

Strategien und Werkzeuge zur Erstellung multimedialer Lehr- und Lernmaterialien auf Basis von XML

Lars Kornelsen, Ulrike Lucke, Djamshid Tavangarian, Matthias Waldhauer, Natalia Ossipova
Universität Rostock, Institut für Informatik, Lehrstuhl für Rechnerarchitektur
{vorname.nachname}@uni-rostock.de

Abstract: Der Beitrag beschreibt ausgehend von den theoretischen Grundlagen eines XML-basierten Authoring-Prozesses die grundlegenden Strategien sowie die dabei einsetzbaren Werkzeuge für die Erstellung von XML-basiertem Multimedia-Material. Dabei wird zwischen der Erstellung in einem anderen Format und anschließender Konvertierung sowie der direkten Erstellung von XML-Dokumenten differenziert. Nach einer Vorstellung verschiedener Import- und Exportmechanismen sowie Autorenwerkzeuge werden diese hinsichtlich ihrer Eignung für den Erstellungsprozess verglichen und bzgl. ihrer Akzeptanz im praktischen Einsatz evaluiert. Dabei gilt ein Hauptaugenmerk der hinreichenden Unterstützung auch von technisch weniger versierten Autoren. Abschließend werden bislang ungelöste Probleme und nötige Entwicklungsarbeiten aufgezeigt.

1 Einleitung

Von ausschlaggebender Bedeutung für die erfolgreiche Gestaltung von eLearning-Szenarien ist eine ausreichende Menge hochqualitativer Inhalte, die entsprechend dem Einsatzzweck ausgewählt und flexibel an individuelle inhaltliche und/oder didaktisch-pädagogische Anforderungen angepasst werden können. Zur Erstellung von Lehr- bzw. Lernmaterial werden in zunehmendem Maße die Extensible Markup Language (XML) bzw. auf XML basierende Sprachen genutzt [Ri03]. Nicht nur textuelle Elemente können so strukturiert, semantisch ausgezeichnet und mit Metadaten annotiert werden, sondern auch dynamische bzw. interaktive Multimedia-Komponenten, Formeln, u. v. m.

- Textauszeichnung auf Basis von XML: DocBook [WM99], IMS Content Packaging [IMS], Learning Material Markup Language (LMML) [Fr02], Multidimensional LearningObjects and Modular Lectures Markup Language (<ML>³) [LTV03]
- Beschreibung von Medienobjekten auf XML-Basis: Scalable Vector Graphics (SVG) [W3C], Synchronized Multimedia Integration Language (SMIL) [W3C], VoiceXML [W3C]
- XML-Sprache zur Beschreibung mathematischer Formeln: MathML [W3C]

Oft wird auf eine Vielzahl von Sprachen zurückgegriffen. Meist werden darüber hinaus auch Komponenten in anderen Formaten (z. B. JPG, WAV, AVI, Java, Flash) integriert. Der Vorteil einer XML-basierten Auszeichnung von nicht-textuellen Objekten ist, dass textbasierte Suchmechanismen auch innerhalb dieser Objekte ansetzen können.

Aufgrund seines abstrakten Charakters kann XML sehr gut automatisch verarbeitet werden. Für die Erstellung von derartigen Inhalten ist dies jedoch nachteilig. Während die Struktur eines Dokuments graphisch in einem Baum problemlos visualisierbar ist, erfolgt die Attributierung einzelner Elemente auf einem oft sehr abstrakten Niveau und ist für den Autor schwer zu überblicken. Die Auswirkungen von Stylesheets auf das Dokument kann eine Autorenumgebung nur eingeschränkt darstellen. Der volle Funktionsumfang einer XML-Sprache ist daher i. A. nur in einer XML-ähnlichen Notation editierbar.

Dies ist für die meisten Autoren jedoch nicht zumutbar. Daher ist hier unbedingt eine WYSIWYG-Umgebung (What You See Is What You Get) zu favorisieren, die das Dokument ähnlich seiner endgültigen Gestalt darstellt und editierbar macht. Dabei sollte auf unterschiedliche Nutzertypen, Rollen, Vorgehensweisen und Arbeitsumgebungen Rücksicht genommen werden. Ziel sollte in jedem Fall eine integrierte Autorenumgebung sein, die alle Phasen des Erstellungsprozesses unterstützt.

Dieser Beitrag stellt im folgenden Kapitel zunächst einige theoretische Grundlagen der Erstellung von XML-basierten eLearning-Dokumenten vor. Kapitel 3 beschreibt hierzu unterschiedliche Strategien und die dafür jeweils geeigneten Werkzeuge. Abschließend gibt Kapitel 4 eine vergleichende Zusammenfassung der Aussagen sowie einen Ausblick auf dringend nötige Entwicklungen zu diesem Thema. Da der Artikel einen Überblickscharakter hat, beinhaltet er keinen expliziten Abschnitt *Related Work*, sondern der aktuelle Entwicklungsstand wurde in die betreffenden Abschnitte eingeflochten.

2 Grundlagen des Authoring-Prozesses

2.1 Phasen-Modell

Bei der XML-basierten Dokument-Erstellung werden mehrere Phasen durchlaufen (z. T. in anderer Reihenfolge bzw. unter Wiederholung / Auslassung einzelner Schritte):

1. Während der *Informationssammlung* werden relevante Themen bzw. Fakten zusammen-gestellt und in einem Drehbuch strukturiert.
2. Die *Formalisierung* bringt das Drehbuch in die ausgewählte XML-Notation. Die nachfolgenden Schritte nutzen dieses interne, maschinenlesbare Format.
3. Durch *Medienobjekte* wird das Dokument multimedial angereichert. Dies setzt geeignete Anknüpfungspunkte im Text voraus.
4. Für den Einsatz in Aus- oder Weiterbildung ist eine Beschreibung *didaktischer Aspekte* des Materials notwendig.
5. Ein dem Einsatzgebiet entsprechendes *Layout* ergänzt das Dokument um gestalterische Informationen.
6. Mit Hilfe von *Transformationen* wird das Dokument (inkl. der Medienobjekte und der didaktischen Informationen und unter Nutzung des entworfenen Layouts) in ein für den Endanwender geeignetes Format überführt.
7. Eine *Evaluation* nach fachlichen, didaktisch-pädagogischen und gestalterischen Gesichtspunkten stellt eine hohe Qualität des Materials sicher.
8. Wurden alle vorherigen Phasen erfolgreich durchlaufen, wird das Material *publiziert* und den Anwendern zur Verfügung gestellt .

Jede Phase ist mit spezifischen Arbeiten verbunden, die gesonderte Anforderungen an den ausführenden Nutzer bzw. das ihn unterstützende Werkzeug stellt.

2.2 Nutzertypen und Rollen

Das im vorgestellten Authoringprozess aktive Nutzerspektrum ist breit gefächert.

- Die Informationssammlung (1) und die fachliche Evaluation (7) werden von einem *Fachexperten* des betreffenden Themenkreises durchgeführt.
- XML-Codierung des Dokuments (2), Erstellung von Transformationswerkzeugen (6) und Publikation des fertigen Dokuments (8) erfolgen durch einen *XML-Experten*.
- Die Erstellung von Medienobjekten (3) realisiert ein *Multimedia-Experte*.

- Didaktische Aspekte (4) werden durch einen *Didaktik-Experten* resp. *Pädagogen* beschrieben, der auch die didaktische Evaluation (7) übernimmt.
- Den Entwurf von Layouts (5) und die gestalterische Evaluation (7) nimmt ein *Grafik-Experte* bzw. künstlerischer Gestalter vor.

Dabei kann eine Person verschiedene Rollen einnehmen. I. A. leitet der Fachexperte den Prozess; meist nimmt er auch die Rolle des Pädagogen und Evaluators wahr. Die restlichen Rollen werden oft unter dem Begriff / in der Person des Produzenten zusammengefasst.

Die zur Erstellung XML-basierter Dokumente auf nativer Ebene benötigten Kenntnisse können nur bei einem ausgewiesenen XML-Experten vorausgesetzt werden. Zumindest bei Fachexperten aus dem technischen Bereich geht man von einem noch vertretbaren Einarbeitungsaufwand aus. Die anderen Produzenten sind jedoch meist nicht mit der XML-Technologie vertraut und benötigen eine Unterstützung durch Werkzeuge auf einem höheren Abstraktionsniveau. Hiervon können wiederum auch XML-Experten profitieren.

2.3 Anforderungen an Werkzeuge

Zu den daraus resultierenden Anforderungen an Autorenwerkzeuge zählen neben hinreichendem Funktionsumfang, intuitiver Benutzerführung und weitestmöglicher Integration in vorhandene Infrastrukturen bzw. Arbeitsumgebungen vor allem auch einige wichtige Kriterien, die aus der Verwendung von XML resultieren:

- Struktur und Inhalte des Dokuments, seine Elemente und deren Attribute müssen vollständig und übersichtlich *visualisiert* werden, ohne den Betrachter zu irritieren.
- Alle strukturellen und inhaltlichen Komponenten müssen komfortabel *editierbar* sein. Direkte Modifikationen im nativen XML-Code sind zu vermeiden.
- Fehlerhafte Elemente oder Attribute (an falscher Stelle, vom falschen Typ oder fehlend) sind zu erkennen. Das Dokument sollte stets *valide* sein.
- Da XML-Dokumente häufig auf dem Konzept der *Referenzierung* beruhen, ist eine zuverlässige Querverweisfunktionalität hilfreich.
- Externe *Medienobjekte* müssen (zumindest in vereinfachter Form als VorschauBild) integrierbar sein.

Alle Werkzeuge, die für die Erstellung XML-basierter Dokumente in Betracht gezogen werden, sollten bereits im Vorfeld kritisch geprüft werden, bevor eine für den gesamten Erstellungsprozess bindende Entscheidung gefällt wird.

3 Verfügbare Werkzeuge

Generell können 2 Strategien der Erstellung von XML-basierten Inhalten unterschieden werden: die Erstellung in einem anderen Format unter anschließender Konvertierung und die Erstellung direkt in XML. Bei Konvertierungsmechanismen ist zwischen Import- und Exportmöglichkeiten von Standardwerkzeugen, Stylesheet-Transformationen sowie eigenen Implementierungen zu unterscheiden. Im Folgenden werden unabhängig von existierenden XML-Sprachen die verfügbaren Werkzeuge gegenübergestellt. Abschnitt 3.1 beschreibt den Import aus anderen Formaten. In Abschnitt 3.2 werden Standardwerkzeuge auf ihre Eignung untersucht. Abschnitt 3.3 behandelt die Möglichkeiten zum Export in andere Formate. Abschließend werden die Werkzeuge in Abschnitt 3.4 bzgl. ihrer praktischen Eignung und der Akzeptanz durch die Nutzer evaluiert.

3.1 Importfilter

Häufig soll bestehendes Material (z.B. Skripten oder Präsentation) wiederverwendet werden. Daher sind Werkzeuge zur Konvertierung von Fremdformaten nach XML nötig. In diesem Abschnitt sollen hierzu verschiedene Ansätze vorgestellt werden.

Skripte in MS Word. Eine Transformation von Word-Dokumenten ist oft problematisch, da ältere Versionen auf einem proprietären Binärformat arbeiten. Die seit MS Office 2003 genutzte XML-Syntax lässt sich mit einer Transformation auf Basis der Extensible Stylesheet Language (XSL) in beliebige XML-Sprachen umsetzen. Um das Binärformat als Ausgangspunkt für XML-Dokumente zu verwenden, sind andere Techniken erforderlich. Es existieren bereits Werkzeuge für eine Übersetzung in ausgewählte Sprachen, die jedoch häufig auf dem verwandten RTF-Format arbeiten [IL]. Eine an der Universität Bayreuth entwickelte Möglichkeit besteht in der Verwendung spezieller Absatz- und Zeichen-vorlagen, die mit den Konstrukten der Zielsprache korrespondieren [GH04]. Dort sind versteckte Strukturierungsanweisungen enthalten. Ein spezielles Word-Plugin [SCH] ermöglicht dann den Export in eine XML-Datei. Der Hauptnachteil dieses Umwandlungs-mechanismus' liegt darin, dass eine installierte Microsoft-Word-Applikation zwingend erforderlich ist. Zudem sind die Dokumente nachträglich mit den speziellen Absatz- und Zeichen-Vorlagen zu formatieren, was einen beträchtlichen Aufwand darstellen kann.

Präsentationen in MS PowerPoint. Ein ähnliches Problem besteht bei der Umwandlung von MS PowerPoint in XML. Analog zu Word handelt es sich beim PPT-Format vor Microsoft Office 2003 um ein proprietäres Binärformat. Da in PowerPoint per se keine Absätze wie bei einer Textverarbeitung existieren, müssen andere Techniken zum Einsatz kommen. Hierzu sind bislang keine Werkzeuge bekannt. Ein an der TU Darmstadt entwickeltes Verfahren [Ni04] basiert auf der COM-Schnittstelle (Component Object Model) bzw. den OLE-Mechanismen (Object Linking and Embedding) [WK94]. Damit ist es nun möglich, PowerPoint-Funktionen von einem externen Programm aus aufzurufen, wodurch sich die Inhalte der Folien auslesen und unter Verwendung eines XML-APIs [SAX][W3C] durch ein Konverterprogramm als XML-Dokument schreiben lassen. Auch hierbei stellt die Notwendigkeit einer lauffähigen PowerPoint-Installation ein Problem dar. Darüber hinaus erfordert diese Transformation prinzipbedingt eine manuelle Nachbearbeitung.

Import von LaTeX-Dokumenten. Das textbasierte, sehr mächtige LaTeX-Format wird häufig für Dokumente aus Forschung und Lehre eingesetzt. Während bereits einige XML2LaTeX-Konverter verfügbar sind, ist der umgekehrte Weg aufgrund des Sprachumfangs und des mangelnden Markup-Charakters schwierig. An der Universität Frankfurt/M. wurde hierfür das frei verfügbare LaTeX-Paket `tex4ht` (TeX for Hypertext) [TeX] eingesetzt, das über den Umweg einer HTML-Umwandlung eine frei konfigurierbare Transformation von LaTeX-Dokumenten nach XML realisiert. Dafür müssen zunächst LaTeX-Konstrukte, die sich von `tex4ht` nicht umsetzen lassen (z. B. Labels und Abbildungen) mit Hilfe von regulären Ausdrücken durch spezielle Makroaufrufe ersetzt werden. Abschließend kann es nötig sein, fehlende Entity-Definitionen durch eine Stylesheet-Transformation in das XML-Dokument zu integrieren.

Import aus anderen XML-Sprachen. Die einfachsten Importmöglichkeiten bestehen für Dokumente, die bereits in einer anderen XML-Sprache codiert sind. Hier lässt sich ein Dokument allein auf Basis von Standard-XML-Werkzeugen, i. A. einer XSL-Transforma-

tion, übersetzen. Aufgrund der variierenden Ausdrucksmöglichkeiten verschiedener Sprachen können Heuristiken notwendig werden, um bestimmte Konzepte der Quell- auf die Zielsprache abzubilden. Da es sich jedoch im eLearning-Bereich in der Regel um dokumentenzentrierte Darstellungen handelt, stimmen die grundlegenden Konzepte wie die Gliederung in einer Paragraphenstruktur oder die Art der verwendeten Inhaltselemente häufig überein, so dass sich eine Abbildung i. A. ohne große Probleme realisieren lässt.

Alle Techniken erzeugen valides XML, das zumindest als Basis für eine Verbesserung des Materials dienen kann. Ein generelles Problem bei der Konvertierung besteht jedoch darin, dass den Formaten in der Regel unterschiedliche Ausdrucksmöglichkeiten zugrunde liegen. Deshalb ist häufig eine manuelle Nachbearbeitung notwendig. Oft unterscheiden sich auch die von den Formaten unterstützten Medienobjekt-Typen, die gerade im eLearning-Bereich ein unverzichtbarer Bestandteil für eine verbesserte Vermittlung des dargebotenen Lehrstoffs sind. Es ist deshalb u. U. erforderlich, weitere Konverter in die Transformation zu integrieren, um die Medienobjekte des Quellformats in unterstützte Medienobjekte des XML-Zielformates zu übersetzen.

3.2 Autorentools

Es liegen zahlreiche Werkzeuge für natives XML-Authoring vor, wie z. B. XMLSpy [Alt]. Für Autoren ohne XML-Kenntnisse sind sie jedoch aus o. g. Gründen nicht geeignet (vgl. Abschnitt 2.2). Das Editieren im WYSIWYG-Modus (beim XMLSpy z. B. durch Anwendung von Stylesheets mit Hilfe des Add-Ons Authentic) ist noch nicht ausgereift.

Darüber hinaus gibt es zahlreiche Editoren für ausgewählte XML-Sprachen, die eine zum Teil sehr hilfreiche Unterstützung für ausgewählte Aspekte bieten. Eine gute Basis für das Editieren unterschiedlicher XML-Formate bieten Arbeitsumgebungen, die dem Nutzer bereits weitestgehend vertraut sind. In den folgenden Abschnitten werden drei Standardwerkzeuge auf Ihre Eignung für das XML-Authoring untersucht: Microsoft Office, OpenOffice.org und Adobe FrameMaker.

MS Office. Die aktuelle Version 2003 von Microsoft Office beinhaltet den Wechsel zu XML als internem Datenformat [MSa]. Allerdings ist diese Funktionalität nur mit den *Professional-* und *Enterprise-Varianten* erhältlich [MSb]. Eigene XML-Schemata können in *Word* genutzt werden. Wie Abbildung 1 zeigt, stehen dazu sowohl eine Ansicht der XML-Dokumentenstruktur (1) als auch einblendbare Tag-Grenzen zur Verfügung (2). Attribute können per Dialog gesetzt werden. Die Validierung von Eingaben erfolgt auf Wunsch sofort während des Editierens.

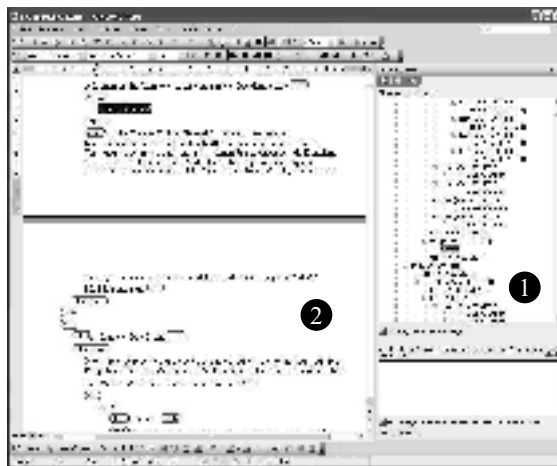


Abbildung 1 MS Word 2003 mit einem XML-Dokument

Die Visualisierung lässt sich durch eine eigene Textformatierung (durch Einsatz einer XSLT-Transformation während des Ladens) an den Kontext der XML-Sprache anpassen. Die Einbindung von Medienobjekten ist über OLE möglich. Mathematische Formeln können durch Software von Drittanbietern in MathML erstellt oder aus dem Microsoft-Format mit einer zu implementierenden Transformation importiert werden. Die Erstellung von Hyperlinks und Querverweisen wird unterstützt. Während des Speicherns eines Dokuments kann bei Bedarf eine weitere Stylesheet-Transformation zur Erzeugung des gewünschten XML-Formats eingesetzt werden.

Trotz des großen Funktionsumfangs findet Word 2003 noch keine breite Akzeptanz als XML-Autorenwerkzeug. Dies liegt auch an der noch nicht sehr langen Verfügbarkeit. Zur Nutzung als Editor für eine spezielle XML-Sprache sind zudem Stylesheets als Import- und Exportfilter nötig, die einen beträchtlichen Entwicklungsaufwand erfordern. Vor allem fällt jedoch die mangelnde Stabilität von MS Word ins Gewicht, denn große bzw. komplexe Dokumente sind wie in früheren Versionen weiterhin problematisch [Ku04].

OpenOffice.org. Das Open-Source-Projekt OpenOffice.org [OOo] findet zunehmende Verbreitung und steigende Akzeptanz. Es enthält u. a. eine XML-fähige Textverarbeitung. Die wachsende Sammlung von Im- und Exportfiltern ermöglicht den Datenaustausch mit anderen Anwendungen. OOo steht auch für Unix-Plattformen zur Verfügung. Neben der Open-Source-Gemeinde steht mit der Firma Sun ein starker Partner hinter dem Projekt.

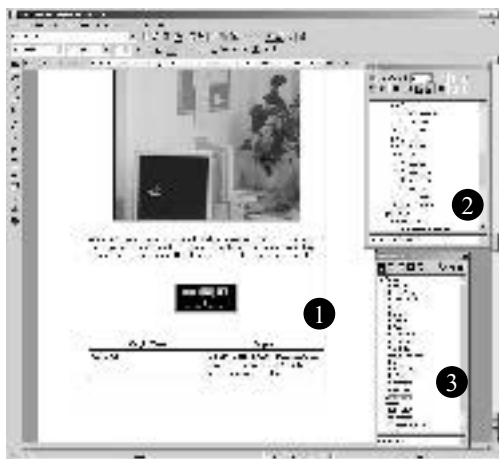


Abbildung 2 OpenOffice mit einem XML-Dokumen

OOo verwendet eigene Document Type Descriptions (DTDs). Mit Hilfe einer Dokumentvorlage und XSL-Stylesheets für den Im- und Export kann der *Writer* als ein WYSIWYG-Editor für XML-Dokumente verwendet werden, wie in Abbildung 2 am Beispiel des Docbook-Formats [Zi03] gezeigt. Dort erkennt man die Dokumentenansicht (1), ein Navigationsfenster (2) und die hierarchische Darstellung der Absatzvorlagen (3). Das Navigationsfenster zeigt verschiedene Ansichten der Elemente des Dokuments, z. B. der Kapitelstruktur, Grafiken und Textbereiche. Um dort eine XML-Struktur zu visualisieren, ist eine Abbildung der jeweiligen Sprachdefinition auf strukturierbare Bestand-

teile wie die Kapitelstruktur erforderlich. Da XML-Schemata nicht unterstützt werden, können Absatzvorlagen zur Abbildung eines Schemas auf eine entsprechend angepasste Formatvorlage oder zur Auswahl aus vordefinierten Attributwerten genutzt werden. Darüber hinaus kann OOo durch eigene Implementierungen erweitert werden. Das API unterstützt die Makroprogrammiersprache Starbasic sowie Java.

MathML-Formeln werden über den internen Formeleditor eingegeben. Querverweise und Hyperlinks können vielfältig eingesetzt werden, und auch die für die Erstellung von Lehr- und Lernmaterialien wichtige Einbindung komplexer Medienobjekte wie Java Applets, Video- oder Audiodateien ist möglich.

Obwohl ein zusätzlicher Aufwand für das XML-Authoring mit OoO besteht, wird dies durch andere Vorteile relativiert. Genannt seien hier u. a. die bestehenden Formatfilter, mit denen z. B. vorhandene Word- und PowerPoint-Dateien für die Umsetzung in ein gewünschtes XML-Format auf einfache Weise importiert werden können, und die native Unterstützung von XML-basierten Dateiformaten wie z. B. SVG und MathML. Dies ermöglicht die Gestaltung einer durchgängigen und konsistenten Authoring-Umgebung.

Adobe FrameMaker.

Das Desktop-Publishing-Programm Adobe FrameMaker erlaubt es Autoren, ein Dokument wie gewohnt zu editieren, es zu strukturieren und seine XML-bezogenen Eigenschaften festzulegen (siehe Abbildung 3). Das Dokumentfenster (1) stellt den formatierten Inhalt des Dokumentes dar, und die Strukturansicht (2) zeigt die Hierarchie seiner Elemente. Beide agieren synchron. Fehler bzgl. der Sprachdefinition werden in der Strukturansicht gekennzeichnet. Das Einfügen von Elementen erfolgt über den Elementkatalog (3). Er enthält stets nur die an der aktuellen Position gültigen Elemente. Die Attribute von Elementen werden in der Strukturansicht angezeigt und können per Dialog bearbeitet werden.

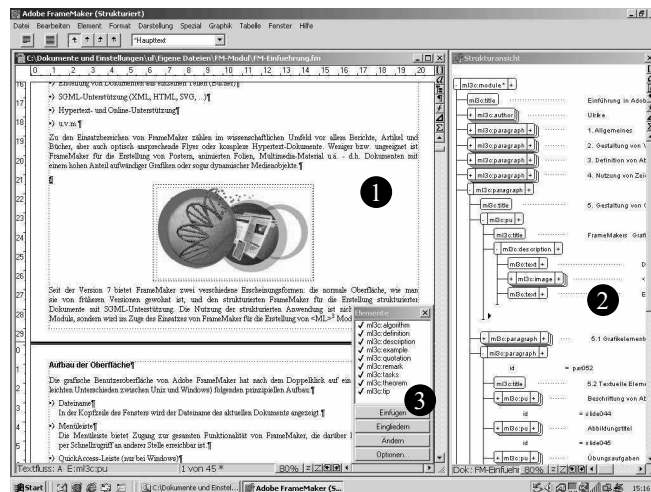


Abbildung 3 Strukturierte Anwendung von Adobe FrameMaker

Werkzeuge zur Bearbeitung strukturierter Dokumente mit FrameMaker werden zu einer strukturierten Anwendung zusammengefasst. Sie enthält eine Strukturbeschreibung in Form einer DTD, Formatierungsregeln, Lese- und Schreibregeln für den Import bzw. Export von XML-Dateien sowie eine Importvorlage. Sollen Schema-basierte Sprachen verwendet werden, ist zunächst eine Konvertierung in ein DTD-basiertes Zwischenformat nötig, bevor das Dokument in FrameMaker modifiziert werden kann. Nach dem Editieren wird es über den Umweg eines anderen DTD-basierten Zwischenformats in die Schema-sprache rückübersetzt. Für die Sprache $\langle ML \rangle^3$ wurden an der Universität Hannover Mechanismen eines derartigen XML-Roundtrippings entwickelt [BO04]. Dabei werden problematische Teile der $\langle ML \rangle^3$ -Definition, wie zum Beispiel gleichnamige Elemente mit verschiedenen Inhalten oder Muster für Attributwerte, auf Mechanismen einer DTD abgebildet und so in FrameMaker importiert.

Die Vorteile von FrameMaker liegen im großen Funktionsumfang, der Verfügbarkeit für verschiedene Plattformen und nicht zuletzt in seiner Stabilität. Als nachteilig sind in jedem Fall die hohen Kosten sowie die eingeschränkten Multimedia-Fähigkeiten zu nennen. Jedoch bietet FrameMaker durch sein auch für herkömmliche Textdokumente bereits sehr gut strukturiertes Konzept die besten Voraussetzungen für ein komfortables Arbeiten mit XML-Dateien.

3.3 Exportfilter

Die abstrakte XML-Beschreibung macht eine Konvertierung in ein anderes Format (d. h. in eine durch den Menschen lesbare Darstellung) erforderlich. Dabei kann zwischen textbasierten Formaten, i. A. ASCII / Unicode, und binären, meist proprietären Formaten unterschieden werden. Jede Klasse erfordert spezifische Transformationstechnologien. Die binären Formate stellen in dieser Hinsicht das größte Problem dar. Der Trend geht jedoch zunehmend in Richtung textbasierte Speicherung, wobei insbesondere XML zum Einsatz kommt. Zahlreiche Konverter sind hier verfügbar.

Stylesheet-Transformationen. Zur Transformation in textuelle Formate kommt speziell die XSL-Sprachfamilie zum Einsatz. Obwohl diese vorwiegend auf Markup-Sprachen wie HTML, XHTML und SVG ausgelegt ist, lassen sich damit auch andere textbasierte Formate, z. B. LaTeX-Dokumente oder ASCII-Dateien, erzeugen. Die Transformation ist dabei in der Regel einstufig, d. h. das Textformat entsteht direkt aus dem XML-Dokument. Insbesondere dynamische Web-Seiten auf Basis von Web-Sprachen wie HTML, ECMA-Script, SVG, CSS oder PHP lassen sich so erzeugen. Inzwischen existieren eine Reihe ausgereifter Werkzeuge für diese Art der Transformation [APa][MSc]. Sie bieten z. T. Erweiterungen, über die sich z. B. Dateien schreiben, Berechnungen auf dem XML-Dokument durchführen und Datenbankzugriffe realisieren lassen [Ja02][Ex]. Der Komplexität der entstehenden Formate sind dadurch kaum Grenzen gesetzt.

Konvertierungsprogramme. Eine andere Möglichkeit der Generierung textbasierter Formate besteht in Transformationen auf Basis einer imperativen Programmiersprache, wie z. B. C, C++ oder Java, vorzugsweise unter Einsatz eines XML-APIs. Dies bietet den Vorteil einer im Vgl. zu interpretierten XSL-Beschreibungen höheren Verarbeitungsgeschwindigkeit. Weiterhin lassen sich beliebige weitere für die Programmiersprache zur Verfügung stehende Bibliotheken nutzen. Dies wird allerdings zu einem immer größeren Teil bereits von den o. g. Erweiterungen der XSL-Prozessoren abgedeckt. Die Nachteile imperativer Sprachen bestehen vor allem in der Notwendigkeit der Überarbeitung des Programms bei Änderungen der XML-Sprache, während bei Verwendung der XSL lediglich eine kleine Anzahl von Templates angepasst werden muss.

Erzeugung von Printmedien. Für Seitenbeschreibungssprachen wie Postscript (PS) oder das Portable Document Format (PDF) als Spezialfälle der textuellen Exportformate können Formatting Objects (XSL-FO) verwendet werden. Prinzipiell ließen sich diese Formate auch mit einer reinen XSL-Transformation erzeugen, allerdings müsste dabei das gesamte Layout manuell gesetzt werden. Stattdessen wird hierfür eine zweistufige Transformation durchgeführt: Im ersten Schritt wird das Dokument mit Layout-Anweisungen angereichert, und im zweiten Schritt werden vom FO-Prozessor das Layout berechnet und das Exportformat generiert. Hierzu existieren zahlreiche Werkzeuge, die jedoch zum großen Teil kommerziell sind. Der freie XSL-FO-Prozessor der Apache Group [Ap04b] bietet für das PDF-Format noch keine vollständige Unterstützung, wird jedoch zunehmend erweitert. Denkbar wäre auch ein Umweg über das LaTeX-Format, für das leistungsfähige Konvertierungs-Tools nach PS und PDF vorhanden sind. Ein entscheidender Vorteil von XSL-FO ist, dass hieraus wiederum verschiedene Exportformate generierbar sind, z. B. PS oder PDF für die Druckvorstufe und SVG für die Präsentation im Web.

Generierung von Binärformaten. Für binäre Formate sind bislang weder Standardtechnologien noch Speziallösungen erhältlich. Daher sollen ausgewählte heuristische Herangehensweisen vorgestellt werden, die für oft verwendete Office-Anwendungen eine Lösung bieten. Sie lassen sich z. T. auch auf andere Formate übertragen.

- Die Elemente, Attribute und Inhalte eines Dokuments können über ein XML-API gelesen und über COM- / OLE-Schnittstellen [WK94] in spezifische Anweisungen der Zielanwendung umgesetzt werden [Ni04]. Da inzwischen viele Windows-Anwendungen (z. B. Corel Draw und Word Perfect) über COM-Schnittstellen verfügen, lassen sich so auch deren Formate erzeugen. Nachteilig ist, dass die Transformationsroutine bei Änderungen der XML-Sprache aufwendig anzupassen ist und stets die installierte Zielanwendung benötigt wird. Leider ist die Dokumentation der COM-Objekte einiger Anwendungen nur dürftig.
- Viele Anwendungsprogramme sind darüber hinaus makrofähig, z. B. MS Office, OpenOffice.org, StarOffice, Corel Draw und Word Perfect. Die Skript-Sprachen sind i. d. R. textbasiert und bieten Zugriff auf alle Funktionen der Applikation und den Dokumenteninhalte. Auf dieser Grundlage lassen sich aus dem XML-Dokument Makros für die jeweilige Zielanwendung generieren, die sowohl die Inhalte als auch die Funktionsaufrufe zum Aufbau des Zieldokumentes beinhalten. Diese werden von der Anwendung verarbeitet, und die Daten lassen sich im nativen Format speichern.

Ein Nachteil aller Transformationen liegt in ihrem großen Aufwand, da auf komplexen Hierarchien gearbeitet wird. Dies führt bei großen XML-Dokumenten bzw. bei einer nicht-linearen Transformation zu einer Verlangsamung. Abhilfe können Verfahren zur Übersetzung von Stylesheets in nativ ausführbaren Programm-Code [APc] schaffen.

Ein weiteres Problem besteht in der variierenden Unterstützung von Medienobjekten durch die Exportformate. Darum müssen u. U. Tools zur Konvertierung der Objekte in ein vom Zielformat unterstütztes Medienformat in die Transformation integriert werden. Dies ist allerdings bei dynamischen oder interaktiven Komponenten schwierig bzw. unmöglich, so dass z. B. Sounds, Videos oder Animationen durch statische Medienobjekte wie Bilder ersetzt werden müssen. In der Sprache <ML>³ ist dies modellierbar.

3.4 Evaluation

Betrachtet man die beiden Methoden zur Erzeugung von XML-basiertem Lehr- / Lernmaterial (die Konvertierung vorhandenen Materials nach XML sowie direktes XML-Authoring), stellt sich die Frage der Eignung im alltäglichen Einsatz. So bestand auch im Projekt Wissenswerkstatt Rechensysteme [WWR] ein hoher Bedarf an Konvertierungs- und Autorentools. Daher wurden zahlreiche vorhandene Werkzeuge untersucht, Weiterentwicklungen durchgeführt und schließlich unterschiedliche Werkzeuge im praktischen Einsatz getestet. Im Rahmen einer projektweiten Evaluation fanden vergleichende Untersuchungen zur Akzeptanz und zum Nutzen derartiger Werkzeuge statt.

Die Importfilter für MS Word, PowerPoint und LaTeX fanden keinen breiten Einsatz, da einige Konzepte der verwendeten Dokumentenbeschreibungssprache <ML>³ nicht vollständig unterstützt wurden und darum eine manuelle Nachbearbeitung nötig war. Eine Ausnahme bildet der Import aus LMML, der zufriedenstellend erfolgte.

Während viele Projektpartner aufgrund ihres technischen Backgrounds die Entwicklung der Materialien auf nativer Ebene direkt in XML vornahmen und mit den verfügbaren Werkzeugen durchaus zufrieden waren, hat von den Desktop-Applikationen nur Adobe FrameMaker eine Akzeptanz gefunden.

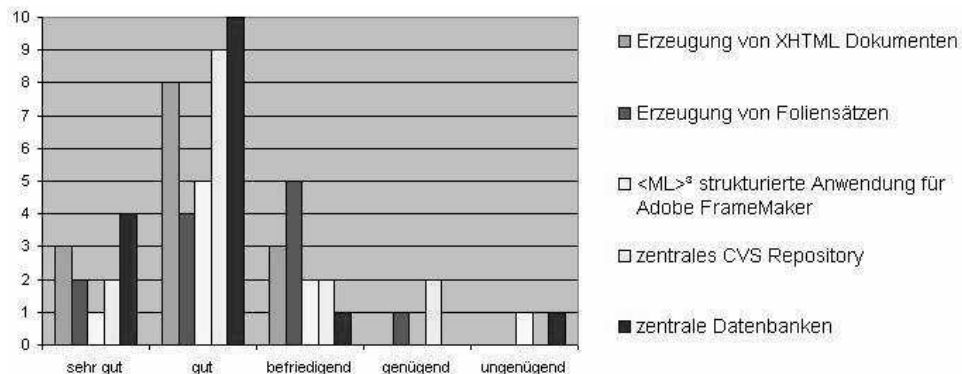


Abbildung 4 Bewertung der meist genutzten Werkzeuge im WWR-Projekt durch die Autoren

Im Bereich der Exportfilter sind im Projekt Konverter nach XHTML, SVG, PDF und PPT entstanden. Diese nutzen Stylesheets, XSL-FO sowie eine OLE-Automation und fanden eine breite Akzeptanz unter den Nutzern.

In Abbildung 4 werden die im Rahmen des Projekts bei der Materialerstellung am häufigsten eingesetzten Werkzeuge gegenübergestellt. Importfilter spielen hier keine bedeutende Rolle, während Adobe FrameMaker und die Exportfilter im Wesentlichen als hilfreich eingeschätzt wurden. Darüber hinaus wird hier insbesondere auch die große Bedeutung einer umgebenden Infrastruktur deutlich: Wenn eine große Zahl von Autoren an aufeinander abzustimmenden Inhalten arbeitet, muss deren Konsistenz durch zentral verfügbare Repositories (hier: Concurrent Versioning System, CVS) und Datenbanken (z. B. für globale Literatur- oder Glossardefinitionen) gewährleistet werden.

4 Zusammenfassung und Ausblick

Die Erstellung XML-basierter Lehr- / Lernmaterialien ist ein komplexer Prozess und wird durch einen interdisziplinären Personenkreis aus den fachlichen, technischen, didaktischen oder künstlerischen Bereichen vorgenommen. Eine weitestgehende Unterstützung durch leistungsfähige, komfortable Werkzeuge ist daher besonders wichtig.

- Da die Wiederverwendung vorhandenen Materials angestrebt wird, sind *Importfilter* zu dessen Umwandlung in die gewählte XML-Sprache unumgänglich. Die Transformation ist jedoch in der Regel komplex und liefert oft keine zufriedenstellenden Resultate, so dass eine aufwändige manuelle Nachbearbeitung erforderlich ist. Diese Vorgehensweise kann bedingt empfohlen werden, falls das Ausgangsmaterial auf ähnlichen Konzepten wie die Zielsprache beruht. Generell sind Konvertierungsverfahren ungeeignet, wenn die Quellen stetig aktualisiert werden und weiterhin nach XML konvertiert werden müssen.
- Zur Modifikation von in der gewählten XML-Sprache vorliegenden Dokumenten bieten gängige *Desktop-Applikationen* inzwischen eine wertvolle Unterstützung. Auch technisch weniger versierten Nutzern wird ein hinreichender Funktionsumfang in einer komfortablen WYSIWYG-Umgebung geboten. Die Entscheidung für ein konkretes Werkzeug fällt auch aufgrund der verwendeten Plattform und bisheriger Gewohnheiten. Im WWR-Projekt wird der Einsatz von FrameMaker favorisiert.
- Um aus dem XML-Format die für das Lernen bzw. die Lehre einsetzbaren Ausgabeformate zu erstellen, existieren bereits vielfältige und leistungsfähige *Exportfilter*.

Aus technischer Sicht sind derartige Verfahren (bis auf die Erzeugung von Drucklayouts) inzwischen ausgereift. Die Transformationswerkzeuge arbeiten vorwiegend auf Basis von Stylesheets und finden eine gute Akzeptanz bei den Nutzern.

Einen nicht zu vernachlässigenden Anteil an der erfolgreichen Erstellung von XML-basierten Lehr- und Lernmaterialien haben grundlegende Infrastrukturen. Ein verteilter Authoring-Prozess erfordert die Unterstützung durch zuverlässige Mechanismen für Datenhaltung und -abgleich. Die Integration der Werkzeuge in eine durchgängige Infrastruktur ist von enormer Bedeutung für den Authoring-Prozess.

Künftige Entwicklungen werden vor allem den Bereich XSL-FO zur Generierung von Seitenbeschreibungssprachen sowie die Erweiterung von Standard-Textverarbeitungen für XML Schema betreffen. Hier sind die technischen Fortschritte der Werkzeuganbieter abzuwarten. Ein weiterer Schwerpunkt der Entwicklungsarbeit sollte auf Konvertierungswerkzeuge zwischen verschiedenen XML-Formaten gelegt werden, um der wachsenden Menge von Beschreibungssprachen gewachsen zu sein. Nur wenn eine Interoperabilität mit etablierten (Quasi-)Standards gegeben ist, haben innovative Formate eine Chance auf breite Akzeptanz; andernfalls werden sie Insellösungen bleiben.

Einige derzeit noch ungelöste Probleme dieses Themenfelds sind eher konzeptioneller Natur. Bei der Erstellung optisch ansprechender Foliensätze aus XML wirkt sich dessen Konzept der grundsätzlichen Trennung von Inhalt und Layout nachteilig aus. Gerade bei Folien bestehen aufgrund des begrenzten Platzes enge Abhängigkeiten, z. B. zwischen Schriftmenge und -größe. Auch die automatische Platzierung von Medienobjekten und Texten ist nicht trivial. Hier sind derzeit keine zufriedenstellenden Lösungen in Sicht.

In diesem Beitrag wurden der Ablauf der Content-Erstellung und damit verbundene Werkzeuge einer kritischen Analyse unterzogen. Diskussionen sollen deren Tragweite für wiederverwendbares, zeitgemäßes Lehr- und Lernmaterial demonstrieren. Abschließend bleibt zu sagen, dass die Erstellung digitaler Lehr- und Lernmedien mit hohem Anspruch verbunden sind und nicht losgelöst von den aktuellen Entwicklungen anderer Bereiche erfolgen darf. So ist z. B. eine Integration in den generellen Prozess der Publikation von digitalen Dokumenten sinnvoll, da hier ähnliche Verfahren und Werkzeuge eingesetzt werden. Darüber hinaus können beide Bereiche gegenseitig voneinander profitieren. Dem eLearning-Material bietet sich durch eine (i.A. mittelfristige) Archivierung in digitalen Bibliotheken oder Medienportalen ein bedeutend größeres Einsatzgebiet. Die bislang i. A. auf fachliche Aspekte fokussierte Erstellung von wissenschaftlichen Publikationen kann von der strukturierten Vorgehensweise im eLearning und der getrennten Berücksichtigung fachlicher, didaktischer und gestalterischer Aspekte profitieren.

5 Literatur

- [Alt] Altova GmbH. <http://www.altova.com/>
- [APa] Apache XML Project: Xalan-Java. <http://xml.apache.org/xalan-j>
- [APb] Apache XML Project: FOP – Formatting Objects Processor. <http://xml.apache.org/fop>
- [APc] Apache XML Project: XSLTC – XSL Stylesheet Compiler and Runtime Processor. <http://xml.apache.org/xalan-j>
- [BO04] J. Brehm, N. Ossipova: <ML>³ Authoring mit FrameMaker. In D. Tavangarian, U. Lucke: Structured eLearning - Wissenswerkstatt Rechensysteme. Universität Rostock, 2004.
- [DS] Design Science: MathType Home Page. <http://www.dessci.com/en/products/mathtype/>
- [Ex] EXSLT Community Initiative. <http://www.exslt.org/>

- [Fr02] B. Freitag: LMML - Eine XML-Sprachfamilie für eLearning Content. 32. Jahrestagung der Gesellschaft für Informatik, Dortmund 2002.
- [GH04] T. Gecks, D. Henrich: Von Microsoft Word zu <ML>³ - Ein Konvertierungswerkzeug. In D. Tavangarian, U. Lucke: Structured eLearning - Wissenswerkstatt Rechensysteme. Universität Rostock, 2004.
- [Ku04] M. Kurzidim: Schreiben ohne Frust. In: c't Magazin für Computertechnik, 07/2004.
- [LTV03] U. Lucke, D. Tavangarian, D. Voigt: Multidimensional Educational Multimedia with <ML>³. World Conference on E-Learning in Corporate, Government, Healthcare and Higher Education (E-Learn) 2003, Phoenix/Arizona/USA.
- [IMS] IMS Global Learning Consortium. <http://www.imsproject.org/>
- [IL] Infinity Loop: UpCast Document Conversion Software. <http://www.infinity-loop.de/>
- [Ja02] J. Jackson: Expand XSL with Extensions – Technique Helps Expand the Capabilities of XSL's Core Features. 2002. <http://www-106.ibm.com/developerworks/library/x-callbk/index.html>
- [Jasc] JascSoftware: WebDraw – The Complete Solution for Creating SVG Graphics and Animation. <http://www.jasc.com/products/webdraw>
- [MSa] Microsoft Developer Network Magazine: XML Files - XML in MS Office Word 2003. <http://msdn.microsoft.com/msdnmag/issues/03/11/XMLFiles/default.aspx>
- [MSb] Microsoft Developer Network Library: Microsoft Office Word 2003 Preview. http://msdn.microsoft.com/library/default.asp?url=/library/en-us/odc_wd2003_ta/html/odc_wdov.asp
- [MSc] Microsoft Corp.: MSXML 4.0 – Microsoft XML Parser. <http://msdn.microsoft.com/xml,2004>
- [Ni04] V. Nimbler: Automatische Erzeugung von MS PowerPoint aus <ML>³. In D. Tavangarian, U. Lucke: Structured eLearning - Wissenswerkstatt Rechensysteme. Universität Rostock, 2004.
- [OOo] OpenOffice.org. <http://www.openoffice.org/>
- [Ri03] U. Rinn, K. Bett, J. Wedekind, P. Zentel, D. M. Meister, W.F. Hesse: Virtuelle Lehre an deutschen Hochschulen im Verbund, Teil 1: Eine empirische Untersuchung der Projektkonzeptionen von Vorhaben zur Förderung des Einsatzes Neuer Medien in der Hochschullehre im Förderprogramm Neue Medien in der Bildung. Institut für Wissensmedien, Tübingen 2003.
- [SAX] The SAX Project: SAX – Simple API for XML. <http://www.saxproject.org/>
- [SCH] SCHEMA Electronic Documentation Solutions: Word als HTML-/XML-/SGML-Editor mit dem MarkupKit. <http://www.schema.de/sitehtml/site-d/htmlexpo.htm>
- [SF] Sourceforge.net: Sodipodi. <http://www.sodipodi.com/>
- [TeX] TeX4ht: LaTeX and TeX for Hypertext. <http://www.tug.org/applications/tex4ht/mn.html>
- [WK94] S. Williams, C. Kindel: Component Object Model - Technical Overview. Microsoft Developer Network, 1994. http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dncomg/html/msdn_compr.asp
- [WM99] N. Walsh, L. Mueller: DocBook - The Definitive Guide. O'Reilly, 1999.
- [WWR] Wissenswerkstatt Rechensysteme. <http://www.wwr-project.de/>
- [W3C] World Wide Web Consortium. <http://www.w3c.org/>
- [Zi03] S. Zic: OpenOffice and DocBook Tutorial. 2003. <http://www.zzoss.com/projects/oowdbk/openoffice/oowdbk.pdf>