

# **Ein Kurseditor für modularisierte Lernressourcen auf der Basis von Learning Objects Metadata zur Erstellung von adaptierbaren Kursen**

Stefan Hoermann<sup>1</sup>, Andreas Faatz<sup>1,3</sup>, Oliver Merkel<sup>1</sup>, Ansgar Hugo<sup>1</sup> und Ralf Steinmetz<sup>1,2</sup>

1 Industrielle Prozess- und Systemkommunikation - KOM  
Fachbereich Elektrotechnik und Informationstechnik  
Technische Universität Darmstadt  
Merckstr. 25 • 64283 Darmstadt

2 Fraunhofer IPSI  
Fraunhofer Gesellschaft  
Dolivostr. 15 • 64293 Darmstadt

3 intelligent views gmbh  
Julius-Reiber-Str. 17  
D-64293 Darmstadt

{hoermann, faatz, merkel, steinmetz}@kom.tu-darmstadt.de, ansgar.hugo@web.de

*Zusammenfassung: k-MED ist ein vom BMB+F gefördertes Gemeinschaftsprojekt elf medizinischer, drei technischer und eines gestalterischen Lehrstuhls innerhalb des Bundeslandes Hessen. An der TU-Darmstadt werden für k-MED Werkzeuge zum Speichern, Verwalten, Auffinden und Strukturieren von modularisierten Lernressourcen für das medizinische Grundstudium entwickelt. Hierfür werden technische Hilfsmittel geboten, um insbesondere bereits bestehende Ressourcen zu beschreiben und mit Ressourcen, die sich in einer multimedialen Wissensbasis befinden, zu verknüpfen. Ziel ist es, dadurch die Wiederverwendung von Lehrmaterialien zu optimieren. Die Nutzung modularisierter Unterrichtsmaterialien stellt Autoren vor die im didaktischen Kontext neuartige Aufgabe, eine wiederverwendbare Wissensbasis zu erstellen. Im vorliegenden Papier wird eine Erweiterung zu Learning Objects Metadata vorgestellt, auf dessen Basis das Erstellen von adaptierbaren Kursen möglich wird. Im Zusammenhang mit rhetorisch-didaktischen Relationen werden Möglichkeiten zur Adaption dieser Kurse gezeigt. Anschließend wird ein Editor beschrieben, mit dem basierend auf einer Wissensbasis adaptierbare Kurse erstellt werden können.*

## **1 Einleitung**

Zur Kostenminimierung bei der Erstellung multimedialer Lernressourcen hat sich deren Modularisierung etabliert. Hierdurch wird die Wiederverwendung der erstellten Ressourcen erheblich verbessert. Einher geht die Verbesserung der Wiederverwendbarkeit modularisierter Lernressourcen mit der Erstellung von Metadaten zu jedem dieser Module. Aus diesem Grund wurde in den letzten Jahren erheblicher Aufwand in die Entwicklung von Metadatenstandards investiert. Ein Ergebnis dieser Bemühungen ist Learning Objects Metadata (LOM), ein Metadatenchema des Learning Technology Standards Committee (LTSC) der IEEE [LWG01], das besonders für die Attributierung von Lernressourcen geeignet ist. Mit Hilfe dieses Schemas werden Lernressourcen, die in der Wissensbasis des k-MED-Rahmensystems gespeichert sind, mit LOM-Datensätzen beschrieben. Im Folgenden werden wir den Begriff

Medienbaustein(MediaBrick) für einzelne Lernmodule benutzen, die mit LOM attribuiert sind[SSFS99].

Durch die Modularisierung der Lernressourcen und der konsequenten Nutzung von LOM wird die Möglichkeit adaptive Lernsysteme zu erstellen fest in den Autorenprozess verankert[See01]. Hierdurch ergeben sich Veränderungen im Autorenprozess und in der Speicherung der Kurse. Ein signifikanter Unterschied des im Verlauf des Papiers vorgestellten Kurseditors zu gängigen anderen Editoren für die Zusammenstellung von Kursen basierend auf modularisierten Lernmaterialien ist, dass die erzeugten Kurse durch Aufbau einer Objekthierarchie mit gerichteten Relationen zwischen LOM-Datensätzen gespeichert werden. Hierfür muss der Kurseditor transparent für Autoren und Dozenten das Erstellen von Relationen und neuen Medienbausteinen, die durch Kombination vorhandener Medienbausteine entstehen, durchführen. Der entstandene LOM-basierte Kurseditor für k-MED verbirgt dieses alles unter einer graphischen Oberfläche, deren Benutzung sich intuitiv erschließt.

Innerhalb des k-MED-Rahmensystems gehört der Kurseditor zusammen mit dem ConceptSpace-Editor[SRHF00] und einer Suchschnittstelle zu den Autorenwerkzeugen der Dozenten. ConceptSpace-Editor und Suchschnittstelle dienen als Quelle für modularisierte Lernressourcen, die per Drag-and-Drop in den Kurseditor hineingezogen werden können. Erstellte Kurse werden in einer Objekthierarchie mit gerichteten Relationen zwischen LOM-Datensätzen in einer relationalen Datenbank gespeichert.

Im vorliegenden Papier werden in Abschnitt 2 notwendige Grundlagen und Erweiterungen zur Erstellung und Speicherung von strukturierten Kursen bestehend aus Medienbausteinen, die mit LOM attribuiert sind, beschrieben. In Abschnitt 3 werden Möglichkeiten diskutiert, nach denen die in LOM-Strukturen gespeicherten Kurse adaptiert werden können. Anschließend wird in Abschnitt 4 ein Kurseditor, der auf den in Abschnitt 2 genannten Konzepten basiert, vorgestellt. Mit Abschnitt 5 schließt das Papier mit einer Zusammenfassung und einem Ausblick ab.

## 2. Kursstrukturen mit LOM

In diesem Abschnitt soll die Abbildung von Kursstrukturen mit LOM gezeigt werden. Hierfür und für die in Abschnitt 4 folgende Implementierung werden zunächst alle wichtigen Datenfelder, die im LOM-Draft definiert sind, vorgestellt. Anschließend wird die Abbildung der Kursstrukturen mit LOM beschrieben. Hierfür wird die Menge der möglichen Aggregationsstufen der einzelnen Bestandteile des Kurses, die im LOM-Draft definiert wurden, erweitert.

### 2.1 Benötigte Datenelemente

LOM bietet Datenelemente und eine dazugehörige Strukturierung der Datenelemente, mit denen ein Lernobjekt beschrieben werden kann. In diesem Abschnitt sollen die LOM-Datenelemente, die für den Kurseditor und sein funktionelles Umfeld relevant sind, besprochen werden. Die Schreibweise 'LOM.aaa.bbb[.bbb]' gibt den Bezeichner der LOM-Kategorie(aaa) und das Datenelement(bbb) an.

In LOM-Datensätzen sind Datenfelder zur Speicherung von Identifikatoren für die weltweit eindeutige Identifikation von Medienbausteinen und LOM-Datensätzen definiert worden. Hierbei übernimmt *LOM.General.Indentifier* die Identifikation des durch einen LOM-Datensatz beschriebenen Medienbausteins. Über *LOM.MetaMetadata.Indentifier* können die einzelnen LOM-Datensätze weltweit eindeutig identifiziert werden. Zur Speicherung des Titels eines Medienbausteins ist das Datenfeld *LOM.General.Title* definiert worden. Dieses Datenfeld eignet sich besonders zur Anzeige des Titels im Kurseditor, ohne die Medienbausteine öffnen zu müssen. Angaben über die Granularität von Medienbausteinen werden in *LOM.General.-*

*AggregationLevel* gespeichert. Im LOM-Draft ist hierfür ein Vokabular mit den Elementen '1', '2', '3', '4' definiert worden, das Aggregationsstufen vom kleinsten hin zu größeren Aggregationsstufen festlegt. Für die Speicherung des MIME-Types eines Medienbausteins ist das Feld *LOM.Technical.Format* definiert worden. Dieses Feld kann zur Assoziation von Icons verwendet werden, um dem Benutzer visuell kodierte Informationen zu dem MIME-Type eines Medienbausteins anzubieten.

Eine wichtige Rolle bei der Abbildung von Kursstrukturen auf LOM-Objekte kommt der Kategorie *LOM.Relation* zu. Sie ermöglicht das Formulieren von gerichteten Relationen ausgehend von dem LOM-Datensatz, indem sie formuliert sind, zu einem Ziel-LOM-Datensatz. Im Datenfeld *LOM.Relation.Kind* wird der Typ einer Relation gespeichert. Im LOM-Draft sind hierfür die folgenden Typen definiert worden: 'IsPartOf', 'HasPart', 'IsVersionOf', 'HasVersion', 'IsFormatOf', 'HasFormat', 'References', 'IsReferencedBy', 'IsBasedOn', 'IsBasisFor', 'Requires', 'IsRequiredBy'. Diese Relationstypen basieren auf Dublin Core [WKLW98] und liegen immer paarweise vor, damit durch zwei entgegengesetzte unidirektionale Relationen eine bidirektionale Relation zwischen zwei LOM-Datensätzen realisiert werden kann. Diese Wertemenge wird um die in [SSFS99] auf Medienbausteinen übertragenen rhetorisch-didaktischen Relationen erweitert. In *LOM.Relation.Resource-Identifier* wird der Identifikator des LOM-Datensatzes des Ziels der Relation gespeichert. Die Wertemenge dieses Feldes setzt sich aus den Identifikatoren aller LOM-Datensätze zusammen.

## 2.2 Abbildung von Kursstrukturen in LOM

Über Relationen, die mit Objekten der Kategorie *LOM.Relation* der LOM-Datensätze formuliert werden können, werden die Medienbausteine eines Kurses miteinander verbunden. Hierfür werden gerichtete unidirektionale Relationen verwendet. Durch sie werden die einzelnen Bestandteile des Kurses miteinander zu einem Baum verbunden. Der Wurzelknoten und alle Wurzelknoten der Teilbäume des Kursbaums bilden durch die Zusammenfassung von Medienbausteinen neue Medienbausteine. Wir verwenden somit eine rekursive Definition der Kursstruktur. Zusammengesetzte Medienbausteine referenzieren ihre Teilmodule über Relationen vom Typ 'HasPart'. In entgegengesetzter Richtung referenzieren die Teilmodule ihre Vatermodule über Relationen vom Typ 'IsPartOf'. Die hierfür notwendigen Relationen vom Typ 'HasPart' und 'IsPartOf' erstellt der Kurseditor transparent für den Benutzer. Die Abbildung der baumförmigen Kursstruktur beruht demnach auf einer sukzessiven Zusammenfassung von Medienbausteinen durch Referenzierung vom Typ 'HasPart' von LOM-Datensätzen durch LOM-Datensätze.

Neben den 'HasPart'- und 'IsPartOf'-Relationen kann der Kurseditor zusätzlich Relationen vom Typ 'References' inklusive der Umkehrrelation 'IsReferencedBy' anlegen, mit denen Querverweise zu Medienbausteinen innerhalb und außerhalb der baumförmigen Kursstruktur hergestellt werden können. Eine Konkatenation mehrerer 'References'-Relationen ist derzeit nicht vorgesehen, um eine klare Abgrenzung von beliebigem Hypertext zu erreichen.

Für die Auswertung der Kursstruktur und der Querverweise, die durch die Relationen der Typen 'HasPart' und 'References' in den LOM-Datensätzen abgebildet wird, stellt deren Ordnung untereinander eine wichtige Rolle dar. Durch die Reihenfolge dieser Relationen wird die Abfolge der Medienbausteine und zusätzlich die Zugehörigkeit der Querverweise zu Medienbausteinen bestimmt. Aus diesem Grund muss die Reihenfolge entgegen den Empfehlungen des LOM-Drafts zumindest für die Relationen der Typen 'HasPart' und 'References' gewährleistet sein.

## 2.3 Erweiterung der Aggregationsstufen

Die Menge der Knoten innerhalb des Kursbaums lässt sich in die inneren Knoten und die Blattknoten teilen. Hierbei bildet die Menge der Blattknoten LOM-Datensätze, die Medienbausteine beschreiben, an die physikalische Daten geknüpft sind. Die Menge der inneren Knoten des Kursbaums enthält Medienbausteine, die nicht direkt physikalische Daten beschreiben. Sie sind je nach der Tiefe im Baum eine sukzessive Zusammenfassung von Medienbausteinen. Die Menge der inneren Knoten lässt sich weiterhin in die Menge von Knoten unterteilen, die Medienbausteine zusammenfassen, ohne dabei eine inhaltliche Untergliederung einzuführen, und der Menge von Knoten, die Medienbausteine zusammenfassen, mit dem Ziel eine inhaltliche Untergliederung zu schaffen.

Damit ein LOM-Datensatz einer der genannten Klassen zugeordnet werden kann, muss dieser gekennzeichnet werden. Für diesen Zweck wird im LOM-Draft das Datenfeld LOM.General.AggregationLevel verwendet. Die definierte Wertemenge für dieses Feld ist nicht für das oben erwähnte Klassifizierungsschema geeignet. Aus diesem Grund haben wir für das k-MED-Rahmensystem die folgende Wertemenge definiert:

**Atom:** LOM-Datensätze, die durch die Aggregationsstufe 'Atom' gekennzeichnet sind attribuieren physikalische Daten, die über einen URI referenziert werden. Die physikalischen Daten bilden inhaltlich eine Sinneinheit und liegen in einem Format vor, das entweder direkt durch einen Web-Browser oder indirekt über Plug-Ins verarbeitet werden kann. Die physikalischen Daten, die durch diesen LOM-Datensatz beschrieben werden, liegen in einer Datei vor und sind in den meisten Fällen monomedial. Diese Art LOM-Datensätze haben keine Referenzen des Typs 'HasPart' zu anderen Medienbausteinen.

**Subatom:** Dieser Typ von LOM-Datensätzen ist eingeführt worden, da die Bedingung, dass LOM-Datensätze des Typs 'Atom' Ressourcen beschreiben, die inhaltlich eine Sinneinheit bilden und aus nur einer Datei bestehen, nicht in allen Fällen erfüllt ist. Mit LOM-Datensätzen des Typs 'Subatom' werden demnach Ressourcen beschrieben, die aus einer Datei bestehen, jedoch keine inhaltliche Sinneinheit bilden. LOM-Datensätze dieses Typs werden für die Benutzer des k-MED-Rahmensystems transparent verwaltet, erfahren aber ansonsten die gleiche Behandlung wie LOM-Datensätze des Typs 'Atom'.

**CollectionOfSubatoms:** Dieser Typ von LOM-Datensätzen ist eingeführt worden, um LOM-Datensätze vom Typ 'Subatom' zu Sinneinheiten zusammenzufassen. Wobei durch die Zusammenfassung keine inhaltliche Untergliederung definiert ist. Durch diese LOM-Datensätze werden keine physikalischen Daten direkt beschrieben, sondern die Zusammenfassung von subatomaren Medienbausteinen über Relationen vom Typ 'HasPart'. Die Menge der Typen von LOM-Datensätzen, die das Ziel einer Relation vom Typ 'HasPart' sind, ist auf 'Subatom' begrenzt. Das Hinzufügen, Entfernen oder Vertauschen von Relationen zu subatomaren LOM-Datensätze ist bei bereits bestehenden LOM-Datensätzen dieses Typs nicht mehr erlaubt. Vom Kurseditor werden diese LOM-Datensätze wie LOM-Datensätze des Typs 'Atom' behandelt.

**CollectionOfAtoms:** Dieser Typ ist zur Zusammenstellung größerer Sinneinheiten ohne inhaltliche Untergliederung eingeführt worden. LOM-Datensätze dieses Typs können über Relationen vom Typ 'HasPart' LOM-Datensätze der Typen 'Atom', 'CollectionOfSubatoms', 'CollectionOfAtoms' referenzieren. Das Hinzufügen, Entfernen oder Vertauschen von Relationen vom Typ 'HasPart' ist erlaubt. Die Gruppierung der Medienbausteine in einer

CollectionOfAtoms wird nur für Autoren und Dozenten sichtbar, für Lernende sind sie transparent.

**Chapter:** Dieser Typ ist zur Zusammenstellung größerer Sinneinheiten mit inhaltlicher Untergliederung eingeführt worden. LOM-Datensätze dieses Typs können über Relationen vom Typ 'HasPart' LOM-Datensätze der Typen 'Atom', 'CollectionOfSubatoms', 'CollectionOfAtoms' und 'Chapter' referenzieren. Eine hierarchische Struktur wird durch die Möglichkeit der Referenzierung von Chapters in Chapters rekursiv definiert. Das Bilden von Zyklen bestehend aus Chapters muss allerdings vermieden werden.

**Course:** Aus technischen Gründen wurde im k-MED-Rahmensystem dieser Typ von LOM-Datensätzen eingeführt. LOM-Datensätze dieses Typs unterscheiden sich nicht von LOM-Datensätzen des Typs 'Chapter' außer dem technischen Unterschied, dass ein Kurs eine abgeschlossene Einheit bildet.

Der Mehrwert der konsequenten Nutzung von LOM zur Erstellung von Kursen liegt nicht nur in der besseren Wiederverwendbarkeit von Medienbausteinen, sondern auch in der Möglichkeit der Unterstützung von Adaptivität in den erstellten Kursen auf mehreren Ebenen. Das bezieht sich insbesondere auf das Entfernen, Tauschen oder Ergänzen von einzelnen Medienbausteinen innerhalb der Kurse.

### 3. Adaptivität von Kursen mit LOM

Die Speicherung der Kurse mit LOM-Datensätzen erfolgt ohne Angabe der Optik und des Layouts der Präsentation dieser Kurse. Der Kursautor legt lediglich die Reihenfolge und Struktur der Medienbausteine fest. Daraus folgt, dass für die Generierung von Präsentationen dieser Kurse Vorlagen benötigt werden, nach denen die Kurse mit einem Layout versehen werden können. Diese Vorlagen bestehen aus mikro- und makrotypographischen Vorgaben, die teilweise durch Designer bestimmt oder automatisch an die Lernenden adaptiert werden können. Bei den Teilen der Vorlagen, die durch Designer festgelegt werden, ist die Auswahl aus einer Menge durch den Benutzer denkbar, sodass die Kurse durch den Lernenden adaptierbar sind.

Bei der automatischen Adaption der Kurse an die Lernenden sind besonders die rhetorisch-didaktischen Relationen zwischen den Medienbausteinen hilfreich. Sie drücken inhaltliche Beziehungen zwischen Medienbausteinen aus. In [SSFS99] sind hierfür die Relationen 'example', 'illustrates', 'instance', 'restricts', 'amplifies', 'continues', 'deepens', 'opposite' und 'alternative' definiert worden. Mit ihnen ist es möglich, inhaltliche Beziehungen zwischen Medienbausteinen auszudrücken, die ggf. auch kursübergreifend angelegt sein können (Abbildung 1).

Die Vorlagen für die Generierung von Präsentationen eines Kurses kann Regeln enthalten, die die rhetorisch-didaktischen Relationen in Abhängigkeit des Wissens und den Wünschen der Lernenden auswertet. Die Relation vom Typ 'alternative' zwischen den Medienbausteinen A und D kann herangezogen werden, wenn statt des Videos in Medienbaustein A besser das Bild in Medienbaustein D in der Präsentation gezeigt wird, da der Lernende nur über ein Modem mit dem Internet verbunden ist. Die Beziehung zwischen Medienbaustein A und B kann Auswirkungen auf das makrotypographische Layout der Präsentation haben. In Abhängigkeit zum Wissensstand des Lernenden kann möglicherweise das Einbinden des Beispiels in Medienbaustein E in die Präsentation sinnvoll sein, währenddessen Medienbaustein F mit entgegengesetzter Aussage besser nicht in die Präsentation eingebettet wird. Möchte der Lernende lediglich einen Überblick über ein Thema bekommen, ist es sicherlich sinnvoll den

vertiefenden Medienbaustein C ebenfalls nicht in die Präsentation einzubetten. Weitere Möglichkeiten lassen sich aus den rhetorisch-didaktischen Relationen ableiten, die in dem Beispiel nicht genannt wurden.

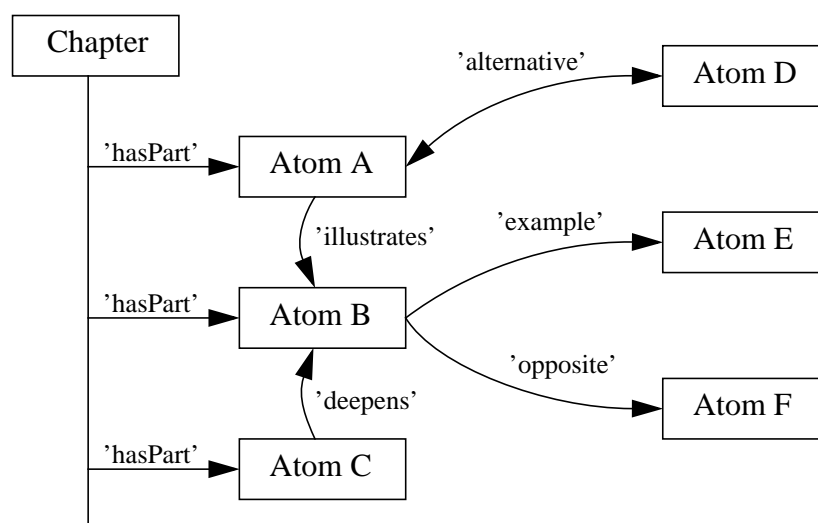


Abbildung 1: Rhetorisch-didaktische Relationen zwischen Medienbausteinen

## 4. Implementierung

Basierend auf den LOM-Draft (Version 6.1) haben wir ein Datenmodell in Java umgesetzt, das als Grundlage für alle Programmkomponenten dient, die das Metadatenschema verwenden. Dieses Java-Datenmodell wurde so gestaltet, dass es mit Hilfe von Castor [ELG01] auf verschiedene relationale Datenbanken abgebildet werden kann. Castor ist ein Open-Source Framework, um Java-Objekte auf XML-Dokumente oder mittels SQL-Schnittstelle auf Datenbanken abzubilden.

Durch den selektiven Zugriff auf bestehende Medienbausteine innerhalb der Datenbank mit Hilfe einer Suchschnittstelle kann eine Neugruppierung und Verwaltung der Medienbausteine im Kurseditor durchgeführt werden. Nach erfolgter Bearbeitung der Medienbausteine werden die zugehörigen Metadaten in der Datenbank hinterlegt. Insbesondere geschieht dies mit den aus der Bearbeitung resultierenden Metadaten der Medienbausteine, die durch Neugruppierung gewonnen wurden.

### 4.1 Suchschnittstelle

Zur Unterstützung des Anwenders bei der Formulierung von Suchanfragen und bei der Darstellung der Suchergebnisse wird eine Suchschnittstelle angeboten. Die Bedienung der Suchschnittstelle erfolgt in den Arbeitsschritten der Formulierung der Suchanfrage, des Startens der Suche und der Anzeige der Suchergebnisse. Die aus der Datenbank selektierten Suchergebnisse werden graphisch separiert von der aktuellen Darstellung des in der Editierung befindlichen Kurses in einer Baumstruktur dargestellt und bilden zusammen mit den bis dahin editierten Kursteilen das jeweils aktuelle Repository für die Bearbeitung des aktuellen Kurses im Kurseditor. Durch wiederholte Suchanfragen kann daher das aktuell angezeigte Repository verändert und entsprechend Benutzerwunsch angepasst werden.

Die Baumstruktur der Suchergebnisse ist entsprechend der Medientypen der dargestellten Medienbausteine strukturiert. Der MIME-Type des Medienbausteins wird durch ein vorangestelltes Icon vor dem Titel des Medienbausteins angezeigt. Auf Wunsch der Benutzer

kann die Anzeige des zum Medienbaustein gehörigen LOM-Datensatzes durch Aufruf des Metadateneditors im separaten Fenster erfolgen.

## **4.2 Darstellung des Kurses**

In der von uns gewählten Darstellung der Kursstruktur der Medienbausteine des zu editierenden Kurses als Baum muss beachtet werden, dass sich die Reihenfolge der Medienbausteine an der linearisierten Kursstruktur orientiert. Die Ordnung in vertikaler Richtung der Darstellung im Kurseditor entspricht demnach der geplanten linearen Bearbeitungsreihenfolge der Medienbausteine des Kurses zum Präsentationszeitpunkt ohne Traversierung von angebotenen Referenzen.

Medienbausteine, die Gruppierungselemente bilden und keine inhaltlichen Mediendaten enthalten, sind innere Knoten des Kursbaumes oberhalb der Blätter und werden im gleichen Baum mit Objekten, die inhaltliche Mediendaten enthalten und als Blätter des Kursbaumes angezeigt werden, gemeinsam dargestellt. Ausnahme bildet hierbei die Anzeige von `CollectionOfSubatoms`, wenn die Darstellung der Subatoms unterdrückt wird.

Die Unterscheidung des Typs der Medienbausteine erfolgt wie in der Anzeige der Suchergebnisse durch ein dem Titel des Medienbausteins vorangestelltes Icon.

## **4.3 Bearbeitung durch Drag-and-Drop**

Neben den Bearbeitungsformen des Markierens mit anschließendem Kopieren und Einfügen durch Tastaturbedienung, menügesteuerter Bedienung oder Knopfleistenbedienbarkeit ist in unserer Implementierung besonders die Bearbeitung der Kursstruktur durch mausgestütztes Drag-and-Drop als intuitive und schnelle Bearbeitungsvariante der Kursstruktur hervorgehoben worden.

Die Medienbausteine können per Drag-and-Drop aus der Anzeige der Suchergebnisse in die Darstellung der Kursstruktur hineingezogen und an der entsprechenden Stelle des Kurses platziert werden. Handelt es sich um Medienbausteine, die einer Gruppierung weiterer Medienbausteine dienen, so werden darin enthaltene Medienbausteine in ihren Strukturinformationen ausgewertet und mit in die Kursstruktur übernommen und traversierbar und editierbar angezeigt.

Die zweite unterstützte Drag-and-Drop-Funktionalität ergibt sich aus der Verschiebbarkeit der Medienbausteine innerhalb der aktuell editierten Kursstruktur. Dabei wurde eine von gängigen Editoren und Objektnavigationen abweichende Implementierung der Drag-and-Drop-Funktionalität erarbeitet. Dies resultiert zum einen in der Anforderung der Anzeige von Gruppierungselementen und Medieninhaltsobjekten in einem einzigen Fenster, das eine Übernahme der Drag-and-Drop-Bedienweise von File-System-Navigatoren wie etwa dem Windows Explorer widerspricht, zum anderen in der Anforderung einer schnellen und intuitiven Bedienung, die gerade durch Drag-and-Drop unterstützt werden soll. So ist es etwa im Netscape Communicator Bookmark Editor nicht möglich, in einem einzigen Bediensschritt mittels Drag-and-Drop bei aufgeklappter Baumstruktur zu entscheiden, ob ein ausgewähltes Bauelement an das Ordnerende oder direkt nach dem Ordnerende auf einer beliebigen höherer Strukturebene, aber vor dem nächsten unter dem Ordnerende angezeigten Baumobjekt eingefügt werden soll.

Die eindimensionale vertikale Editierung reicht daher für eine Verschiebung per Drag-and-Drop in einem Bearbeitungsschritt nicht aus. Um dies zu ermöglichen, muss in zweiter Dimension in horizontaler Richtung die Einfügetiefe des per Drag-and-Drop zu verschiebenden Medienbausteins im Kurseditor ausgewertet und graphisch rückgemeldet werden (siehe Abbildung 2).

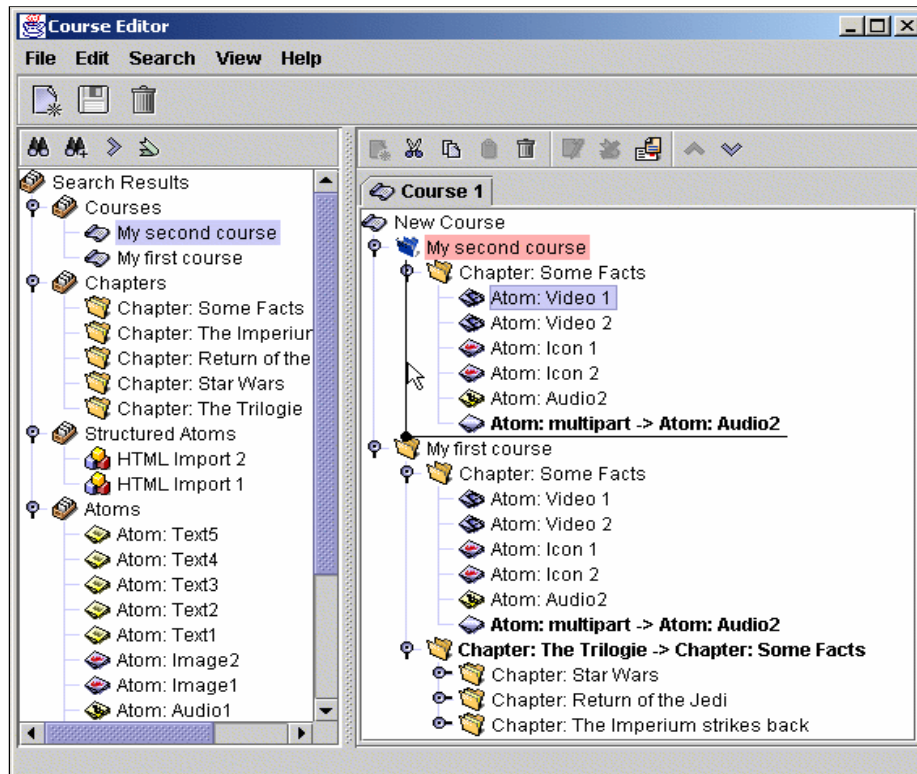


Abbildung 2: 2-D Einfügemarke bei Drag-and-Drop

#### 4.4 Hinterlegen der Metadaten in Datenbank

Kursbearbeitungsstände können als LOM-Datensätze in einer Datenbank hinterlegt werden. Dabei werden für neu generierte Gruppierungselemente jeweils eigene LOM-Datensätze erzeugt, die keine Verweise auf physikalische Daten sondern auf weitere LOM-Datensätze enthalten.

#### 5. Schlussfolgerung und Ausblick

In dem vorliegenden Papier haben wir einen Kurseditor zur Erstellung von adaptierbaren Kursen, die auf modularisierten Lernressourcen basieren, vorgestellt. Zunächst haben wir hierfür das zugrundeliegende Datenmodell LOM beschrieben. Dieses wird sowohl als Eingabeformat in Form von Metadatenbeschreibungen für Medienbausteine als auch als Ausgabeformat für die erstellten Kurse verwendet. Hierfür wird die Hierarchie der Kursstruktur auf einen Baum von LOM-Datensätzen unter Verwendung unserer vorgestellten Erweiterungen abgebildet. Durch die modular abgespeicherte Kursstruktur ergeben sich nicht nur Vorteile bei der späteren Aktualisierung bereits bestehender Kurse, sondern auch verschiedene Möglichkeiten der Adaption dieser Kurse. Diese haben wir anhand eines Beispiels erläutert. Anschließend haben wir die Applikation des Kurseditors vorgestellt. Hier haben wir insbesondere den Workflow bei der Erstellung von Kursen, die intuitiv zu bedienende Drag-and-Drop-fähige Oberfläche und das Speichern der Kurse in der LOM-Datenbank herausgehoben.

Erweiterungen für den vorgestellten Kurseditor sehen wir insbesondere in der Pflege der Wissensbasis. Hier ist es notwendig identische LOM-Datensätze zu finden und zu löschen oder unvollständige LOM-Datensätze zu vermeiden. Zusätzlich muss darauf geachtet werden, dass keine Zyklen durch Chapter und CollectionOfAtoms in der Datenbank entstehen. Ein weiteres



Thema für Erweiterungen stellt eine Abfragemöglichkeit der Wissensbasis durch Beschreibungslogik dar. Im Laufe des Projektes wird der Kurseditor durch die Möglichkeit der Generierung von Präsentationen der erstellten Kurse erweitert und von Instruktionsdesignern evaluiert.

## Literatur

- [Bro01] Patria Brown, *Java Look and Feel Design Guidelines*, second edition, <http://java.sun.com/products/jlf/ed2/book/index.html>, Sun Microsystems, 2001
- [ELG01] ExoLab Group, *Castor*, <http://castor.exolab.org>, 2001
- [LWG01] LOM working group, IEEE P1484.12/D6.1, IEEE Learning Technology Standards Committee, *Draft Standard for Learning Objects Metadata*, <http://ltsc.ieee.org/wg12/index.html>
- [See01] C.Seeberg, *Modulare Wissensbasen zur Erzeugung adaptiver und kohärenter Lehrdokumente*, Dissertationsschrift, Technische Universität Darmstadt, 2001
- [SRHF00] C.Seeberg, I.Rimac, S.Hörmann, A.Faatz, A.Steinacker, A.El Saddik, R.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), pages 96-105, 2000
- [SSFS99] A.Steinacker, C.Seeberg, S.Fischer, R.Steinmetz, *MultiBook: Meta-data for Webbased Learning Systems*, Fachgebiet Industrielle Prozess- und Systemkommunikation, in Proceedings of the 2nd International Conference on New Learning Technologies, 1999
- [WKLW98] S.Weibel, J.Kunze, C.Lagoze, M.Wolf, *Dublin Core Metadata for Resource Discovery*, <http://purl.org/dc>, RFC2413, 1998