type>
54      </Requirement>
55      <InstallationRemarks Language="de">keine</InstallationRemarks>
56      <Duration>00:20:00</Duration>
57    </Technical>

```

Abbildung 1: Beispiel eines Lernobjekts (wird fortgesetzt in Abb. 2)

```

58     <Educational InteractivityType="Expositive" InteractivityLevel="Medium"
59         LearningResourceType="NarrativeText"
60         SemanticDensity="Low" IntendedEndUserRole="Learner"
61         Context="HigherEducation" Difficulty="Medium">
62     <TypicalAgeRange Language="de">18-75</TypicalAgeRange>
63     <TypicalLearningTime>00:15:00</TypicalLearningTime>
64     <Description Language="de">Einführungstext</Description>
65     <Language>de</Language>
66 </Educational>
67 <Rights Cost="No" CopyrightAndOtherRestrictions="No">
68     <Description Language="de">Der Text wurde neu erstellt, allerdings gibt
69         es mehrere Zitate</Description>
70 </Rights>
71 <Relation Kind="IsBasisFor">
72     <Resource>
73         <Identifier Catalog="MiLCA" Entry="A-5-3-1"/>
74         <Description Language="de">Basis für fast alle weiteren
75             Lerneinheiten</Description>
76     </Resource>
77 </Relation>
78 <Classification Purpose="EducationalLevel">
79     <TaxonPath>
80         <Source Language="de">Linguistik</Source>
81         <Taxon Id="T1">
82             <Entry Language="de">Linguistik-Computerlinguistik-Hypertext</Entry>
83         </Taxon>
84     </TaxonPath>
85     <Keyword Language="de">Linguistik</Keyword>
86     <Keyword Language="de">Computerlinguistik</Keyword>
87     <Keyword Language="de">Hypertext</Keyword>
88 </Classification>
89 </MetaData>
90 <LayoutInformation CSS-URL="milca.css"/>
91 <Content>
92     <Text>
93         <Paragraph>
94             HTML (die <Emph>H</Emph>yper<Emph>t</Emph>ext <Emph>M</Emph>arkup
95             <Emph>L</Emph>anguage) ist die populärste Markup-Sprache. Ihre
96             Popularität verdankt HTML ihrer Einfachheit und Universalität.
97             Entwickelt und gepflegt wurde und wird HTML vom World Wide Web
98             Consortium (<Link Href="http://www.w3.org"
99             Title="Homepage des W3C">W3C</Link>).
100         </Paragraph>
101         <Paragraph Characteristic="Headline">
102             Aufbau einer HTML-Datei
103         </Paragraph>
104         <Paragraph>
105             ...
106         </Paragraph>
107         <Paragraph>
108             <Example Id="HTML-Darstellung" Language="de" Status="Well-formed">
109                 <ExampleText><Code>&lt;h1&gt;HTML&lt;/h1&gt;</Code></ExampleText>
110             </Example>
111         </Paragraph>
112             ...
113     </Text>
114 </Content>
115 <LearningObject>

```

Abbildung 2: Beispiel eines Lernobjekts (Fortsetzung aus Abb. 1)

Format vorliegen, in ein XML-Format. Da das Projekt nicht mehr fortgesetzt wird, werden auch XSLT-Stylesheets des „BibTeX-XML-HTML Bibliography Project“ genutzt.⁴ Die Verwendung BibTeX-kompatibler Angaben in Verbindung mit Import- und Exportskripten ermöglicht eine einfache Wiederverwendung sowohl der in MiLCA verwendeten als auch bereits vorliegender Literaturangaben.

Das Glossar enthält einzelne Einträge (`GlossaryItem`), denen eine oder mehrere Definitionen zugeordnet werden können. Da die Einträge per XML-Referenzen mit den Definitionen verbunden werden, können keine Glossareinträge ohne erklärende Definition vorkommen. Auch im Fließtext als Glossareintrag markierte Wörter müssen zunächst im Glossar mit einer Definition eingetragen werden.

Zum aktuellen Zeitpunkt ist die Markup-Sprache bei allen Projektpartnern (vgl. Abschnitt 2) im produktiven Einsatz. Seit einigen Monaten waren keinerlei Änderungen an der DTD und der XML-Schema-Beschreibung notwendig, Korrekturarbeiten finden allenfalls an den XSLT-Stylesheets zur Konvertierung statt. Wesentliche Aufgaben für die zukünftige Weiterentwicklung sind Ergänzungen in Bezug auf didaktische Konzepte (siehe Abschnitt 6) und die Integration von Ontologien und Taxonomien, die eine präzisere Verwaltung der Lernobjekte ermöglichen werden.

5 Einsatz der Lehr-/Lernplattform ILIAS

Lernobjekte werden von den Autoren innerhalb des MiLCA-Projekts mit Hilfe beliebiger XML-Editoren erstellt, von denen eine große Auswahl sowohl im kommerziellen als auch im Open-Source-Bereich für nahezu alle Betriebssysteme erhältlich ist. Nachdem ein Lernobjekt erzeugt und mit Hilfe eines XML-Parsers auf seine Gültigkeit (Validität) überprüft worden ist, folgt eine Verarbeitung durch die XSLT-Stylesheets `1om2html.xslt` bzw. `1om2fo.xslt` (vgl. Clark, 1999). Diese erlauben eine automatische Überführung der Inhalte nach XHTML (Pemberton, 2002) mit eingebetteten MathML-Annotationen zur Darstellung von Formeln und anderen mathematischen Inhalten für die Online-Darstellung und XSL FO mit anschließender Wandlung nach PDF als Format zum Offline-Lesen.⁵

5.1 Funktionsmerkmale von ILIAS

Die Projektpartner des MiLCA-Verbundes nutzen die Open-Source-Lernplattform ILIAS („Integriertes Lern-, Informations- und Arbeitskooperations-System“, <http://www.ilias.uni-koeln.de>).⁶ ILIAS ist eine Client-Server-Anwendung, die das Erstellen, Be-

⁴ Siehe <http://www.sfs.uni-tuebingen.de/~auth/xml2html/>.

⁵ Die XSLT-Stylesheets entsprechen prinzipiell den in Miller (2003) dargestellten, bieten jedoch sehr viel umfangreichere Funktionen.

⁶ ILIAS wurde ursprünglich im Rahmen des Projekts VIRTUS („Virtuelle Universitätssysteme“, Laufzeit 1997–2001) an der Wirtschafts- und Sozialwissenschaftlichen Fakultät der Universität Köln entwickelt (siehe hierzu auch Barthold, 2002). Das Entwicklerteam in Köln umfasst derzeit vier wissenschaftliche Mitarbeiter und vier Hilfskräfte, die hauptamtlich mit der Entwicklung von ILIAS beschäftigt sind. Externe Partner sind mit der Implementierung einzelner Teilfunktionen beauftragt bzw. führen diese auf freiwilliger Basis durch (z. B. die Universität der Bundeswehr Hamburg).

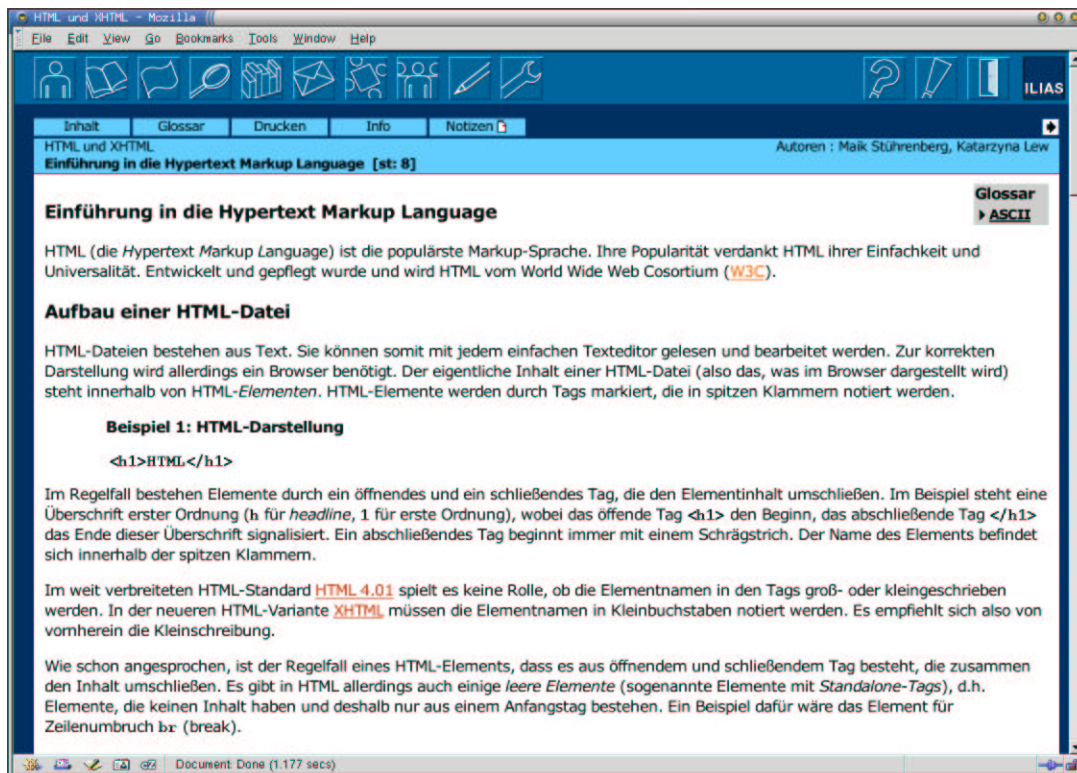


Abbildung 3: Ansicht einer Lerneinheit in der eLearning-Plattform ILIAS

arbeiten und Darstellen von Lehrmaterialien in einer integrierten, Web-basierten Umgebung ermöglicht. ILIAS ist ein LAMP-System, d. h. auf einem Linux-Rechner arbeitet Apache als Webserver und sämtliche Daten werden in einer MySQL-Datenbank gepflegt, auf die PHP als Middleware zugreift. Zahlreiche öffentlich geförderte Projekte und Vorhaben setzen ILIAS ein, so wurde das System beispielsweise – nach einer umfangreichen Evaluation von 136 Plattformen – im Herbst 2002 von einer Kommission des Landes Brandenburg als möglicher Kandidat für den landesweiten Einsatz in der Erwachsenenbildung ausgewählt (Berger, 2002). Nach Angaben der ILIAS-Entwickler waren Anfang 2003 mehr als 430 registrierte ILIAS-Server im Einsatz, wobei die Zahl der nicht registrierten Installationen um ein vielfaches höher sein dürfte.⁷

Nach der Authentifizierung gegenüber dem System fungiert der persönliche Schreibtisch des Benutzers als Schaltzentrale. Über Icons und benutzerfreundliche Aufstellungen kann der Lerner auf freigegebene Lerneinheiten zugreifen, Bookmarks ortsunabhängig verwalten, in verfügbaren Lerneinheiten nach Textteilen oder Personen suchen, Forenbeiträge lesen und verfassen oder ein Chat-Werkzeug aktivieren, um in verteilten Kleingruppen zu arbeiten. Abbildung 3 zeigt einen Ausschnitt der Lerneinheit

⁷ Der Zweck der offiziellen Registrierung einer produktiven Installation besteht darin, einen ILIAS-Server durch eine eindeutige Nummer referenzieren zu können, um Querverweise auf Lerneinheiten Dritter zu realisieren.

„HTML und XHTML“, die als Einführung innerhalb des MiLCA-Moduls „Texttechnologie“ entstanden und deren XML-Quelle in den Abbildungen 1 und 2 in gekürzter Form dargestellt ist.

5.2 Import von Lernobjekten in die Lernplattform

Der Import der Lernobjekte in ILIAS erfolgt derzeit noch manuell, d. h. eine mittels XSLT-Stylesheet (`1om2i1ias.xslt`) speziell für die Plattform aufbereitete XHTML-Ausgabe wird von Hand in ILIAS-Lerneinheiten eingebettet, wobei evtl. vorhandene lokale Referenzen angepasst werden müssen.

Derzeit wird an Möglichkeiten für den unmittelbaren Import XML-basierter Repräsentationen von Lerneinheiten für die Version 3 der ILIAS-Plattform gearbeitet, die im Frühjahr 2003 als Alpha-Version veröffentlicht wurde. Die Entwickler verfolgen dabei einen Weg, der auf den Vorarbeiten des MiLCA-Projekts basiert, die im September 2002 auf der ersten ILIAS-Entwicklerkonferenz vorgestellt wurden (vgl. auch Stührenberg, 2002). Da die ILIAS-DTD zur Beschreibung gültiger Lernobjekte in weiten Teilen auf der MiLCA-DTD basiert, wird sich der zukünftige Import von MiLCA-Lernobjekten in die eLearning-Plattform auf eine XSLT-basierte Konvertierung in die ILIAS-DTD sowie den anschließenden Upload beschränken, im besten Fall wird dieser Prozess vollautomatisch durchführbar sein.

Für den nativen Im- und Export XML-basierter Lernobjekte implementieren die ILIAS-Entwickler generische Verfahren, die den flexiblen Umgang mit arbiträren XML-annotierten Daten innerhalb relationaler Datenbankstrukturen ermöglichen.⁸ Importierte oder im ILIAS-eigenen Editor angefertigte bzw. manipulierte XML-Dokumente werden daraufhin einmalig in die benötigten Zielformate konvertiert, die anschließend als HTML- und PDF-Versionen im Dateisystem abgelegt werden, so dass keine dynamische „on the fly“-Transformation stattfinden muss, die sich negativ auf die Performanz des System auswirken würde.

6 Die Zukunft – adaptives Lernen und intelligente eLearning-Systeme

Der hier dargestellte Stand von eLearning-Systemen wird schon in naher Zukunft überholt sein. Aktuelle Forschungen im stetig an Bedeutung gewinnenden 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 sollen. 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 Lernobjekten zu integrieren: Die Educational Modelling Language (EML) stellt

⁸ Diese Methoden sind angelehnt an die Software pyxy, vgl. <http://sourceforge.net/projects/pyxy/>, <http://www.informatics.bangor.ac.uk/~rich/> und Edwards und Hope (2000).

beispielsweise ein didaktisches Rollenmodell zur Verfügung. Darauf aufbauend kombiniert die Spezifikation IMS Learning Design die Ansätze von EML mit bisherigen Arbeiten erstmalig zu einem Gesamtbild (<http://www.imsproject.org/learningdesign/index.cfm>). Darüber hinaus wird derzeit durch die Arbeitsgruppe „Standardisierung – Didaktik eLearning“ des Referats „Entwicklungsbegleitende Normung“ des Deutschen Instituts für Normung (DIN) ein standardisiertes Beschreibungsformat zur Abbildung didaktischer Modelle konzipiert.

Eine entsprechend angepasste eLearning-Plattform wäre mit einer Auswahl von Lernobjekten 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 Lernobjekte wählen. Einsteiger können damit behutsam an ein Thema herangeführt werden, während fortgeschrittene Lerner automatisch Abschnitte überspringen, deren Inhalt sie bereits kennen. Dadurch wird die Lernmotivation durchgängig auf einem hohen Niveau gehalten und jeder Einzelne kann gezielt gefordert und gefördert werden. Voraussetzung dafür ist die Verfeinerung der Abbildung der Relationen von Lernobjekten untereinander – die „Web-Didaktik“ von Meder (2003) stellt diesbezüglich einen vielversprechenden Ansatz dar. Ebenso denkbar wäre der Einsatz von RDF-Beschreibungen (Resource Description Framework, Lassila und Swick, 1999), um die Beziehungen zwischen Lernobjekten in Form von Taxonomien bzw. Ontologien darzustellen und der Plattform darüber hinaus die Möglichkeit zu geben, Schlüsse über didaktische Modellierungen ziehen zu können.

Ein dritter Aspekt besteht schließlich darin, die Materialien selbst in unterschiedlicher medialer und presentationeller Form verfügbar zu machen. XML-strukturierte Lernobjekte verhalten sich neutral gegenüber ihrer medialen Aufbereitung. Schon jetzt werden im MiLCA-Projekt aus ein und demselben Pool von Lernobjekten nicht nur Web-basierte eLearning-Systeme unterstützt, sondern parallel dazu auch Print-Versionen hergestellt. Ein weiterer Schritt ist die Generierung alternativer Versionen, die für andere digitale Endgeräte optimiert sind. 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 persönlicher Gedächtnis- und Verständnisleistungen mit maßgeschneiderten Lehrmaterialien in unterschiedlichen medialen Ausprägungen versorgen, die erzielten Lernfortschritte kontrollieren und in die Berechnung der nächsten Lernschritte einfließen lassen. Grundlage dafür bilden stark mit Metadaten angereicherte, systematisch didaktisch strukturierte Lernobjekte, die system- und medienunabhängig zugänglich sind.

Literatur

- Barthold, Hans-Martin (2002): "Gutes Werkzeug". *DUZ – Das unabhängige Hochschulmagazin* (24): S. 23.
- Berger, Konrad (2002): "Empfehlungen zur Auswahl einer eLearning-Plattform im Rahmen des INNOPUNKT-Programms „Neues Lernen made in Brandenburg“". Arbeitsbericht, frauenhoferinstitut für produktionsanlagen und konstruktionstechnik (ipk), PRO-KMU – Betriebliche Qualifizierung im innovativen Lernverbund. Gefördert aus Mitteln des Landes Brandenburg und der Europäischen Union. Online verfügbar: <http://www.tqua.de/download/arbeitsbericht.pdf>.
- Bray, Tim; Paoli, Jean; Sperberg-McQueen, C. M. und Maler, Eve (2000): "Extensible Markup Language (XML) 1.0 (Second Edition)". Technische Spezifikation, World Wide Web Consortium. Online verfügbar: <http://www.w3.org/TR/2000/REC-xml-20001006>.
- Clark, James (1999): "XSL Transformations (Version 1.0)". Technische Spezifikation, World Wide Web Consortium. Online verfügbar: <http://www.w3.org/TR/xslt/>.
- Conlan, Owen; Hockemeyer, Cord; Lefrere, Paul; Wade, Vincent und Albert, Dietrich (2001): "Extending Educational Metadata Schemas to describe Adaptive Learning Resources". In: *Hypertext '01: Proceedings of the twelfth ACM Conference on Hypertext and Hypermedia*, herausgegeben von Davies, Hugh; Douglas, Yellowlees und Durand, David G. Association for Computing Machinery, ACM, New York, S. 161–162.
- Edwards, Richard und Hope, Sian (2000): "Persistent DOM: An architecture for XML repositories in relational databases". In: *Intelligent Data Engineering and Automated Learning – IDEAL 2000. Data Mining, Financial Engineering, and Intelligent Agents*, herausgegeben von Leung, K. S.; Chan, L. und Meng, H., Springer, Nummer 1983 in Lecture Notes in Computer Science, S. 416–421.
- Fischer, Stephan (2001): "Course and Exercise Sequencing Using Metadata in Adaptive Hypermedia Learning Systems". *ACM Journal of Educational Resources in Computing (JERIC)* 1 (1).
- IEEE P1484.12 Learning Object Metadata Working Group (2002): *Standard for Learning Object Metadata, Final Draft V. 1.0*. Learning Technology Standards Committee (LTSC). Online verfügbar: <http://ltsc.ieee.org/wg12/> (zuletzt eingesehen am 20. Dezember 2002).
- ISO 8879 (1986): "Information Processing – Text and Office Information Systems – Standard Generalized Markup Language". Internationaler Standard, International Organization for Standardization, Genf.
- Lassila, Ora und Swick, Ralph R. (1999): "Resource Description Framework (RDF). Model and Syntax Specification". Technische Spezifikation, World Wide Web Consortium. Online verfügbar: <http://www.w3.org/TR/REC-rdf-syntax/>.
- Lobin, Henning (2000): "Service-Handbücher – Linguistische Aspekte im Document-Lifecycle". In: *Raum, Zeit, Medium – Sprache und ihre Determinanten. Festschrift für Hans Ramge*, herausgegeben von Richter, Gerd; Riecke, Jörg und Schuster, Britt-Marie, Darmstadt: Hessische Historische Kommission, S. 791–808.
- Lobin, Henning (2001): *Informationsmodellierung in XML und SGML*. Berlin, Heidelberg: Springer.
- Lobin, Henning und Lemnitzer, Lothar (Herausgeber) (2003): *Texttechnologie – Anwendungen und Perspektiven*. Tübingen: Stauffenburg. Erscheint.
- Meder, Norbert (2003): *Web-Didaktik. Eine neue Didaktik webbasierten, vernetzten Lernens*. Bielefeld: Bertelsmann Verlag.

- Miller, Mark (2003): "XML Coursebuilder V.1.0 – XML and XSLT format course modules into a comprehensive presentation". *XML Journal* 4 (4): S. 22–25.
- Pemberton, Steven (2002): "XHTML 1.0: The Extensible Hypertext Markup Language (Second Edition)". Technische Spezifikation, World Wide Web Consortium. Online verfügbar: <http://www.w3.org/TR/xhtml1/>.
- Raggett, Dave; Hors, Arnaud Le und Jacobs, Ian (1999): "HTML 4.01 Specification". Technische Spezifikation, World Wide Web Consortium. Online verfügbar: <http://www.w3.org/TR/html401/>.
- Rehm, Georg (2003): "Texttechnologische Grundlagen". In: *Computerlinguistik und Sprachtechnologie – Eine Einführung*, herausgegeben von Klabunde, Ralf; Carstensen, Kai-Uwe; Ebert, Christian; Endriss, Cornelia; Jekat, Susanne; Langer, Hagen und Schiehlen, Michael, Heidelberg: Spektrum. 2. Auflage. Erscheint.
- Saddik, Abdulmotaleb El; Fischer, Stephan und Steinmetz, Ralf (2001): "Reusability and Adaptability of Interactive Resources In Web-Based Educational Systems". *ACM Journal of Educational Resources in Computing (JERIC)* 1 (1).
- Saddik, Abdulmotaleb El; Ghavam, Amir; Fischer, Stephan und Steinmetz, Ralf (2002): "Metadata for Smart Multimedia Learning Objects". *Proceedings of the fourth Australasian Computing Education Conference* .
- Sauter, Annette M. und Sauter, Werner (2002): *Blended Learning – Effiziente Integration von E-Learning und Präsenztraining*. Neuwied, Kriftel: Luchterhand.
- Schulmeister, Rolf (2001): *Virtuelle Universität – Virtuelles Lernen*. München, Wien: Oldenbourg.
- Sperberg-McQueen, C. M.; Huitfeldt, Claus und Renear, Allen (2000): "Meaning and Interpretation of Markup". *Markup Languages - Theory & Practice* 2 (3): S. 215–234.
- Stührenberg, Maik (2002): "Using XML Learning Objects – Current and Future Benefits for WBT Scenarios". In: *International Workshop on Interactive Computer Aided Learning (ICL)*, herausgegeben von Auer, Michael E. und Auer, Ursula. Kassel University Press. Fachhochschule Kärnten, Villach.
- Wendt, Matthias (2003): *CBT und WBT – konzipieren, entwickeln, gestalten*. München, Wien: Hanser.
- Wiest, Simon und Zell, Andreas (2001): "Improving Web Based Training using an XML content base". In: *World Conference on Educational Multimedia, Hypermedia & Telecommunication (ED-MEDIA)*. Tampere/Finland, S. 2045–2050.