<ML>³ Authoring mit Adobe FrameMaker

J. Brehm, N. Ossipova

Institut für Systems Engineering - System- und Rechnerarchitektur
Universität Hannover
Appelstraße 4, 30167 Hannover
{brehm,ossipova}@sra.uni-hannover.de

Kurzfassung. Der vorliegende Beitrag gibt einen Einblick in die autorbezogenen Aspekte der Entwicklung von Lehr- und Lernmodulen im Rahmen des Verbundprojektes Wissenswerkstatt Rechensysteme (WWR). Das Desktop-Publishing-Programm Adobe FrameMaker wird als ein WYSIWYG-unterstützendes Autorenwerkzeug für die Erstellung von Modulen vorgestellt, die in der XML-basierten Sprache <ML>³ implementiert werden. Aufgrund der hohen Anforderungen der <ML>³ an die XML-fähigen Autorensysteme beschränkte sich deren Auswahl auf FrameMaker. Um die Bearbeitung von <ML>³ Modulen in FrameMaker zu ermöglichen, wurde die <ML>³ strukturierte Anwendung entwickelt, mittels deren der Import und Export von <ML>³ Modulen gesteuert und das Layout für die visuelle Darstellung der zu bearbeitenden Dokumente festgelegt wird. Um das Java-Tool xmlConverter ergänzt stellt die <ML>³ strukturierte Anwendung dem Autor den gesamten <ML>³ Sprachumfang zu Verfügung.

1 Einleitung

Hauptziel des Verbundprojektes Wissenswerkstatt Rechensysteme (WWR) [1] ist die Erstellung von multimedialen, skalierbaren und rekombinierbaren Lehr- und Lernmodulen. Die Implementierung der Module erfolgt in der im Rahmen des Projektes entwickelten XML-basierten Beschreibungssprache <ML>³ [2]. <ML>³ stellt einen großen Umfang von Auszeichnungsmöglichkeiten bereit, um die geforderte Skalierbarkeit und Rekombinierbarkeit der Module zu erreichen.

Es ist schwerlich denkbar, dass die Autoren von <ML>³ Modulen die oft sehr umfangreichen Modulinhalte auf der Quellcode-Ebene erstellen würden. Angesichts des zum Anfang der Projektarbeit ungedeckten Bedarfs an komfortablen Autorenwerkzeugen wurde die Modulerstellung grundsätzlich in zwei folgende Schritte geteilt:

- Entwurf eines Drehbuchs, Sammlung von Inhalten;
- Umsetzung von Inhalten in XML, Erstellung und Einbindung von multimedialen Objekten.

Der letztere Schritt wird im universitären Alltag i.d.R. vom Autor an einen Produzenten delegiert.

Die wichtigsten Merkmale der <ML>³ Modulauszeichnungen sind folgende:

- Alle Modulinhalte werden nach Einsatzzweck *inhaltlich charakterisiert* (z.B. Definition, Anmerkung, Beispiel, Aufgabe, Lösung u.a.).

- Um die Skalierbarkeit der Module zu realisieren, erhalten die einzelnen Elemente jedes Moduls Auszeichnungen in drei Dimensionen: *Intensität* (basic, advanced, expert), *Zielgruppe* (student, teacher), *Ausgabemedium* (slide, online, script).
- Für eine flexible Anpassung von Modulen an verschiedene Lehr- und Lernmodelle wird jedes Modul von einer *didaktischen Beschreibung* begleitet, die eine Aufteilung von Modulinhalten in Lektionen und Lernschritte darstellt.

Eine hohe Qualität von Modulen wird ausschließlich dann sichergestellt, wenn die notwendigen inhaltlichen und didaktischen Ausprägungen der Module vom Autor selbst festgelegt werden. Um dies zu ermöglichen, wird ein komfortables Werkzeug benötigt, das den Autor bei dieser komplexen Aufgabe unterstützt. Die wichtigste Anforderung an das potentielle Autorenwerkzeug ist dabei die WYSIWYG-Unterstützung für die Erstellung und Wartung von <ML>³ Modulen. Weitere Anforderungen an das Autorenwerkzeug werden in Kapitel 2 beschrieben. Kapitel 3 stellt FrameMaker 7.0 als Authoring-Umgebung für die Bearbeitung von XML-Dokumenten vor. Kapitel 4 widmet sich der <ML>³ strukturierten Anwendung und dem xmlConverter.

2 Autorenwerkzeuge für <ML>³ Authoring

2.1 Anforderungen der <ML>³ an Autorenwerkzeuge

Die im Rahmen des Projektes entwickelte Beschreibungssprache <ML>³ implementiert eine Reihe von innovativen Konzepten, die eine flexible Gestaltung von Lehr- und Lernmodulen ermöglichen [3]. Die geforderte Flexibilität der Modulerstellung resultiert aus der Vielzahl der durch die <ML>³ bereitgestellten Auszeichnungsmöglichkeiten. Hierbei wird der gesamte Modulinhalt als eine strukturierte Sammlung von einzelnen Inhaltselementen betrachtet. Für jedes dieser Inhaltselemente wird explizit definiert, welchem *Einsatzzweck* es entspricht (z.B. Anmerkung, Definition, Zitat (vgl. Abbildung 1), Beispiel, Aufgabe u.a.) und wie es mit den anderen Inhaltselementen

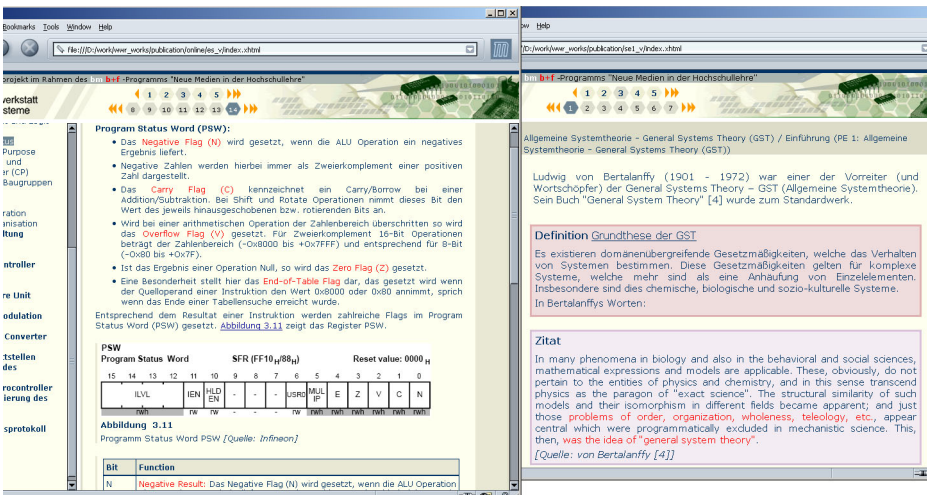

Abbildung 1. Darstellung der <ML>³ Inhaltselemente in der Online-Version

kombiniert wird. Durch dieses Modell der Modulimplementierung wird das Prinzip der Trennung von Inhalt und Layout bewahrt, da der Einsatzzweck lediglich eine Richtlinie für die Visualisierung der Inhalte ist und keine konkreten Layoutangaben vermittelt.

Entsprechend dem <ML>³ Metamodell [3] unterliegt der gesamte Modulinhalt dem sachlogischen und dem didaktischen Strukturierungsverfahren. Die sachlogische Strukturierung dient als Grundlage für eine Zusammensetzung eines Moduls aus diversen Inhaltskomponenten, wie Textabschnitten, multimedialen oder interaktiven Komponenten, die durch eine Vielzahl von <ML>³ Elementtypen repräsentiert werden. Demzufolge besteht der Erstellungsprozess von Modulinhalten in der Entwicklung von Text- und Medienobjekten mit deren gleichzeitiger Zuordnung zu einem dem jeweiligen Einsatzzweck entsprechenden Elementtyp.

Für einen produktiven Erstellungsprozess wird ein Autorenwerkzeug benötigt, das in der Lage ist, mindestens alle Textkomponenten und einfachen multimedialen Objekte (statische Bilder) anhand deren Elementtypen adäquat zu visualisieren und eine komfortable Bearbeitung Ersterer zu ermöglichen. Im Falle von komplexen multimedialen und interaktiven Objekten muss es möglich sein, diese bequem und anschaulich innerhalb der Module einzubinden. Diese allgemeinen Anforderungen setzen voraus, dass das potentielle Autorenwerkzeug vor allem über eine ausreichende Menge von Formatoptionen verfügen muss, die für jedes Inhaltsobjekt abhängig vom Elementtyp festgelegt werden können.

Zu den Elementen der <ML>³, deren Visualisierung in einer Autorenumgebung am häufigsten Probleme verursacht, gehören Tabellen, Fußnoten, Quelltexte, mathematische Ausdrücke und Grafiken.

- *Tabellen* werden von den meisten Textbearbeitungsprogrammen mit einer starren, unveränderbaren Struktur versehen.
- *Fußnoten* werden i.d.R. ausschließlich von den mächtigen Texteditoren unterstützt, die ihrerseits keine oder eine unzureichende XML-Unterstützung bieten.
- *Quelltexte* zeichnen sich als Problemstelle dadurch aus, dass die mittels Leerzeichen, Zeilenumbrüchen und Tabulatoren vorgenommene Formatierung beim Import und Export im XML-Format beibehalten werden muss.
- Zu den innovativen Konzepten der <ML>³ zählt die Berücksichtigung der Empfehlungen des W3-Konsortiums [4] bei der Auswahl von Datenformaten für multimediale Komponenten. Infolgedessen wird beispielsweise MathML für die Beschreibung von *mathematischen Ausdrücken* und Scalable Vector Graphics (SVG) für *Grafiken* verwendet. Jedoch werden die auf den W3C-Empfehlungen basierenden Formate derzeit nur von wenigen Standardanwendungen unterstützt.

In der XML-basierten Beschreibungssprache <ML>³ spielen *Attribute* eine wesentliche Rolle, da sie wichtige Informationen, wie z.B. Skalierungswerte der drei Dimensionen oder Referenzen auf multimediale Objekte, liefern. Demnach ist eine Möglichkeit zur Festlegung der Attributwerte durch das Autorenwerkzeug unverzichtbar.

Die didaktische Strukturierung eines Moduls, die Modulinhalte hinsichtlich verschiedener Lehr-/Lernziele und -situationen ohne Änderungen der sachlogischen Struktur kombinieren lässt, stellt eine weitere Anforderung an das potentielle Autorenwerk-

zeug. Eine didaktische Beschreibung besteht hauptsächlich aus *Verweisen auf die Inhaltskomponenten*, die auf der sachlogischen Seite definiert werden. Demzufolge benötigt das Autorenwerkzeug einen bequemen Mechanismus für die Erstellung und konsistente Bearbeitung von Querverweisen.

Da sowohl die sachlogische als auch die didaktische Auszeichnungsform eine verschachtelte Struktur widerspiegeln, wird eine Unterstützung der XML-typischen *Hierarchie der Elemente* vom potentiellen Autorenwerkzeug gefordert.

Der Evaluation der vorhandenen Autorenwerkzeuge bezüglich der hier angesprochenen Anforderungen widmet sich der folgende Abschnitt.

2.2 Evaluation der Autorenwerkzeuge

Parallel zur Modulerstellung wurde im Rahmen der Projektarbeit nach einem geeigneten Autorenwerkzeug gesucht, das die im vorherigen Abschnitt beschriebenen Anforderungen erfüllen würde. Hierbei wurden die meist verbreiteten Textbearbeitungsprogramme bevorzugt, da diese Programme einem breiten Nutzerkreis vertraut sind. Des Weiteren lagen viele Modulinhalte schon zum Projektbeginn in den von diesen Programmen verwendeten Formaten vor.

Zur Anfangszeit der Modulerstellung im Rahmen des Projektes bot noch keine der Standardanwendungen eine WYSIWYG-Unterstützung für die Bearbeitung von XML-Dokumenten. Zugleich verfügten mehrere Textbearbeitungsprogramme über Exportmöglichkeiten von XML, zu denen u.a. *FrameMaker 6.0* und *MS Word 2000* gehörten. Um FrameMaker 6.0 und MS Word 2000 [5] zur Erstellung von <ML>³ Modulen heranzuziehen, wurden für jedes dieser Programme jeweils eine spezielle Vorlage und ein Konvertierungstool entwickelt. Die aufgrund dieser Vorlagen erstellten und anschließend nach XML exportierten Dokumente konnten mithilfe der Konvertierungstools an das <ML>³ Format angepasst werden. Jedoch war eine manuelle Nachbearbeitung von XML-Quelltexten stets erforderlich. Darüber hinaus wiesen die Vorlagen mehrere Einschränkungen der <ML>³ gegenüber auf, z.B. keine Verschachtelung von Tabellen und Listen. FrameMaker 6.0 bot keine Exportmöglichkeit von mathematischen Ausdrücken im XML-Format. Ein Mechanismus zur Festlegung von Attributen war auf einer benutzerfreundlichen Weise nicht zu realisieren. Der Einsatz der beiden Programme zur Wartung von Modulen wurde jedoch dadurch ausgeschlossen, dass der Import von <ML>³ Dokumenten in FrameMaker 6.0 und MS Word 2000 nicht implementierbar war.

Ein weiterer Ansatz, der auf Basis einer Formatvorlage und eines Konvertierungstools funktioniert, ist *OpenOffice.org*. OpenOffice.org unterstützt MathML und SVG, Einbindung von multimedialen Komponenten sowie Referenzierung von Inhalten und bietet viele Erweiterungsmöglickeiten durch Makros, eigene Java-Klassen und ein Filter API. Die Implementierung des Imports und Exports von <ML>³ Modulen sowie einer kompletten Arbeitsumgebung ist jedoch mit einem hohem Aufwand verbunden [6].

Eines der ersten Programme, das eine Bearbeitung von XML-Dokumenten auf der visuellen Ebene bietet, war *Altova XMLSpy*. Die WYSIWYG-Unterstützung wird durch die zwei folgenden Komponenten von XMLSpy ermöglicht: Stylesheet Designer und Authentic. Authentic stellt XML-Dokumente aufgrund der mithilfe vom Stylesheet Designer entwickelten Templates visuell dar. Die Bearbeitung von Inhalten erfolgt direkt

in XML-Dateien, wodurch das Importieren und Exportieren entfallen. Leider fehlen dem Programmpaket z.Z. noch einige wichtige Funktionalitäten. Z.B. können unformatierte Zeichenfolgen vom XML-Datentyp CDATA im Authentic-View nicht dargestellt werden. Weiterhin besitzt XMLSpy keinen Verweismechanismus. Fußnoten, einige Tabellenmodelle und MathML-Formeln werden ebenfalls nicht korrekt dargestellt. Die Implementierung des Werkzeugs ist außerdem noch nicht ausgereift, wodurch dessen Nutzung etwas unkomfortabel ist und die <ML>³ nicht in ihrem vollen Umfang unterstützt wird. Die Weiterentwicklung dieser Programme lässt dennoch die Hoffnung auf einen funktionsreichen nativen WYSIWYG-Editor für <ML>³ Module.

Mit dem Erscheinen der Version 7.0 von FrameMaker steht dem Autor eine komplett strukturierte Authoring-Umgebung zur Verfügung. *FrameMaker 7.0* bietet eine Reihe von Werkzeugen, die eine komfortable Arbeit mit XML-Dokumenten auf der WYSIWYG-Ebene ermöglichen. Des Weiteren erlaubt FrameMaker 7.0 eine Erweiterung um sogenannte strukturierte Anwendungen, mittels deren sowohl der Import und Export von XML-Dokumenten gesteuert als auch das Layout für die visuelle Darstellung der zu bearbeitenden Dokumente festgelegt wird. Die zu diesem Zweck entwickelte <ML>³ strukturierte Anwendung stellt dem Autor den gesamten <ML>³ Sprachumfang, mit derzeitiger Ausnahme der Tabellenverschachtelung, zur Verfügung.

Zu den wesentlichen Vorteilen der <ML>³ Modulerstellung mit FrameMaker unter Verwendung der <ML>³ strukturierten Anwendung zählen neben der WYSIWYG-Unterstützung folgende:

- Visualisierung der Elementhierarchie sowie eine bequeme Bearbeitung von Attributwerten und Namensraumdefinitionen,
- Nutzung der vordefinierten Tabellen,
- Unterstützung aller gängigen Grafikformate sowie des SVG-Formats,
- ein umfangreicher Querverweismechanismus,
- Beibehaltung der vorgenommenen Formatierung von Quellcodes beim Import und Export von XML-Dokumenten sowie
- Import und Export von MathML-Formeln.

Die für das <ML>³ Authoring relevanten Werkzeuge in FrameMaker 7.0 sowie die <ML>³ strukturierte Anwendung werden in den folgenden Abschnitten behandelt. FrameMaker 7.0 wird fortan als FrameMaker bezeichnet.

3 Werkzeuge in FrameMaker

3.1 Strukturierte Authoring-Umgebung

Das Desktop-Publishing-Programm FrameMaker kombiniert eine mächtige Textverarbeitungsumgebung mit den wichtigsten Funktionalitäten eines XML-Editors. Einerseits kann der Autor ein Dokument auf die gewöhnliche Weise im WYSIWYG-Modus bearbeiten, andererseits besteht dabei die Möglichkeit, das Dokument zugleich zu strukturieren und alle notwendigen XML-bezogenen Eigenschaften des Dokumentes festzulegen. Letztere setzt jedoch eine Vertrautheit mit den wichtigsten XML-Bestandteilen voraus, wie z.B. Elemente, Attribute, Namensräume, Elementstruktur u.a. Für die Erstellung von hochqualitativen Lehr-/Lernmodulen ist das Verständnis des Einsatz-

zwecks jedes einzelnen Elementes und Attributes der <ML>³ von wesentlicher Bedeutung, da nur dadurch der volle Umfang der <ML>³ Auszeichnungsmöglichkeiten ausgenutzt werden kann.

Die für die Bearbeitung von XML-Dokumenten relevanten Komponenten der Authoring-Umgebung (s. Abbildung 2) werden in den folgenden Abschnitten vorgestellt.

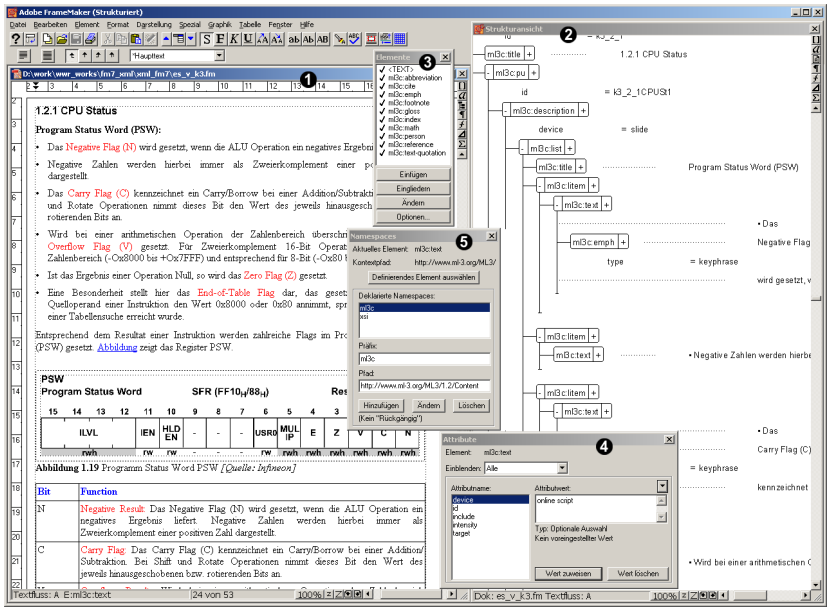

Abbildung 2. Authoring-Umgebung in FrameMaker

Dokumentfenster und Strukturansicht. FrameMaker stellt dem Autor zwei verschiedene Darstellungen eines Dokumentes zur Verfügung: die Dokumentansicht (1) und die Strukturansicht (2). Während das Dokumentenfenster den formatierten Inhalt eines Dokumentes darstellt, visualisiert die Strukturansicht die Elementhierarchie des Dokumentes. Das Dokumentenfenster und die Strukturansicht agieren synchron und spiegeln jeweils die im anderen Fenster vorgenommenen Aktionen wider.

Entspricht ein Element oder ein Attribut im Dokument nicht seiner Definition, so liegt ein Fehler vor. Dieser wird in der Strukturansicht durch die Farbe Rot und ein dem Typ des Fehlers entsprechendes Symbol gekennzeichnet. Zu den möglichen Fehlern zählen fehlende obligatorische Elemente, Elemente an einer ungültigen Position, undefinierte Elemente, ungültige Attribute und fehlende obligatorische Attributwerte.

Elementkatalog. Das Einfügen von Elementen in ein strukturiertes Dokument erfolgt über den Elementkatalog (3). Der Elementkatalog enthält eine Auflistung der an der aktuellen Position gültigen Elemente und stellt Befehle zum Hinzufügen und Bearbeiten von Elementen bereit.

Attribute. Attribute für jedes Element werden in der Strukturansicht angezeigt. Weist ein Element Attribute auf, erscheint auf der rechten Seite des Elementrechtecks ein Pluszeichen, wenn einige oder alle Attribute ausgeblendet sind, oder ein Minuszeichen, wenn alle Attribute eingeblendet sind. Attribute werden im Dialogfenster "Attribute" (4) bearbeitet.

Namensräume. FrameMaker unterstützt die Verwendung von Namensräumen in strukturierten Dokumenten. Beim Import und Export von XML-Dateien werden alle Namensraumdefinitionen beibehalten. Bearbeitung von Namensräumen erfolgt im Dialog "Namespaces" (5).

3.2 Strukturierungswerkzeuge

Werkzeuge, die es ermöglichen, einem FrameMaker-Dokument eine Struktur hinzuzufügen, werden zu einer strukturierten Anwendung (*structured application*) kombiniert. Die im Rahmen des WWR-Projektes entwickelte <ML>³ strukturierte Anwendung erlaubt die Bearbeitung von <ML>³ Modulen als strukturierte FrameMaker-Dokumente. Eine strukturierte Anwendung ist eine Zusammenstellung von Dateien, die beschreiben, wie ein strukturiertes Dokument von FrameMaker verarbeitet werden soll. Sie enthält eine Strukturbeschreibung in Form einer Document Type Definition (DTD), Formatregeln, Lese- und Schreibregeln für den Import bzw. Export von XML-Dateien sowie eine Importvorlage.

Elementdefinitionen und Formatregeln werden in einem strukturierten FrameMaker-Dokument namens *Element Definition Document* (EDD) festgelegt, das eine interne Beschreibungssprache implementiert. Diese beschreibt Elemente, deren Unterelemente und Attribute und lässt sowohl Import als auch Export einer DTD zu. *Formatregeln* sind ein Teil dieser Beschreibungssprache. Für jedes im EDD definierte Element wird mittels der Formatregeln festgelegt, wie es in der Dokumentansicht dargestellt wird. Abbildung 3 zeigt ein Beispiel einer Elementdefinition im EDD.

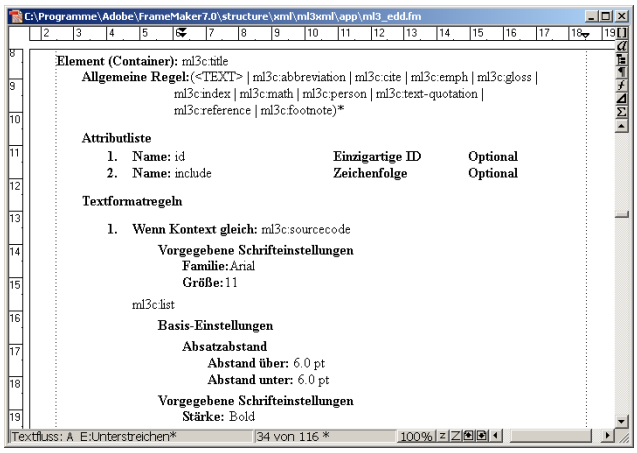

Abbildung 3. Elementdefinition im EDD

In Ergänzung zum EDD existiert die Möglichkeit, Import und Export von XML-Dokumenten explizit mittels *Lese- und Schreibregeln* zu steuern. Hierbei können beispielsweise Elemente oder Attribute umbenannt oder gelöscht, Attributwerte zugewiesen, Tabellenteile identifiziert oder Grafiken verknüpft werden.

Elementdefinitionen und Formatregeln des EDD werden in eine *Importvorlage* übertragen, die anschließend dem Autor ausgeliefert wird. Die aufgrund dieser Vorlage erstellten Dokumente können im XML-Format exportiert werden.

Die Funktionalitäten der in FrameMaker verfügbaren Strukturierungswerkzeuge reichen i.d.R. aus, um alle DTD-basierten XML-Dokumente zu unterstützen. Jedoch basiert die <ML>³ auf einer anderen der W3C-Empfehlungen, dem XML-Schema. Daraufhin wurde die <ML>³ strukturierte Anwendung um das Java-Tool xmlConverter ergänzt, um den Import und Export von <ML>³ Dateien zu ermöglichen. Die <ML>³ strukturierte Anwendung und der xmlConverter werden im nächsten Kapitel behandelt.

4 <ML>³ strukturierte Anwendung und xmlConverter

4.1 <ML>³ XML-Roundtripping

Die im Rahmen des WWR-Projektes entwickelte <ML>³ strukturierte Anwendung implementiert die <ML>³ Elementdefinitionen. Durch die Tatsache, dass die im EDD definierte Struktur in Form einer DTD formuliert wird, während die <ML>³ als XML-Schema vorliegt, ist eine direkte Übernahme von <ML>³ Elementdefinitionen in das EDD nicht realisierbar. Demzufolge weist die EDD-Elementdefinition einige Unterschiede zur <ML>³ auf. Z.B. unterstützt EDD keine gleichnamigen Elemente mit verschiedenen Inhalten und keine Muster für Attributwerte.

Der gesamte Vorgang des Imports eines XML-Dokumentes in FrameMaker mit dessen anschließendem Export in das XML-Format, auch *XML-Roundtripping* genannt, besteht im Falle der <ML>³ aus folgenden Schritten (vgl. Abbildung 4):

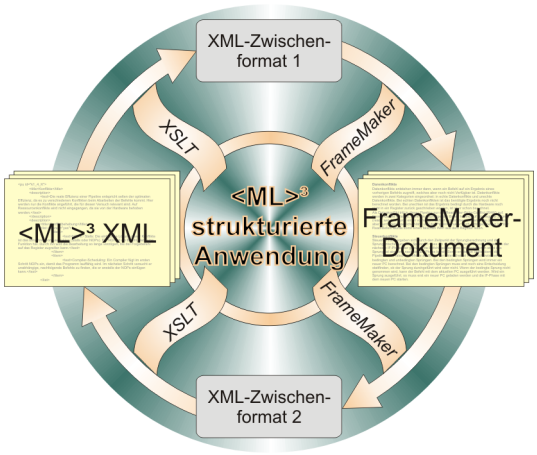

Abbildung 4. <ML>³ XML-Roundtripping

- Vor dem Import in FrameMaker wird die zu importierende <ML>³ Datei mittels einer Stylesheet-Transformation (XSLT) an das EDD angepasst, dabei wird die resultierende XML-Datei in einem EDD-konformen Zwischenformat gespeichert.
- Anschließend kann diese XML-Datei in FrameMaker geöffnet und unter Rücksicht auf die Unterschiede zwischen der <ML>³ und dem EDD bearbeitet werden.
- Nach dem Bearbeiten wird das Dokument in ein zweites XML-Zwischenformat exportiert. Eine weitere Stylesheet-Transformation bringt das Dokument in eine <ML>³ konforme Form. Das zweite Zwischenformat ergibt sich daraus, dass interne Querverweise und Doctype-Deklarationen derzeit von FrameMaker nicht dem ersten Zwischenformat identisch exportiert werden.

Die Stylesheet-Transformationen der zu importierenden und der exportierten XML-Dateien werden mittels des xmlConverters durchgeführt.

4.2 xmlConverter

Die plattformunabhängige Java-Anwendung xmlConverter wurde entwickelt, um eine einfache bidirektionale Konvertierung zwischen der <ML>³ und dem EDD-Format zu ermöglichen. Abbildung 5 zeigt das Programmfenster des xmlConverters. Im oberen

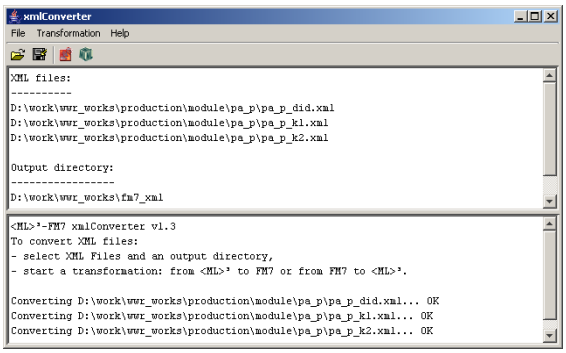

Abbildung 5. xmlConverter

Teil des Fensters werden die Namen der zur Konvertierung ausgewählten XML-Dateien und des Ausgabeverzeichnisses angezeigt, im unteren Hinweise zur Transformation sowie Status- und eventuelle Fehlermeldungen ausgegeben.

Im Rahmen jeder Konvertierung werden folgende Aktionen vom xmlConverter unter Verwendung der bereitgestellten Stylesheets durchgeführt:

- Um die Unterschiede zwischen der <ML>³ und der EDD-Elementdefinition auszugleichen, werden einzelne Elemente und Attribute mithilfe der Stylesheets erzeugt, gelöscht oder umbenannt.
- Da MathML-Entitäten bei jeder Stylesheet-Transformation automatisch aufgelöst werden, müssen sie vor der Transformation auskommentiert und im Anschluss wieder in die ursprüngliche Form gebracht werden.

- Um den Import von MathML-Entitäten durch FrameMaker zu ermöglichen, wird in den zu importierenden Dateien eine DTD referenziert, welche die EDD-Elementdefinition widerspiegelt und die MathML-Entitätsdeklarationen inkludiert.

Das Programm basiert auf den gleichen Java-Klassen, die auch dem XSLT-Prozessor Apache Xalan-Java 2 zugrunde liegen, wodurch ein adäquates Ergebnis der Stylesheet-Transformation auf jeder Plattform sichergestellt wird.

5 Zusammenfassung

Die Entwicklung von Lehr- und Lernmodulen im Rahmen des WWR-Projektes stellt aufgrund des großen Umfangs der <ML>³ Auszeichnungsmittel eine hohe Herausforderung für den Autor dar. Um dem Autor eine produktive Modulerstellung zu ermöglichen, wird ein komfortables Autorenwerkzeug benötigt. Die wichtigsten Kriterien während der Suche nach einem geeigneten Autorenwerkzeug waren zum einen eine umfassende WYSIWYG-Unterstützung für die Bearbeitung von XML-Dokumenten, zum anderen die für die Festlegung verschiedener Modulausprägungen notwendigen XML-bezogenen Funktionalitäten. In einer Evaluation von Autorenwerkzeugen wurde festgestellt, dass FrameMaker alle ausgearbeiteten Kriterien erfüllt. FrameMaker bietet eine funktionsreiche Authoring-Umgebung für die Arbeit mit XML-Dokumenten sowie eine einfache Erweiterungsmöglichkeit um eigene strukturierte Anwendungen, welche die Verarbeitung von XML-Dokumenten durch FrameMaker steuern.

Die <ML>³ strukturierte Anwendung, in der die <ML>³ Definition gefasst wird, ermöglicht das XML-Roundtripping für <ML>³ Dokumente, das deren Import, Bearbeitung und Export einschließt. Aufgrund der XML-Schema-basierten Implementierung der <ML>³ wurde eine Ergänzung der <ML>³ strukturierten Anwendung um ein Java-Tool zur bidirektionalen Übersetzung zwischen der <ML>³ und dem EDD-kompatiblen XML-Format erforderlich.

Unter Verwendung der <ML>³ strukturierten Anwendung stellt FrameMaker ein leistungsstarkes Autorenwerkzeug dar, das über den Projektrahmen hinaus zur Erstellung und Wartung von <ML>³ Modulen einsetzbar ist.

Literatur

[1] "Wissenswerkstatt Rechensysteme". http://www.wwr-project.de

[2] "Multidimensional LearningObjects and Modular Lectures Markup Language (<ML>³)". http://www.ml-3.org

[3] L. Kornelsen, U. Lucke, D. Tavangarian, D. Voigt, M. Waldhauer: "Inhalte und Ergebnisse des Verbundprojekts Wissenswerkstatt Rechensysteme (WWR)", GI Softwaretechnik-Trends, Band 24, Heft 1, ISSN 0720-8928, Februar 2004, S. 72-81.

[4] World Wide Web Consortium. http://www.w3.org

[5] T. Gecks, D. Henrich: "Von Microsoft Word nach <ML>³ - Ein Konvertierungswerkzeug". In diesem Band.

[6] L. Kornelsen, U. Lucke, D. Tavangarian, M. Waldhauer, N. Ossipova: "Strategien und Werkzeuge zur Erstellung multimedialer Lehr- und Lernmaterialien auf Basis von XML". Eingereicht zu DeLFI 2004. http://www.delfi2004.de.