

Bausteine eines durchgängigen Workflows für die Inhaltserstellung in e-Learning-Systemen

Martin Memmel

Universität Kaiserslautern, Fachbereich Informatik
AG Algorithmisches Lernen
Postfach 3049, 67653 Kaiserslautern, Germany
Email: memmel@informatik.uni-kl.de

Zusammenfassung

Die Anforderungen an Inhalte, die in e-Learning-Systemen zur Verfügung gestellt werden, unterscheiden sich stark von den Anforderungen an klassische Lehrmaterialien. Sie müssen u. a. mediengerecht aufgearbeitet sein, d. h. sie müssen auf dem Bildschirm in adäquater Form dargestellt werden, sie müssen Adaptivität bzw. Adaptierbarkeit ermöglichen, und die Einbindung multimedialer Lernelemente muß möglich sein. Diese und andere Aspekte machen die Eingabe von Lernelementen im Kontext des e-Learning zu einem sehr komplexen Prozeß, bei dem die Angabe zahlreicher Metadaten vonnöten ist – die Inhalte müssen demnach gewissen syntaktischen Ansprüchen genügen.

Autoren benötigen beim Erstellen von Inhalten also eine adäquate Werkzeugunterstützung, die zum einen eine möglichst einfache und intuitive Eingabe ermöglicht und zum anderen die Syntax überprüft und dem Autor entsprechende Rückmeldungen liefert.

Schließlich müssen die Inhalte auch noch in das e-Learning-System integriert werden können, unter Berücksichtigung der im entsprechenden e-Learning-System implementierten Rechte- und Rollenkonzepte.

Alle diese Bausteine eines Workflows für die Inhaltserstellung in e-Learning-Systemen sowie die zugehörigen Schnittstellen müssen hohen Ansprüchen genügen, insbesondere dann, wenn viele Inhaltsersteller in einem System arbeiten und wenn der Aspekt der Wiederverwendbarkeit von Lernelementen beachtet werden soll.

Im folgenden werden nach einer kurzen Einführung über Lerninhalte im Kontext des e-Learning die Bausteine im Einzelnen vorgestellt und am Beispiel des DaMiT-Systems konkret beleuchtet.

1 Lerninhalte im e-Learning

Der Einsatz neuer Medien und Technologien – eines der Hauptziele im e-Learning und ein großer Vorteil gegenüber herkömmlichem Lernen – sowie die Erweiterung und Verbesserung herkömmlicher Lernformen erfordern besondere didaktische Konzeptionen.

Besonders im e-Learning gilt es, das Verhalten des Lernenden zu antizipieren und entsprechende Strukturen zur Bewältigung dieser Aufgabe zu schaffen. Dabei müssen eine Vielzahl von Aspekten berücksichtigt werden, u. a.:

- Organisation
- Ablauf
- benötigte Hilfen
- Sozialformen

Inhalte müssen und sollen nicht einfach nur sequentiell wie in einem Lehrbuch aneinandergereiht werden. Das Angeben von empfohlenen Pfaden, möglichen Verzweigungen und Beziehungen zwischen den Lerninhalten (z. B. Abhängigkeiten, Verwandtschaften) ermöglicht dem Benutzer, die Inhalte in der von ihm bevorzugten Reihenfolge zu bearbeiten.

Des weiteren sind Lernziele von erheblicher Bedeutung. Ihre Kontrolle kann mit interaktiven Übungen, Praktika und Tests (sowohl tutoriell als auch kollaborativ und kooperativ) erfolgen.

Ein weiterer großer Vorteil des e-Learning im Vergleich zu konventionellem Lehren und Lernen liegt in der möglichen Adaptivität bzw. Adaptierbarkeit eines Systems (siehe [Spe99]). Ein traditioneller *one-size-fits-all*-Ansatz wird den Anforderungen in den meisten Fällen nicht gerecht. Individuelle Erfahrungen und Lernziele müssen berücksichtigt werden, sowohl bei funktionaler als auch bei inhaltlicher Adaptivität.

Ein geeignetes Mittel, um all diesen Anforderungen gerecht zu werden, sind sogenannte Storyboards (siehe [TTR02] oder [TD01]) wie in der folgenden Abbildung.

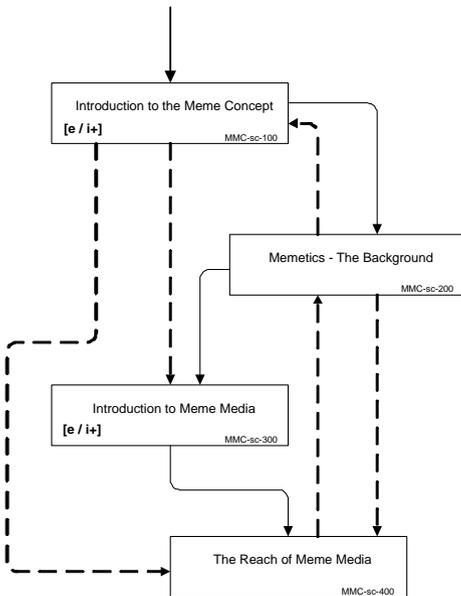


Abbildung 1: Ein Storyboard

Ein Storyboard ist ein hierarchischer kanten- und knotenbewerteter Graph, die Knoten sind einzelne Szenen, die in verschiedenen Varianten vorliegen und ihrerseits wieder aus Teilszenen bestehen können. Verschiedene Typen von Kanten drücken verschiedene Arten der Navigation zwischen Szenen aus.

2 Bausteine der Inhaltserstellung

Der in Abbildung 2 dargestellte Workflow für die Inhaltserstellung in e-Learning-Systemen besteht aus vier großen Bausteinen: Konzeption, Eingabe, Verarbeitung und Import. Diese Bausteine werden nun im Einzelnen vorgestellt.

2.1 Konzeption

Natürlich muß sich jeder Autor zunächst genau überlegen, welche Inhalte er in welcher Form erstel-



Abbildung 2: Die vier Bausteine der Inhaltserstellung

len will. Dies ist – wie bereits im vorigen Abschnitt über Lerninhalte im e-Learning deutlich wurde – ein sehr komplexer und aufwendiger Prozeß. Die Konzeption hängt zudem nicht nur von den Vorstellungen des Autors, sondern auch von den vom System zur Verfügung gestellten Möglichkeiten ab. Autoren müssen also bereits in dieser Phase recht genaue Kenntnisse über das System besitzen und dürfen nicht erst bei der Eingabe der Inhalte auf eventuelle Einschränkungen – oder Möglichkeiten – aufmerksam werden. Daraus ergibt sich automatisch die Forderung, daß eine Werkzeugunterstützung im Prozeß der Inhaltserstellung so früh wie möglich – nämlich bereits bei der Konzeption der Inhalte – vorhanden sein sollte. Idealerweise kann der Autor mit Hilfe dieses Werkzeugs ein komplettes Storyboard entwerfen und sukzessive mit konkreten Inhalten füllen.

2.2 Eingabe

Sind die Inhalte konzipiert, müssen sie in einem bestimmten Eingabeformat eingegeben werden. Dieses Format kann sich durchaus von dem letztendlich im System verwendeten Format unterscheiden. Bei der Eingabe können bzw. müssen mit Hilfe von Metadaten zahlreiche Informationen über die Inhal-

te sowie verschiedenste Beziehungen zwischen den Lernelementen angegeben werden. Außerdem ist in vielen Fällen die Verknüpfung zu externen Elementen wie Filmen, Audio-Files oder interaktiven Illustrationen nötig.

Es handelt sich somit auch hier um einen nicht-trivialen Prozeß, bei dem die Autoren mit Hilfe entsprechender Werkzeuge zur Vereinfachung der Eingabe unterstützt werden müssen.

2.3 Verarbeitung

Hat der Autor Inhalte eingegeben, muß evtl. zunächst die syntaktische Korrektheit der Eingabe validiert werden. Zudem kann die Überprüfung verwendeter Bezeichner (IDs) vonnöten sein, wozu unter Umständen eine Verbindung zum e-Learning-System vorhanden sein muß.

Bei positivem Ergebnis der Validierung erfolgt in den meisten Fällen die Übersetzung der Eingabe in das vom System benötigte, endgültige Inhaltsformat.

2.4 Import

Zu guter Letzt müssen die Inhalte in das e-Learning-System importiert werden können. Dieses Importieren besteht aus dem Einspielen und Integrieren der Inhalte. Hierbei muß besonders die Frage geklärt sein, *wer* Inhalte *an welcher Stelle* in das System integrieren kann. Dies hängt natürlich von dem im entsprechenden e-Learning-System implementierten Rechte- und Rollenkonzept ab.

3 Das DaMiT-System

DaMiT¹ ist ein im Rahmen des BMBF-Programms "Neue Medien in der Bildung"² gefördertes Projekt. Das Konsortium besteht aus zehn Universitäten und Fachhochschulen aus zehn Bundesländern.

DaMiT steht für *Data Mining Tutor* und ist ein generisches Lernsystem, dessen Inhalte auf das Gebiet des Data-Mining ausgerichtet sind. Es werden sowohl Grundlagen des Data-Mining als auch komplexe Verfahren behandelt – die Benutzer haben die Möglichkeit, nicht nur theoretische Hintergründe, sondern auch praktische Fähigkeiten zu erlernen.

Das DaMiT-System zeichnet sich u. a. durch folgende Besonderheiten aus (vgl. [GLM03]):

Komplexe Übungsaufgaben: Beim Data-Mining werden relativ kleine, abgeschlossene Übungsaufgaben und Tests wie Multiple-Choice-Questions und Fill-In-Blank-Questions den Erfordernissen in vielen Fällen nicht gerecht. Im DaMiT-System findet sich deshalb neben den klassischen Übungsaufgaben ein auf die speziellen Anforderungen zugeschnittener Aufgabentyp, die sogenannte „Competitive Exercise“ (siehe [SD02]).

Multimediale Inhalte: Einer der wesentlichen Vorteile des e-Learning im Vergleich zu konventionellen Lehrmaterialien besteht in der Möglichkeit, multimediale Inhalte einzubinden. In DaMiT gibt es beispielsweise Filme, Audio-Files, interaktive Illustrationen etc.

Rollenkonzept: Im DaMiT-System kann man sich als Benutzer in verschiedenen Rollen registrieren, und jede Rolle bedeutet eine andere Sicht auf die zur Verfügung stehenden Inhalte und Funktionalitäten. Im DaMiT-System gibt es eine Vielzahl solcher Rollen, z.B. *anonymous*, *standard learner*, *content provider*, *teacher* und *admin* (siehe [SBHG02]).

Payment: Einige Inhalte im System sind kostenpflichtig, beispielsweise müssen Benutzer für Expertenurteile über kommerzielle Data-Mining-Systeme einen bestimmten Beitrag zahlen. Die Kosten für ein Lernelement sind jedoch nicht statisch, sondern hängen z. B. von der Benutzerrolle ab.

Adaptivität/Adaptierbarkeit: Um individuellen Lernprozessen gerecht zu werden, müssen verschiedene Sichten auf Inhalte ermöglicht werden. Adaptivität bzw. Adaptierbarkeit kann im DaMiT-System auf verschiedene Weise ermöglicht werden. So gibt es verschiedene Schwierigkeits- und Präsentationsniveaus, und durch den Einsatz diverser Filter kann zudem eine Auswahl bestimmter Typen von Lernelementen angezeigt bzw. ausgeblendet werden.

Wiederverwendbarkeit: Nicht nur bei der Ermöglichung verschiedener Sichten auf Inhalte spielt die Wiederverwendbarkeit von Lernelementen eine große Rolle. Sie erleichtert auch den Autoren die Arbeit und hilft beim Einhalten einheitlicher Notationen.

¹<http://damit.dfki.de/>

²<http://www.medien-bildung.net/>

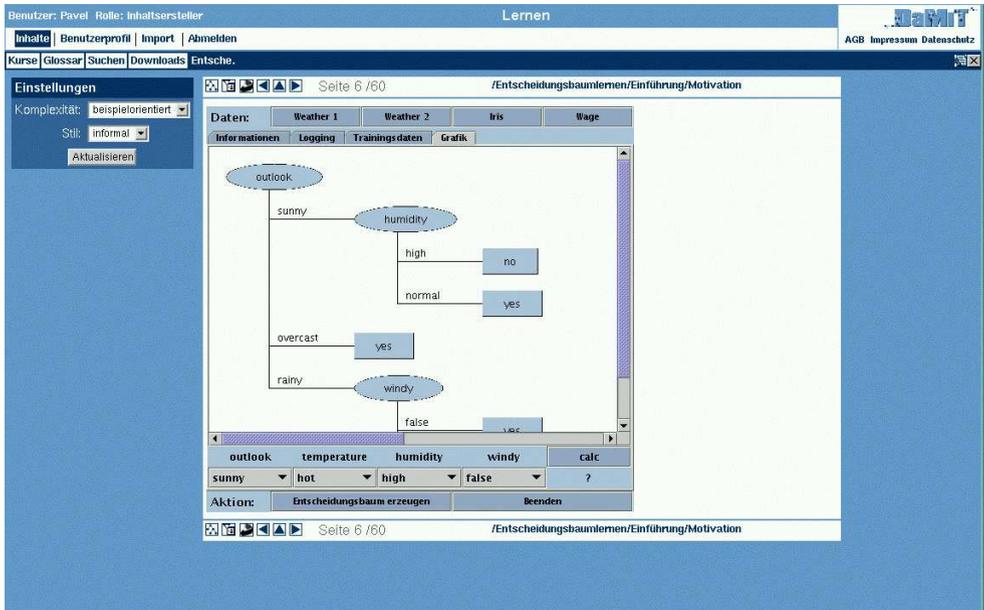


Abbildung 3: Ein Screenshot aus dem DaMiT-Lernsystem

Abbildung 3 zeigt einen Screenshot aus dem DaMiT-Lernsystem, in dem ein interaktives Applet zum Erzeugen von Entscheidungsbäumen angezeigt wird.

Neben der Umsetzung der genannten Besonderheiten wurde bei der Entwicklung des DaMiT-Systems wurde zudem großer Wert auf folgende Aspekte gelegt:

Einhalten von Standards: Durch die durchgehende Verwendung von Standards wie MVC, XML, XML Schema, IMS, SAX usw. ist eine Anbindung neuer Module jederzeit möglich. Gerade bei der Inhaltserstellung hat dies den Vorteil, daß viele Quellen herangezogen werden können.

Verwendung von Open-Source-Produkten: Durch die weitestgehende Verwendung von Open-Source-Produkten entfallen teure Lizenzgebühren, was insbesondere im wichtigsten Einsatzgebiet von DaMiT – der akademischen Lehre – von nicht unerheblicher Bedeutung ist.

Nachhaltigkeit: Durch die Verwendung aktuellster Techniken wie XML, MVC, JavaBeans, Java-

Servlets, MVC usw. kann sichergestellt werden, daß DaMiT auch noch in mehreren Jahren den Ansprüchen eines modernen e-Learning-Systems genügt. Für detaillierte Informationen zum Thema „Nachhaltigkeit von e-Learning-Projekten“ sei hier auf [Deg03] verwiesen.

3.1 Das DaMiT-Inhaltsformat

Allen in Abschnitt 3 dargestellten Besonderheiten mußte bei der Systematisierung und Festlegung des DaMiT-Inhaltsformats natürlich Rechnung getragen werden.

Zur Beschreibung der Metadaten wurde auf den Standard des IMS Global Learning Consortium^{3,4,5} zurückgegriffen. Hierbei wurde einerseits auf einige Teile verzichtet, andererseits waren auch Erweiterungen nötig.

Im DaMiT-System gibt es drei Typen von Lernelementen mit folgender Bedeutung:

³<http://www.imspjproject.org/>

⁴Primer for the IMS Learner Information Package, 2001.

⁵<http://www.imsglobal.org/metadata/>

content: Das „kleinste“ Element. Es besteht aus einem Inhalt eines bestimmten Typs (z. B. HTML, SWF, JPG oder GIF), wobei in den Metadaten keinerlei semantische Informationen (also etwa Schwierigkeitsgrad, benötigtes Vorwissen etc.) angegeben werden.

unit: Ein semantisch nicht mehr auf sinnvolle Weise teilbares Lernelement eines bestimmten Typs wie etwa eine Definition, ein Satz oder eine Illustration. **unit**-Elemente bestehen aus einem oder mehreren **content**-Elementen und können in verschiedenen Präsentationsstilen definiert werden.

topic: Ein **topic**-Element entspricht einem Kapitel und besteht aus weiteren **topic** bzw. **unit**-Elementen. **topic**-Elemente können in verschiedenen Schwierigkeitsgraden angegeben werden.

Auf eine detaillierte Angabe aller zu den einzelnen Elementen gehörenden Metadaten wird hier verzichtet. Exemplarisch dient Abbildung 4 zur Veranschaulichung des DaMiT-Metadatenformats.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<topic>
<general name="KL_pattern" title="Patterns"
language="de" description="Ein Kurs über Patterns."
keyword="gl_pattern"/>
<lifecycle version="1.0" status="unavailable">
<contribute date="2002-12-5-11:20:6.000000">
<authors responsible="KL_MM">
<author name="KL_MM"/>
</authors>
</contribute>
</lifecycle>
<educational difficulty="any"/>
<rights cost="0"/>
<relation kind="has_part" newpage="true">
<resource name="KL_pattern_motivation" author="KL_MM"/>
</relation>
<relation kind="has_part" newpage="true">
<resource name="KL_pattern_sprachen" author="KL_MM"/>
</relation>
<relation kind="has_part" newpage="true">
<resource name="KL_pattern_learning" author="KL_MM"/>
</relation>
<damitspec isLecture="true" isCourse="false"/>
</topic>
```

Abbildung 4: Ein **topic**-Element im DaMiT-Inhaltsformat

3.2 Das Eingabeformat für DaMiT

Die Eingabe von Lernelementen wie im in Abbildung 4 gezeigten Format ist natürlich möglich, aber

sehr unkomfortabel. Oft können Einträge automatisch erfolgen, und die Inhaltsstruktur der **topic**- bzw. **unit**-Elemente kann aus der Eingabe extrahiert und dementsprechend in die Metadaten eingetragen werden. Das größte Problem wäre jedoch, Inhalte direkt so einzugeben, wie sie später auf dem Bildschirm erscheinen sollen. Dies gilt insbesondere für die Darstellung von Formeln.

Um diese Probleme zu lösen und die Eingabe zu erleichtern, wurde ein eigenes Eingabeformat definiert, das anschließend in das endgültige Format umgewandelt wird. Mit Hilfe dieses Formats können Lernelemente auch weiterhin wie in Abbildung 4 eingegeben werden, aber zahlreiche Vereinfachungen sind möglich, u. a.:

- Default-Werte müssen nicht angegeben werden.
- Die Definition verschiedener Varianten eines Lernelements (z. B. bzgl. Stil und Schwierigkeitsgrad) kann erheblich vereinfacht werden, da sich die meisten Metadaten nicht unterscheiden.
- Lernelemente können vollständig innerhalb anderer Lernelemente definiert werden. Die Struktur eines **topic**- bzw. **unit**-Elements wird dann automatisch aus der Eingabe extrahiert.

Natürlich stellt sich immer noch die Frage, auf welche Weise die später auf dem Bildschirm erscheinenden Inhalte eines Lernelements eingegeben werden sollen. Da in DaMiT viele mathematische Inhalte integriert sind, stellt sich insbesondere die Frage nach der Darstellung von Formeln im System. Sicherlich wird hier MathML⁶ in Zukunft von immer größerer Bedeutung sein, doch dieser Standard noch nicht von ausreichend vielen Browsern unterstützt. Aus diesem Grund fiel die Entscheidung auf eine Darstellung von Formeln mit Hilfe von Bildern.

Um diese Bilder zu erhalten, bot sich natürlich die Verwendung von \LaTeX an. \LaTeX ist ein weit verbreitetes Satzsystem, mit dem sich sehr gute Ergebnisse ohne genauere typographische oder gestalterische Kenntnisse erreichen lassen (ein Standardwerk für eine Einführung in \LaTeX ist [Kop02]).

Für \LaTeX sprechen zudem die folgenden Punkte:

- \LaTeX ist ein Standardformat für wissenschaftliche Arbeiten. Daher sind viele kompetente Autoren mit diesem Format vertraut, und es

⁶<http://www.w3.org/Math/>

```

%<unit multi="embedded,illustrated">
%<general name="dt_definition" description="Definition eines Entscheidungsbaums"
keyword="gl_classification,gl_tree,gl_decision_tree"/>
%<classification type="definition"/>
%<content variant="embedded">
Gegeben seien die Attribute  $a_0, \dots, a_n, \dots$  sowie die Klassen  $c_0, \dots, c_m, \dots$ 
in  $N$ .
Für alle Attribute  $a_i, 0 \leq i \leq n$ , bezeichne der Wertebereich  $\text{val}(a_i)$  die
möglichen Ausprägungen von  $a_i$ .
Ein Entscheidungsbaum ist ein Baum, in dem jedem inneren Knoten  $K$  eine
Entscheidungsfunktion  $f: \text{val}(a_0) \times \text{val}(a_1) \times \dots \times \text{val}(a_n) \rightarrow \{1, \dots, \text{grad}(K)\}$  sowie jedem Blatt  $L$  eine Klasse
 $c \in \{c_0, \dots, c_m\}$  zugeordnet ist.
%</content>
%<content variant="illustrated">
Ein Entscheidungsbaum ist ein Baum, in dem jedem inneren Knoten eine
Testfunktion sowie jedem Blatt eine Klasse zugeordnet ist. Das Ergebnis einer
Testfunktion wird bestimmt anhand der Attributwerte eines Datensatzes. Die Anzahl
der möglichen Testergebnisse in einem Knoten  $\mu$  endlich sein und entspricht
jeweils der Anzahl der S ohne dieses Knotens.
%</content>
%</unit>

```

Abbildung 5: Mit \LaTeX im DaMiT-Eingabeformat eingegebene Lernelemente

liegen bereits viele Inhalte vor, die für DaMiT genutzt werden können.

- Formeln, die in \LaTeX angegeben werden, können mit Hilfe von `latex2html`⁷ recht problemlos in entsprechende Grafiken (vom Typ PNG bzw. GIF) umgewandelt werden, und es besteht die berechtigte Hoffnung, daß sie bald auch nach MathML konvertiert werden können (dies ist derzeit nur auf Umwegen möglich).
- \LaTeX ist plattformunabhängig und unter der GNU GPL⁸ frei erhältlich.
- \LaTeX erlaubt die Eingabe von Unicode (UTF-8). Dies ist gerade bzgl. der Internationalisierung von Systemen ein sehr wichtiger Faktor.
- Es kann ein beliebiger Editor gewählt werden, der ASCII- bzw. UTF-8-Ausgaben produziert. Somit können die Autoren in vielen Fällen weiter mit ihrem gewohnten Editor (z. B. emacs) arbeiten.

Abbildung 5 zeigt mit \LaTeX im DaMiT-Eingabeformat eingegebene Lernelemente. Es handelt sich um die Definition eines Entscheidungsbaums, wobei diese Definition in den Varianten *embedded* und *illustrated* angegeben ist. Metadaten sind dabei einfach in Kommentarzeilen angegeben.

⁷<http://www.latex2html.org/>

⁸General Public License

3.3 Verarbeitung der Eingabe

Die Verarbeitung der Eingabe besteht aus zwei Schritten:

1. Validierung der Eingabe
2. Übersetzen des XML-Eingabeformats in das DaMiT-Inhaltsformat (natürlich nur, falls die Validierung ein positives Ergebnis geliefert hat)

Um diese Schritte durchzuführen, wird ein selbst entwickeltes, in Perl geschriebenes Tool benutzt: `latex2damit`. Für die Benutzung von Perl sprechen mehrere Gründe. Zum einen sind dies die Plattformunabhängigkeit sowie die freie Verfügbarkeit. Zum anderen – und das ist sicherlich von größerer Bedeutung – eignet sich Perl hervorragend zur Verarbeitung von Strings, und es gibt zahlreiche Module, die die Verarbeitung von XML unterstützen.

3.3.1 Validierung der Eingabe

Die Validierung wird weitestgehend mit Hilfe von XML Schema⁹ durchgeführt. XML Schema ist eine vom W3C¹⁰ empfohlene Schemasprache zur Validierung von XML-Dokumenten.

⁹<http://www.w3.org/TR/xmlschema-0/>

¹⁰<http://www.w3c.org/>

XML Schema ist leider – genauso wie fast alle anderen Schemasprachen (beispielsweise XML Schematron¹¹) – nicht mächtig genug. Daher ist ein vollständiger Test hiermit zwar nicht möglich, aber die Arbeit des zur Übersetzung eingesetzten Parsers wird erheblich vereinfacht, und es sind dort nur noch wenige Tests nötig.

3.3.2 Übersetzung in das DaMiT-Inhaltsformat

Um nach erfolgreicher Validierung letzte Tests vorzunehmen und die Eingabe in das endgültige DaMiT-Inhaltsformat zu übersetzen, wird ein eventorientierter Parser eingesetzt. Als Schnittstelle wird hierbei auf SAX¹² (*Simple API for XML*) zurückgegriffen. SAX ist eine standardisierte Schnittstelle zur Unterstützung eventorientierten Parsens von XML. Ursprünglich stand diese API nur für Java zur Verfügung, doch mittlerweile wird dieser Standard in vielen Programmiersprachen unterstützt. In Perl steht beispielsweise das in DaMiT verwendete Modul `XML::Parser::PerlSAX` zur Verfügung. Dies ist ein frei verfügbares Modul (unter der GNU GPL), das auf CPAN¹³ (dem Comprehensive Perl Archive Network) erhältlich ist.

Inhaltsersteller können Inhalte sowohl lokal als auch serverseitig per Web-Interface mit `latex2damit` bearbeiten. Zur Benutzung des Web-Interface muß man sich zunächst als Inhaltsersteller im DaMiT-System authentifizieren und dann eine zip-Datei mit allen benötigten Dateien (also tex-Files, Bilder, Multimedia-Dateien etc.) hochladen. War die Verarbeitung erfolgreich, kann eine zip-Datei mit allen zum Import benötigten Dateien (also Lernelementen, Multimedia-Dateien etc.) im vom System benötigten Format heruntergeladen werden.

In dem in Abbildung 5 gezeigten Beispiel liefert die Übersetzung mit `latex2damit` vier Lernelemente, unter anderem das in Abbildung 6 angegebene **unit**-Element. In diesem **unit**-Element wurden zahlreiche Metadaten ergänzt, und auf das in der Quelle innerhalb des Elements definierte **content**-Element wird mittels **relation** verwiesen. Das entsprechende Element ist in Abbildung 7 zu sehen. In diesem **content**-Element findet sich lediglich ein Verweis auf einen Inhalt in HTML-Format.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<unit>
<general name="dt_definition" language="de"
description="Definition eines Entscheidungsbaums"
keyword="gl_classification,gl_tree,
gl_decision_tree"/>
<lifecycle status="final"/>
<educational presentation="embedded"/>
<rights cost="0"/>
<relation kind="has_part" newpage="false">
<resource name="dt_definition-embedded-0"/>
</relation>
<classification type="definition"/>
</unit>
```

Abbildung 6: Ein aus der in Abbildung 5 gezeigten Eingabe generiertes **unit**-Element im DaMiT-Inhaltsformat

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<content>
<general name="dt_definition-embedded-0"/>
<technical format="HTML"
location="xml/dt_definition-embedded-0/
dt_definition-embedded-0.html"/>
</content>
```

Abbildung 7: Beispiel für die Definition eines **content**-Elements im DaMiT-Inhaltsformat

Dieser Inhalt ist in Abbildung 8 abgebildet und entspricht der *embedded*-Variante der in Abbildung 5 gezeigten Definitionen eines Entscheidungsbaums. Die Formeln sind dabei durch entsprechende Grafiken des Typs PNG ersetzt worden.

3.4 Import in das Lernsystem

Liegen die Lernelemente im DaMiT-Inhaltsformat vor, müssen sie natürlich noch in das Lernsystem eingespielt und integriert werden können. Um diese Funktionalitäten nutzen zu können, muß man sich zunächst wieder in der Rolle als *content provider* im System authentifizieren.

Beim Einspielen der Inhalte wird dann wie folgt vorgegangen:

Ein Thema anlegen/auswählen: Ein Thema dient dazu, eine Gruppe von Lernelementen (z. B. eine Lektion) zusammenzufassen. Wurde eine Gruppe ausgewählt, besteht die Möglichkeit, dort Inhalte einzuspielen bzw. zugehörige Inhalte zu integrieren.

¹¹<http://www.ascc.net/xml/resource/schematron/schematron.html>

¹²<http://www.saxproject.org/>

¹³<http://www.cpan.org/>

```

Gegeben seien die Attribute <IMG WIDTH="149"
HEIGHT="17" ALIGN="MIDDLE" BORDER="0"
SRC="img1.png" ALT="$ a_0, \ldots, a_n, \sim n \in \mathbb{N}$">
sowie die Klassen <IMG WIDTH="154" HEIGHT="17"
ALIGN="MIDDLE" BORDER="0" SRC="img2.png"
ALT="$ c_0, \ldots, c_m, \sim m \in \mathbb{N}$">.
<BR>
F&#252;r alle Attribute <IMG WIDTH="112"
HEIGHT="17" ALIGN="MIDDLE" BORDER="0"
SRC="img3.png" ALT="$ a_i, \ 0 \leq i \leq n$">,
bezeichne der Wertebereich <IMG WIDTH="62"
HEIGHT="20" ALIGN="MIDDLE" BORDER="0"
SRC="img4.png" ALT="$ \text{val}(a_i)$"> die
m&#246;glichen Auspr&#228;gungen von
<IMG WIDTH="20" HEIGHT="13" ALIGN="MIDDLE"
BORDER="0" SRC="img5.png" ALT="$ a_i$">.
<BR>
Ein Entscheidungsbaum ist ein Baum, in dem jedem
inneren Knoten <IMG WIDTH="22" HEIGHT="13"
ALIGN="BOTTOM" BORDER="0" SRC="img6.png" ALT="$ K$">
eine Entscheidungsfunktion <BR><BR>
<DIV ALIGN="CENTER">
<IMG WIDTH="470" HEIGHT="20" ALIGN="MIDDLE"
BORDER="0" SRC="img7.png" ALT="$ \displaystyle
f : \text{val}(a_0) \times \text{val}(a_1) \times \dots \times \text{val}(a_n) \rightarrow \{1, \dots, \text{grad}(K)\}$">
</DIV><BR><BR>
sowie jedem Blatt <IMG WIDTH="17" HEIGHT="13"
ALIGN="BOTTOM" BORDER="0" SRC="img8.png" ALT="$ L$">
eine Klasse <IMG WIDTH="133" HEIGHT="20"
ALIGN="MIDDLE" BORDER="0" SRC="img9.png"
ALT="$ c \ \text{in} \ \{c_0, \dots, c_m\}$"> zugeordnet ist.
<BR>

```

Abbildung 8: Das HTML-Element, auf das in Abbildung 7 verwiesen wird

Lernelemente einspielen: Die lokal oder auf dem Server erzeugten Lernelemente und zugehörigen Dateien (wie z. B. Bilder oder Flash-Filme) müssen als zip-File in das System hochgeladen werden. Dort erfolgt zunächst eine nochmalige Überprüfung (Vollständigkeit der Dateien, Eindeutigkeit der IDs etc.) und ggf. das Eintragen einiger letzter Metadaten (z. B. Autoren-IDs). Waren die Daten korrekt, besteht die Möglichkeit, die Inhalte in das System zu integrieren.

Bevor die Inhalte integriert sind, stehen sie nur dem Inhaltsersteller wie eine normale Lektion im System zur Verfügung und können auf Fehler untersucht werden. Für alle anderen Benutzer sind die Inhalte bis zu diesem Zeitpunkt nicht sichtbar.

Inhalte integrieren: Um eingespielte Inhalte auch für andere sichtbar zu machen, müssen sie vom Inhaltsersteller freigegeben werden. Danach können sie, wenn man sich in der Rolle als *teacher* im System authentifiziert hat, in das Inhaltsverzeichnis integriert werden.

In Abbildung 9 ist die uns bereits bekannte Definition nach erfolgreichem Einspielen in das DaMiT-System zu sehen. Das Layout (also Überschrift, Rahmen und Symbole) wird dabei durch den **unit-Typ** bestimmt.

Definition
Gegeben seien die Attribute $a_0, \dots, a_n, n \in \mathbb{N}$ sowie die Klassen $c_0, \dots, c_m, m \in \mathbb{N}$. Für alle Attribute $a_i, 0 \leq i \leq n$, bezeichne der Wertebereich $\text{Val}(a_i)$ die möglichen Ausprägungen von a_i . Ein Entscheidungsbaum ist ein Baum, in dem jedem inneren Knoten K eine Entscheidungsfunktion

$$f : \text{Val}(a_0) \times \text{Val}(a_1) \times \dots \times \text{Val}(a_n) \rightarrow \{1, \dots, \text{grad}(K)\}$$

sowie jedem Blatt L eine Klasse $c \in \{c_0, \dots, c_m\}$ zugeordnet ist.

Abbildung 9: Das in Abbildung 6,7 und 8 definierte Lernelement im DaMiT-System

4 Ausblick

Die am konkreten Beispiel DaMiT vorgestellte Architektur bietet viele Möglichkeiten zur Erweiterung, insbesondere im Bereich des Autorentools. Die Eingabe ist zwar wesentlich vereinfacht, aber sicher noch nicht so intuitiv, daß anspruchsvolle Inhalte von neuen Autoren ohne einen gewissen Einarbeitungsaufwand erstellt werden können. Wünschenswert wäre insbesondere eine durchgehende Werkzeugunterstützung bereits bei der Erstellung von Storyboards. Zudem sind weitere Vereinfachungen sowie das Anbinden einer GUI denkbar.

Des weiteren ist die Beschränkung auf \LaTeX als Eingabeformat nicht zwangsläufig nötig. Auch andere Formate könnten zur Inhaltserstellung herangezogen werden, allerdings muß jeweils untersucht werden, ob und wie die Verarbeitung von Formeln funktionieren kann.

Aktuell wird an einer Preview-Möglichkeit für die Autoren gearbeitet. Mit Hilfe dieser Funktionalität können Inhalte nicht nur lokal bearbeitet, sondern auch – zumindest in eingeschränkter Form – angezeigt werden. Somit müssen die Inhalte in Zukunft nicht mehr zuerst in das System eingespielt werden, um überprüft zu werden.

Zusammenfassung

Die in Abbildung 10 nochmals veranschaulichte Systematisierung des Workflows für die Inhaltserstellung in DaMiT zeigt, daß die hier verwendete

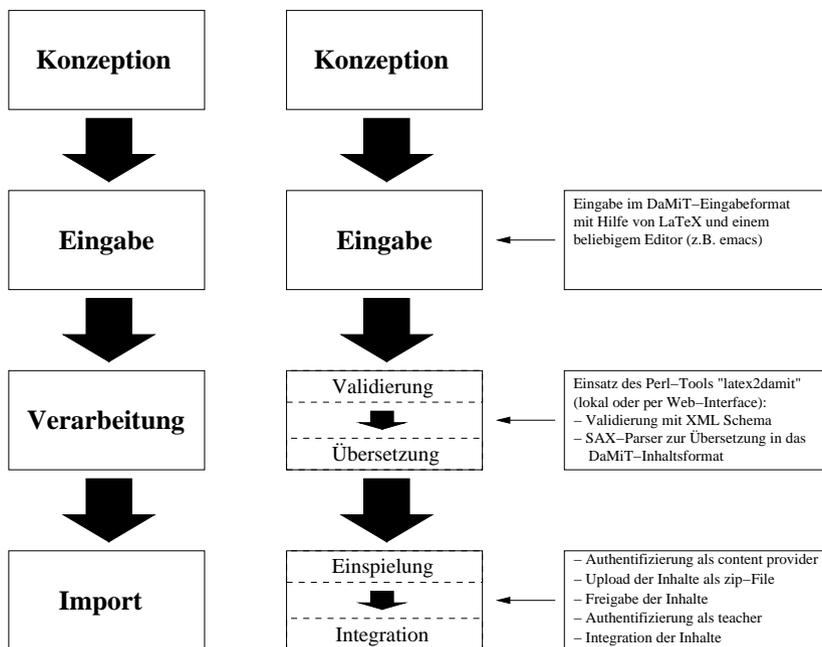


Abbildung 10: Der Workflow für die Inhaltserstellung im DaMiT-System

Architektur trotz der hohen Komplexität des Systems eine problemlose, transparente und an Standards orientierte Verknüpfung zwischen den Bausteinen Eingabe, Verarbeitung und Import ermöglicht. Besonders die Verwendung von XML, \LaTeX und Perl hat sich in diesem Kontext als eine gute Wahl erwiesen.

Alle Schnittstellen sind klar definiert, und einzelne Elemente des Workflows können ohne Beeinträchtigung des Gesamtprozesses modifiziert werden. Dies ist gerade in einem Verbundprojekt von großer Bedeutung, da hier verteiltes Arbeiten nicht nur gelegentlich stattfindet, sondern die übliche Vorgehensweise ist.

Durch die Definition eines eigenen Eingabeformats wird die Arbeit der Autoren erheblich vereinfacht, und es wird nicht zu viel technisches Hintergrundwissen vorausgesetzt.

Ein weiterer Vorteil für Autoren ist die Möglichkeit, in vielen Fällen mit ihrem bevorzugten Editor arbeiten zu können. Auch kann die Verarbeitung des Inhalts sowohl vollständig offline (bis auf die noch fehlende Preview-Funktionalität) oder komplett webbasiert erfolgen.

Literatur

- [Deg03] G. Degel. Nachhaltigkeit von e-Learning Projekten. Erfahrungen und Erwartungen am Beispiel DaMiT; im vorliegenden Band.
- [GLM03] G. Grieser, S. Lange, and M. Memmel. DaMiT: Ein adaptives Tutorsystem für Data-Mining; im vorliegenden Band.
- [Kop02] H. Kopka. *\LaTeX – Band 1: Einführung*. Addison-Wesley, 3rd edition, 2002.
- [RM02] E. T. Ray and J. McIntosh. *Perl & XML*. O'Reilly & Associates, 2002.
- [SBHG02] B. Schulz-Brünken, K. Herrmann, and R. Grimm. Kundenrollen als Vermarktungskonzept im E-Learning. In J. Herrmann K. P. Jantke and W. S. Wittig, editors, *Von e-Learning bis e-Payment. Das Internet als sicherer Marktplatz*, pages 20–26. Akademische Verlagsgesellschaft Aka, 2002.
- [SD02] J. Strutz and G. Degel. Offene Übungsaufgaben und Praktika im e-Learning. In J. Herrmann K. P. Jantke and W. S.

- Wittig, editors, *Von e-Learning bis e-Payment. Das Internet als sicherer Marktplatz*, pages 410–420. Akademische Verlagsgesellschaft Aka, 2002.
- [Spe99] M. Specht. *Adaptive Methoden in computerbasierten Lehr/Lernsystemen*. PhD thesis, Universität Trier, 1999.
- [TD01] B. Thalheim and A. Düsterhöft. Conceptual modelling of internet sites. In H.S. Kunii, S. Jajodia, and A. Solvberg, editors, *Proc. ER'2001*, number 2224 in Lecture Notes in Computer Science. Springer-Verlag, 2001.
- [TTR02] B. Tschiedel, B. Thalheim, and O. Ros-tanin. Kundenrollen als Vermarktungs-konzept im E-Learning. In J. Herrmann K. P. Jantke and W. S. Wittig, editors, *Von e-Learning bis e-Payment. Das Internet als sicherer Marktplatz*, pages 330–338. Akademische Verlags-gesellschaft Aka, 2002.
- [WCO00] L. Wall, T. Christiansen, and J. Orwant. *Programming Perl*. O'Reilly & Associates, 3rd edition, 2000.