

Wiederverwendungsorientiertes Content Authoring nach dem Single-Source Prinzip

Ludger Thomas, Eric Ras

Education and Training
Fraunhofer Institut Experimentelles Software Engineering (IESE)
Fraunhofer-Platz 1
67663 Kaiserslautern
{ludger.thomas, eric.ras}@iese.fraunhofer.de

Abstract: Auf Basis einer Beschreibung des aktuellen Stands der Content Entwicklung wird aufgezeigt, dass die Wiederverwendung von Informationen und Lerninhalten zwar ein allgemein akzeptiertes Problem der Praxis darstellt jedoch die am häufigsten verwendeten Werkzeuge nicht dafür ausgelegt sind, dies auch zu unterstützen. Darauf aufbauend werden Prinzipien einer Wiederverwendungsstrategie im Kontext des Single-Source-Ansatzes entwickelt. Es wird ein Vorhaben vorgestellt, das mit Standard XML-Technologien und Freier Software einen wiederverwendungsorientierten Single-Source Multiple-Purpose-Publishing-Ansatz für die Lerninhalterstellung umsetzt.

1 Stand der aktuellen Content Entwicklung

Im Bereich E-Learning liegt gegenwärtig der Schwerpunkt der einschlägigen Fachpublikationen allem Augenschein nach auf dem, was mit dem Terminus E-Learning 2.0 bezeichnet wird [Do05]. Hinter diesem Begriff verbirgt sich bei genauerer Betrachtung kein einheitliches Konzept, sondern vielmehr ein Sammelsurium unterschiedlichster Konzepte und Technologien, von Social Software, Blogging und Wikis bis hin zum informellen Lernen und individuellen Wissensmanagement.

Abseits der Modeworte, ist für die große Mehrheit der Praktiker, die E-Learning in Unternehmen und Universitäten einsetzen, unverändert das Motto „Content is the king“ zutreffend. Das jedenfalls belegen die wenigen empirischen Ergebnisse, die sich mit dem aktuellen Stand von Content-Entwicklung und -Einsatz befassen. So zeigt beispielsweise eine Studie der eLearningguild [El05], dass drei Viertel der Befragten der o.g. Aussage nach der Bedeutung von Content für das E-Learning in der eigenen Organisation zustimmen und auch die dabei eingesetzten Content-Typen überwiegend konventionell sind: Es dominieren Tests und konvertierte Inhalte, gefolgt von asynchronem Lernen, Präsentationen, Audioaufzeichnungen und Simulationen. Auch belegen aktuelle Projekte, wie „Contentforum“ oder „EXPLAIN“, dass die Erstellung und Wiederverwendung von E-Learning Content im industriellen Umfeld nach wie vor große Bedeutung besitzt und sich E-Learning 2.0 in diesem Umfeld noch nicht etabliert hat.

1.1 Gegenwärtige Praxis des Courseware Authoring

Mittlerweile liegen zahlreiche Ansätze vor, wie die Content Erstellung nach ingenieurmäßigen Prinzipien im Bereich E-Learning aussehen kann ([DIN04] [GPR02]), oder wie sich die Entwicklung von Rich-Media-Content in industriellen Kontexten effizienter gestalten lässt [ZK03]. Ebenso wird zunehmend die Rolle des Software Engineering im Kontext der Content-Produktion diskutiert [POS05a]. Auf der Werkzeugebene scheinen sich die dabei diskutierten Prinzipien jedoch noch nicht etabliert zu haben. Gerade die schwache Unterstützung der Wiederverwendung von Inhalten, sowie die fehlende Prozessunterstützung scheinen Schwachpunkte der aktuellen Werkzeuge zu sein. In einer Untersuchung der elearningguild [El05] geben 40% der 440 Befragten, überwiegend Instructional Designer aus Kleineren und Mittleren Unternehmen, an, dass bei den aktuell verwendeten Werkzeugen die Aspekte „Content re-use and re-purposing“ und „More instructional design process in the tools“ verbessert werden sollten. Die Untersuchung gibt auch weitere Aufschlüsse darüber, welche Werkzeuge in der Entwicklungspraxis genutzt wurden. So wurden 2005 Flash (Adobe, 74%), PowerPoint (Microsoft, 62%), Dreamweaver (Adobe, 60%), Captivate (Adobe, 46%) und Authorware (Adobe, 30%) als am häufigsten genutzte Programme genannt [El05]. Vergleichbare Ergebnisse konnte auch schon Brandon-Hall [Br01] 2001 vorlegen: Nach seinen Untersuchungen waren PowerPoint und Word die am verbreitetsten genutzten Werkzeuge für die Erstellung von E-Learning Content, gefolgt von Dreamweaver und Flash. Auch eine Umfrage von Elliot Masie [Ma01] erbrachte das Ergebnis, das „more and more of E-Learning content being authored with everyday document tools.“ So gaben 85% der 1615 Rückläufe an, Office Produkte für die Erstellung von E-Learning content zu benutzen, gefolgt von Tools für die Web-Entwicklung (71%) sowie spezielle Authoring-Umgebungen wie Authorware oder Toolbook (52%). Die Ergebnisse der vorgelegten Untersuchungen weichen in einzelnen Details voneinander ab; dennoch sind gewisse Gemeinsamkeiten erkennbar:

- Beim Courseware Authoring dominieren proprietäre Werkzeuge und Technologien, die nicht speziell für die Erstellung von E-Learning Content entwickelt wurden, die den Autoren jedoch aus anderen Arbeitskontexten vertraut sind. Engineering wird nicht oder nur ansatzweise unterstützt.
- Die zweitgrößte Gruppe bilden proprietäre, speziell für die Herstellung von interaktiven Lerninhalten ausgerichtete Werkzeuge.
- Die dritte Gruppe bilden Werkzeuge für die schnelle kostengünstige Erstellung von Inhalten, in Form von Formatkonvertierungen, (engl. Content Conversion) oder für die Aufzeichnung von Vorträgen und Bildschirminhalten.
- Die aktuellen Untersuchungen geben keinen Aufschluss darüber, welchen Stellenwert „selbstentwickelte“ Lösungen mit HTML besitzen. Während diese Vorgehensweise bei Brandon Hall mit noch 34% vertreten war, wird sie in der aktuellen Studie nicht mehr oder nur implizit aufgeführt.

Betrachtet man die o.g. Ergebnisse und stellt sie einer der zentralen Aussage des Untersuchungsergebnisse der elearningguild [E105] gegenüber, nämlich dem ermittelten Bedarf nach „re-use“ und „re-purposing“, fällt eine gewisse Ambivalenz ins Auge: So ist keines der meist verwendeten Werkzeuge dafür ausgelegt, Inhalte wiederzuverwenden noch diese in andere Kontexte einzusetzen. Eine modulare Datenhaltung ist kaum möglich und selbst einfache Änderungen am Aussehen der Endprodukte lassen sich z.T. nur mit hohem Aufwand durchführen. Etwas überspitzt kann man formulieren: Die populärsten Werkzeuge für das Erstellen von Lerninhalten unterstützen nicht oder nur bedingt das, was die Nutzer mittel- oder langfristig benötigen: Nämlich „re-use“ und „re-purpose“ von Trainingsinhalten. Hier laufen die E-Learning Produzenten also Gefahr, Werkzeuge einzusetzen, die gerade mittel- oder langfristige Bedürfnisse nicht befriedigen und keine ausreichenden Investitionssicherheiten bieten. Insbesondere in Hinblick auf die Zukunftssicherheit der erzeugten Inhalte erscheint deshalb die Verwendung von weltweit verwendeten *Standards* im XML-Bereich als sinnvoll. Diese offenen Austauschformate werden von einer ständig wachsenden Zahl von Werkzeugen unterstützt und können über XSL in zahlreiche andere Formen überführt werden. Durch Standard Technologien wie XML-Inclusion (W3C) können die Daten mit einfachsten Mitteln modular vorgehalten werden, sodass einer Wiederverwendung von einzelnen Inhalten im Rahmen eines Single-Source Ansatzes nichts entgegensteht.

2 Content-Produktion nach dem Single-Source-Prinzip

Im Bereich des elektronischen Publizierens ist die Erstellung von Inhalten nach dem Single-Source-Prinzip gegenwärtig Stand der Technik. Zentrales Prinzip hierbei ist, Informationen einmal abzulegen, aber mehrfach in unterschiedlichsten Medien und Kontexten zu verwenden. Um dies zu erreichen, wird bei den Inhalten eine weitgehende Trennung von Inhalt-, Struktur- und Layoutkomponenten vorgenommen. Bei geschickter Separation dieser drei Dimensionen können Inhaltsbausteine zu neuen Dokumenten re-kombiniert werden. Durch diese Re-komposition und die text-typische Serialisierung von Informationen sowie eine anschließende Transformation können verschiedene Ausgabeformate, z.B. Handbücher, Hilfe-Systeme oder Lernmedien, für verschiedene Endgeräte, Produkte, Zielgruppen oder Betriebssysteme aus einer einzigen Quelle erzeugt werden. Der gesamte Vorgang gliedert sich grob in die folgenden sechs Phasen:

1. **Analyse und Planung** der Inhalte sowie der Informationsmodellierung
2. **Modularisierung** in Inhaltsbausteine und semantische Modellierung
3. **Speicherung** der Inhaltsbausteine
4. **Re-Komposition** und **Serialisierung** der Bausteine zu einem neuen Dokument
5. **Transformation** des Dokuments in das Ausgabeformat
6. **Präsentation** des fertigen Dokumentes

Insbesondere die beiden ersten Phasen des Prozesses verlangen dabei eine tiefgehende Analyse der vorhandenen oder geplanten Informationen sowie eine Wiederverwendungsstrategie. Die hierbei adressierten Probleme entsprechen in einigen Aspekten denen, die auch für den Bereich der Lernobjekte diskutiert werden [DH03].

Im Vergleich zu herkömmlichen Ansätzen des elektronischen Publizierens, beispielsweise mit layout-orientierten DTP-Programmen, bieten Single-Source-basierte Ansätze unter Verwendung von semantischen Auszeichnungen i. A. folgende Vorteile:

- **Effizientere Wartbarkeit:** Nach dem Single-Source Prinzip werden Informationen nur einmal gespeichert aber mehrfach verwendet. Daher müssen Inhalte im Idealfall immer nur an einer zentralen Stelle geändert werden. Die Änderungen wirken sich dann automatisch auf alle Dokumente aus, die die Ursprungsinformation inkludieren. Hierdurch reduzieren sich die Wartungsaufwände für Dokumente auf der inhaltlichen Ebene wesentlich.
- **Erhöhte Wiederverwendung:** Durch die modulare Datenhaltung, sowie die semantische Auszeichnung, kann bei entsprechender inhaltlicher Konzeption eine erhöhte Wiederverwendung von Inhalten begünstigt werden. Dies führt i.A. zu einer erhöhten Produktivitäts- und Qualitätssteigerung. Neben der freien Wahl des Medienformats, sieht Gersdorf [Ge03] gerade in der Wiederverwendung von Inhalten in unterschiedlichen Kontexten, dem sog. Single-Source-Multiple-Usage Publishing, den größten ökonomischen Effekt. So ermöglicht der Ansatz das „Publizieren für verschiedene Zielgruppen, mit unterschiedlichen Detaillierungsgrad oder auch in unterschiedlichen Sprachen.“ (ebd.)
- **Layoutunabhängigkeit:** Studien berichten, dass Autoren mehr als 60% ihrer Zeit damit verbringen, ihre Dokumente zu finden, zu strukturieren und zu formatieren [Za03]. Laut dieser Zahlen werden also nur 40% der Zeit darauf verwendet, wirklich Inhalte zu erstellen. Selbst wenn diese Zahlen mit Vorsicht zu genießen sind, ist zu vermuten, dass ein Single-Source-basierter Ansatz auf Basis von XML-Datenformaten in Bezug auf die Verteilung der Aufwände Vorteile bieten kann, da neben Texterstellung und semantischer Auszeichnung keine weiteren Layoutarbeiten erforderlich sind.

Es liegen bereits mehrere Arbeiten vor, die eine Übertragung des Single-Source-Prinzips vom Bereich Content Management auf den Bereich der Lernmedienproduktion zum Gegenstand haben ([Ge03] [LTV02] usw.). In der konkreten Umsetzung der zugrunde liegenden Vorhaben wurde dabei überwiegend eine eigens entwickelte Datenstrukturierung, beispielsweise MI3 [LTV02], verwendet. Der vorliegende Ansatz baut auf den Erkenntnissen der Arbeiten auf, verwendet aber in der praktischen Umsetzung einen verbreiteten und standardisierten XML-Dialekt anstatt eines proprietären Formats. Dies bietet u. a. den Vorteil, dass eine Fülle frei verfügbarer Werkzeuge existiert, mit deren Hilfe Informationsbausteine erstellt, komponiert und transformiert werden können.

3 Strategische Wiederverwendung als Bestandteil des Single-Source-Ansatzes

Im Kontext von Single-Source-Multiple-Purpose-Publishings versteht man unter dem Begriff Wiederverwendung die Wiederverwendung von Inhalten und Prozessen innerhalb und zwischen der Dokumentation von Software Systemen (angelehnt an [FK05, RB91]). Nicht nur die Wiederverwendbarkeit der Inhalte selbst, sondern auch die Erfahrungen der Mitarbeiter in Bezug auf Wiederverwendung – Was funktioniert? Was nicht? - sind dabei Bestandteile einer Wiederverwendungsstrategie. Bei vergleichender Betrachtung, lassen sich zwischen Wiederverwendung im Software Engineering und der Content Entwicklung beispielsweise im Bereich des E-Learning Gemeinsamkeiten erkennen: In beiden Bereichen werden Inhalte modularisiert, zum Teil abstrahiert, dekontextualisiert und standardisiert, Domänen werden modelliert und Produktfamilien definiert, Ähnlichkeiten und Variabilitäten werden festgelegt, Inhalte werden klassifiziert in unterschiedlichen Datenbanken abgelegt und nach einer evtl. Anpassung wieder verwendet. Dieser Abschnitt soll daher zentrale Erfahrungen aus dem Software Engineering darstellen und hieraus Hinweise für schnelle Erfolge bei der Wiederverwendung im Bereich der Content-Entwicklung ableiten. Nur bei Vorliegen entsprechender Strategien, können Technologien und Methoden ihre vollen Potenziale entfalten.

Die folgenden Richtlinien basieren auf einer Analyse von in der Fachliteratur dokumentierten Erfahrungen, die bei der Wiederverwendung von Software Produkten, Prozessen und Informationen im Software Engineering bisher gemacht wurden, sowie auf eigenen Erfahrungen in Projekten. Sie können helfen eine geeignete Wiederverwendungsstrategie im Kontext des Single-Source-Ansatzes zu entwickeln:

- **Wiederverwendungspotenzial frühzeitig bestimmen** [MET02]: Oft werden erhoffte Erfolge schnell durch die Einsicht, dass das Potential der Wiederverwendung zu hoch eingeschätzt wurde, gemindert. Deswegen soll eine Analyse auf Ähnlichkeiten der zu dokumentierten Software Produkte, deren Funktionalitäten und der Zielgruppe der Dokumentation durchgeführt werden. Eine zu geringe Überschneidung führt zu weniger Inhaltswiederverwendung.
- **Zustimmung des Managements einholen** [MET02]: Neben einer klaren Zustimmung des Managements zur projekt- oder produktübergreifenden Wiederverwendung soll, vor allem am Anfang der Einführung von Wiederverwendung, Zusatzaufwand eingeplant werden. Die Wiederverwendung erfordert ein grundsätzliches Umdenken der Mitarbeiter und eine Einlernphase. Arbeitsprozesse müssen für die Wiederverwendung angepasst werden und eine technische Infrastruktur muss aufgesetzt werden.
- **Schnelle und messbare Erfolge durch Fokussierung** [FK05, RB91]: Es ist wichtig, am Anfang die Wiederverwendung auf kleine Bereiche einzuschränken, in denen schnelle Erfolge mit kalkulierbarem Aufwand zu erwarten sind. Dabei ist es wichtig Messdaten zu erfassen, wie z.B. der Grad der Wiederverwendung und Anpassung, Qualität des Endproduktes etc.

- **Entwicklung für die Wiederverwendung** [RB91]: Die Entwicklung *mit* Wiederverwendung von Inhalten erfordert auch eine Entwicklung *für* die Wiederverwendung. Inhalte, die für die spätere Wiederverwendung geeignet sein sollen, müssen in Hinblick auf Standardisierung, De-Kontextualisierung und Wahl der Granularität geplant und entwickelt werden.
- **Wiederverwendung von Produkten, Prozessen und Erfahrungs- und Domänenwissen** [RB91][LL97][MET02]: Ein häufiger Fehler bei der Wiederverwendung ist die reine Fokussierung auf Inhalte – Prozesse und Wissen werden vernachlässigt. Bestimmte Prozesse oder Aktivitäten (z.B. Anpassungsmethoden, Attribut oder Vokabularauswahl), die sich bewährt haben, sollten abgelegt werden. Des Weiteren sind Erfahrungen und Wissen über die Wiederverwendung von Inhalten oder Prozessen sehr hilfreich.

Auch im Bereich der Domänenmodellierungen können bewährte Prinzipien des Software Engineerings auf die Entwicklung von Content übertragen werden. Ansätze die diese Prinzipien umsetzen (z.B. Produktlinienansatz, komponentenbasierte Entwicklung) werden dabei bislang in der Praxis nur eingeschränkt umgesetzt [RMW05] [POS05a]. Im Folgenden werden grundlegende Anleitungen zur Umsetzung dieser Prinzipien anhand eines Beispiels illustriert. Abbildung 1 zeigt einen Ausschnitt eines wiederverwendbaren Inhaltsbausteins einer Produktdokumentation.

- **Bildung von Varianten:** Für Softwareprodukte einer Produktfamilie (z.B. Microsoft Produkte) unterscheiden sich die Handbücher, Online-Hilfen, FAQ, und Tutorials. Trotzdem sind viele Funktionalitäten dieser Produkte ähnlich und Dokumentationsinhalte können durch Variantenbildung effizient wiederverwendet werden. Der Produktlinienansatz aus dem Software Engineering schreibt vor, die Menge an Varianten auf die wichtigsten einzuschränken (siehe [POS05a] im Kontext für Lernobjekte). Durch das Verwenden von Attributen können Dokumentationsvarianten für unterschiedliche Zielgruppen (Anfänger, Experte) Zwecke (Online-Kurs, FAQ, PDF-Handbuch) oder Domänen (Automotive, IuK) definiert werden (siehe Zeile 2 in Abb. 1). Wichtig an dieser Stelle ist es, ein vordefiniertes Attributvokabular festzulegen, um die Konsistenz zwischen den Inhaltsbausteinen zu bewahren (Standardisierung). Durch das Setzen dieser Attribute können beim Profiling die Varianten automatisch erzeugt werden (siehe nächsten Abschnitt).
- **Wiederverwendung von Inhaltsbausteinen:** Durch das Definieren von so genannten Entities (z.B. `<!ENTITY product_name "Microsoft Word">`), können wiederkehrende Texte oder Objekte (z.B. Grafiken von Buttons) konsistent im ganzen Dokument durch Referenzierung wiederverwendet werden (siehe Zeile 6 & 30 in Abb. 1). Dies erleichtert ebenfalls die Wartung, weil die Daten nur an einem Ort geändert werden müssen. Eine einheitliche Nomenklatur der Entities muss hier eingehalten werden. Entities sind verbindlich von den Mitarbeitern zu verwenden.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <section level="advanced" purpose="course" domain="automotive">
3   <info>
4     <author>Klaus Mustermann</author>
5     <relatedProduct>
6       <title>&product_name;</title>
7       <release>
8         <number>1.0</number>
9       </release>
10    </relatedProduct>
11    <revhistory>
12      <revision>
13        <number>0.1</number>
14        <date>23.03.06</date>
15        <revdescription>Editing history</revdescription>
16        <revstatus>verified</revstatus>
17      </revision>
18    </revhistory>
19    <reuse>
20      <judges>
21        <judge>
22          <date>27-03-06</date>
23          <usefulness>3</usefulness>
24          <quality>2</quality>
25        </judge>
26      </judges>
27      <number-success>2</number-success>
28      <number-fail>0</number-fail>
29    </reuse>
30    <abstract>&product_abstract;</abstract>
31    <keywords>Keyword</keywords>
32    <relatedUserTask>T-01-2</relatedUserTask>
33    <relatedObjectives>O-01-2</relatedObjectives>
34  </info>
35  <title>Saving the document ...</title>
36  <xi:include href="generalremarks.xml"/>
37  <para>...</para>
38 </section>

```

Abbildung 1: Beispiel

- **Klassifizierung von Inhaltsbausteinen:** Bei der Wiederverwendung spielt das Auffinden von Content eine zentrale Rolle. In der Software Wiederverwendung hat sich hier die Facettenklassifizierung bewährt [P90]. Die Objekte werden anhand von Facetten klassifiziert. Vorteil dieser Schemata sind ihre Skalierbarkeit, die Verwendung von Standardvokabularen und die Anwendung von ähnlichkeitsbasierten Suchverfahren und Thesaurus. Als Facetten werden die Attribute für die Variantenbildung, als auch Attribute die den Bezug zum Produkt und dem Endbenutzer herstellen, verwendet (z.B. <name>, <release> oder <relatedUserTask>:Bezug zur Funktionalität des Produktes T-01-2 „Speichern“, siehe Zeile 6-9 & 32 in Abb. 1).
- **Beurteilung der Qualität von Inhaltsbausteinen:** Es ist wichtig für den Autor die Qualität des Inhaltsbausteins im Sinne von Wiederverwendbarkeit beurteilen zu können. Auch in der Software Entwicklung haben sich bestimmte Verfahren bewährt. Wichtig für die Qualitätsbeurteilung ist die Historie des

Inhaltsbausteines, die beschreibt, wer wann den Baustein angepasst oder verwendet hat (siehe Zeile 4 & 11-18). Die Verwendung eines eindeutigen Vokabulars zum Status des Bausteins (z.B. draft, verified, etc.) gibt ebenfalls Auskunft über die Qualität des Bausteins (siehe Zeile 16). Weitere wichtige Informationen für die Einschätzung der Wiederverwendbarkeit sind Angaben von anderen Kollegen zu Nützlichkeit und Qualität dieser Bausteine, Anzahl der erfolgreichen Wiederverwendungen, Anzahl der Misserfolge bei der Wiederverwendung, oder sogar aufzubringende Aufwände bei der Anpassung (siehe Zeile 19-29). Zum größten Teil können die Informationen automatisch vom System eingebracht und gepflegt werden.

4 Umsetzung auf Basis von „DocBook“ und Freier Software

Bei DocBook [WM99] handelt es sich um ein mehr als 400 Elemente umfassendes standardisiertes XML-Format für (Software) Dokumentation. Es wurde Anfang der 90er im Verlagsumfeld entwickelt und hat seitdem die Entwicklung der XML-Technologien nachhaltig beeinflusst. Neben Unternehmen wie SUN benutzen beispielsweise der Verlag O'Reilly, sowie unzählige Open-Source Projekte das DocBook Format bei der Erstellung ihrer Dokumente. Einen wesentlichen Mehrwert bei der Arbeit mit DocBook stellt insbesondere eine frei verfügbare, umfangreiche XSL-Distribution dar, mit deren Hilfe DocBook-XML in andere Ausgabeformate wie HTML und PDF umgewandelt werden kann; einschließlich Index, Inhaltsverzeichnis, Glossare, Bibliographien usw.

Das grundlegende Paradigma für die Strukturierung von Dokumenten in DocBook basiert auf dem einer sequentiell-hierarchischen Anordnung von Informationsblöcken (Kapitel, Sektion, Unter-Sektion usw.). Durch die Vielzahl der verfügbaren Strukturelemente lassen sich auch größere und komplexe Dokumentenzusammenstellungen modellieren. Auf der Ebene der semantischen Auszeichnung von Inhalten, bietet DocBook eine Vielzahl speziell auf die Softwaredokumentation abgestimmter Elemente. So gibt es beispielsweise neben Elementen wie guitem oder guimenu spezielle Auszeichnungsmöglichkeiten für Source-Code sowie darin enthaltene Erläuterungsmarken. Die Modularisierung von Informationen in DocBook kann auf technischer Ebene durch Entities sowie XML-Inclusion (W3C) realisiert werden. Durch diese Mechanismen können beliebig feingranulare XML-Fragmente zu einem großen Ganzen integriert werden. Durch geschickte Auslagerung und Inklusion kann so mit einfachsten Mitteln ein Text Repository geschaffen werden, dessen Elemente in beliebig vielen Dokumenten verwendet werden können. Für die kontextspezifische Komposition und Aufbereitung der Informationen, sowie die Spezialisierung von Elementen, stellt DocBook für jedes Element eine Reihe von Attributen zur Verfügung.

Im Vergleich zu anderen Ansätzen aus dem Bereich der Educational Modeling Languages (EML) (z.B. [LTV02]), die auf eigenen Datenformaten aufsetzen, zeigt sich, dass über Erweiterungen und Spezialisierungen viele der Ansätze auch mit DocBook realisierbar sind. So können beispielsweise über das Attribut *role* spezielle Verwendungen für ein Element definiert werden, also z.B. `<para role="objective">` für einen Absatz der ein Lernziel enthält. Durch weitere Attribute kann man das Element weiter spezifi-

zieren: So ergibt `<para role="objective" condition="course" userlevel="beginner">` beispielsweise ein Element, das ein Lernziel für einen Online-Kurs auf dem Niveau „beginner“ markiert.

Auf technischer Ebene lassen sich bei der Umsetzung von DocBook-basierten Single-Source-Ansätzen grundsätzlich vier aufeinander aufbauende Schichten (layer) unterscheiden, die bei der technischen Erstellung und Aufbereitung von Dokumentationen nacheinander durchlaufen werden:

1. **Datenhaltung** (storage layer): In dieser Schicht werden die Quelldateien und weiteren Bestandteile der Dokumentationen in DocBook-XML erstellt und für die weitere Verarbeitung vorgehalten.
2. **Komposition** (composition layer): Diese Schicht dient der Aggregation der Bestandteile zu einem validen DocBook-Dokument (profiled intermediate-File), das als Grundlage für die weiteren Verarbeitungsprozesse dient.
3. **Transformation** (transformation layer): Auf dieser Ebene werden die aggregierten Dokumente regelbasiert in ein Ausgabeformat (PDF, HTML) überführt.
4. **Präsentation** (presentation layer): Die fertig formatierten Dokumentationen werden in dieser Schicht den Benutzerinnen und Benutzern präsentiert.

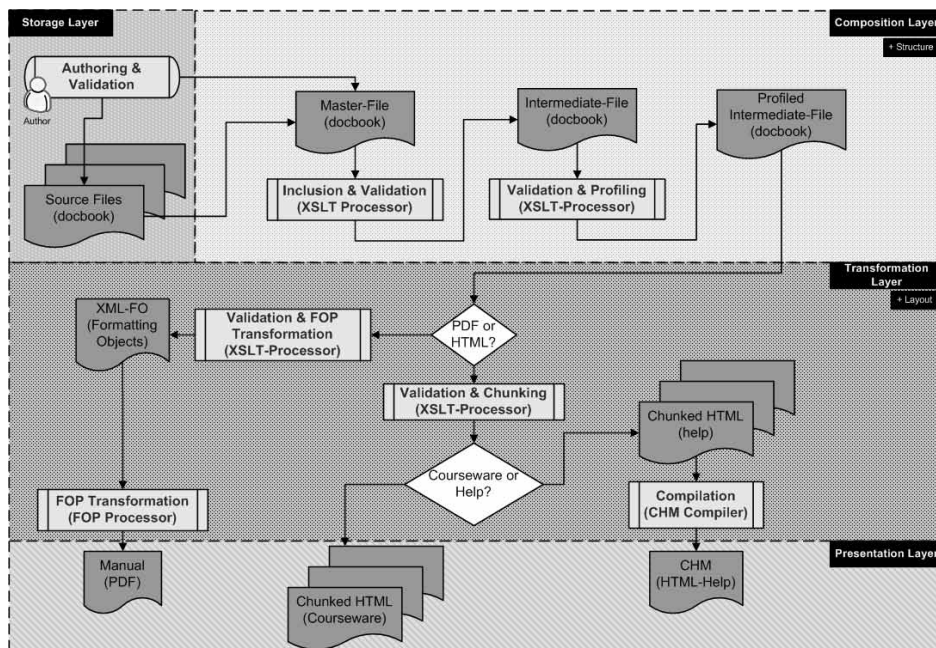


Abbildung 2: Architektur auf Basis von DocBook [TR05]

Die genannten Schichten und Vorgänge werden im Fall von DocBook durch eine Vielzahl frei verfügbarer Werkzeuge unterstützt: So erlauben z.B. die frei verfügbaren Versionsverwaltungssysteme die Speicherung der erzeugten Dateien. Über die ebenfalls freien XSLT-Prozessoren können die Dateien aggregiert und transformiert werden. Auch für die Erstellung von PDF-Dokumenten aus XSL-FO Dateien steht freie Software zur Verfügung. Diese steht im Leistungsumfang kommerziellen Produkten jedoch deutlich nach. Auch für die Erstellung von DocBook Dateien existieren mehrere freie und kommerzielle Werkzeuge, die sich grob in zwei Klassen aufteilen lassen: Zum einen die Gruppe der klassischen XML Editoren, die über ausgereifte Funktionalitäten zur Erstellung von XML Source-Code verfügen. Zum anderen diejenigen Tools, die den Source-Code möglichst vor dem Benutzer verbergen und ihm stattdessen die Arbeit in einer Art WYSIWYG Ansicht ermöglichen. Bei den kommerziellen Werkzeugen gibt es darüber hinaus hybride Produkte, die beide Ansichten miteinander vereinen.

5 Diskussion des Ansatzes

Der entscheidende Vorteil von DocBook gegenüber proprietären XML-Formaten, wie einer der EMLs, besteht darin, dass mit DocBook ein *standardisiertes* Format zur Verfügung steht, das weltweit genutzt und von zahlreichen Werkzeugen unterstützt wird. So bietet beispielsweise das weit verbreitete OpenOffice einen entsprechenden Im- und Exportfilter an. Ebenso gibt es zahlreiche freie und kommerzielle XML-Editoren, die das Format erzeugen, validieren und transformieren können. Der zweite große Vorteil besteht in der Offenheit des Formates selbst, das individuelle Änderungen und Anpassungen erlaubt und somit dem Nutzer die Möglichkeit gibt, die gewünschten Informationen angemessen zu modellieren. So können auch einige Schwächen des Formates in Bezug auf die Erstellung von Lerninhalten ausgeglichen werden. Auch auf der Ausgabeseite zeigt sich DocBook offen: So können mit relativ geringem Aufwand Transformationen von DocBook in die häufig verwendeten E-Learning Formate (IMS-CP/LD, SCORM) vorgenommen werden [TR05]. Schwächen zeigt DocBook in dem zugrunde liegenden Kontext vor allem auf der Ebene der Modellierung von Übungsaufgaben. Diese können erst im zukünftigem DocBook Standard V5 in das Schema integrieren werden, ohne die Validität des Dokumentes zu zerstören. Aktuell müssen sie extern erstellt werden, um dann im Rahmen des Transformationsprozesses eingebunden zu werden.

Auch in Bezug auf die Wiederverwendung von Inhalten stellt XML bzw. DocBook alle notwendigen Mechanismen bereit, um dies technisch umzusetzen. Die zentrale Herausforderung auf dieser Ebene besteht vielmehr darin, eine domänen-, produkt- oder projektübergreifende Wiederverwendungsstrategie zu definieren und auf organisatorischer und inhaltlicher Ebene umzusetzen. Aufgrund vorliegender Erfahrungen aus den Bereichen Softwareentwicklung oder auch Learning Objects, muss jedoch davon ausgegangen werden, dass Potenziale in der Praxis prinzipiell begrenzt sind. So nennt beispielsweise Pankratius [POS05a] in seiner Übertragung der Software-Productlines-Prinzipien auf das E-Learning allgemeine Informationen, wie Layout oder nicht näher definierte pädagogische Richtlinien als Beispiele für mögliche Wiederverwendungen:

„For example, possible crosscutting features in an educational context are layout files, pedagogical guidelines, security, digital rights, or pricing information.“ [POS05a]

Die frei verfügbaren Werkzeuge für die Umsetzung des Single-Source-Ansatzes mit DocBook haben sich weltweit im praktischen Einsatz in den unterschiedlichsten Zusammenhängen bewährt. Insbesondere beim Authoring der XML-Daten bieten kommerzielle Lösungen aus dem Publikationsbereich jedoch deutliche Mehrwerte, beispielsweise mit der Integration von Textbausteinbibliotheken, Ontologien oder auch Terminologie- und Sprachkontrolle (z.B. alle Handlungsanweisungen stehen im aktiv). XML-basierte Produkte aus dem E-Learning Bereich glänzen mit der einfachen Integration von Texten, Bildern, Interaktionen, Simulationen und Übungen.

6 Zusammenfassung

Gerade vor dem geschilderten Hintergrund, dass „re-purpose“ und „re-use“ von Content mittel- und langfristig zentrale Herausforderungen der aktuellen Content Entwicklung im E-Learning Bereich darstellen, scheinen Ansätze, Content nach den Prinzipien des Single-Source-Publishings und mit Blick auf eine mögliche Wiederverwendung der Inhalte zu erstellen, eine Alternative für die bislang betriebenen Vorgehensweisen bei der Content Produktion zu sein. Insbesondere textbasierte Informationen lassen sich mit den in DocBook standardmäßig gebotenen Mechanismen gut verarbeiten und so vorhalten, dass eine Wiederverwendung ermöglicht wird. Eine Herangehensweise mit Hilfe Freier Software ist für die Realisierung des Ansatzes dabei unter Umständen eine ernst zu nehmende Alternative zu kommerziellen Systemen.

Der wesentliche Unterschied des hier geschilderten Ansatzes gegenüber den bisherigen Vorgehensweisen besteht darin, dass auf Basis eines *standardisierten* XML-Formats aus dem Bereich der Softwaredokumentation versucht wird, neben Handbüchern und Online-Hilfen auch Lernsoftware aus derselben Datenquelle zu erstellen. Damit ergeben sich zwangsläufig auch auf der Prozess- und Inhaltsebene Überlappungen und Synergien zwischen Technischer Dokumentation und der ingenieurmäßigen Entwicklung von E-Learning Content, die sich in der Praxis letztlich positiv auf beiden Bereiche auswirken können. Wiederverwendung jedenfalls, spielt in beiden Bereichen eine zentrale Rolle.

Literaturverzeichnis

- [Br01] Brandon-Hall: Authoring Tools Survey 2001
<<http://www.brandonhall.com/public/forms/atsurvey/>> (nicht mehr zugreifbar)
- [DIN04] Deutsches Institut für Normung: PAS 1032-2: Aus- und Weiterbildung unter besonderer Berücksichtigung von e-Learning. Berlin, 2004
- [Do05] Downes, S.: E-learning 2.0. 2005
<<http://elearnmag.org/subpage.cfm?section=articles&article=29-1>> (13.03.06)
- [DH03] Duval, E.; Hodgins, W.: A LOM Research Agenda. 2003, <<http://www.www2003.org/cdrom/papers/alternate/P659/p659-duval.html.html>> (13.03.06)

- [FF95] Frakes, W.B.; Fox, C.J.: Sixteen Questions about Software Reuse, Communications of ACM, Vol 38 (6), 1995, pp 75-87
- [FK05] Frakes, W.B.; K. Kang: Software Reuse Research: Status and Future, IEEE Transactions on Software Engineering, Vol. 31(7), Juli 2005
- [EI05] The eLearningguild: The Content Authoring Research Report 2005. <<http://www.elearningguild.com/pdf/1/Oct05-contentauthor.pdf>> (13.03.06)
- [Ge03] Gersdorf, R.: Eine Content-Management-Architektur für die Umsetzung verteilter Redaktionsprozesse bei der Erstellung wieder verwendbarer Inhalte für das eLearning. In (Uhr, W.; Esswein, W.; Schoop, E., Hrsg.) Wirtschaftsinformatik 2003 / Band I. Physica-Verlag, Heidelberg, S 633-652
- [GPR02] Grützner, I.; Pfahl, D.; Ruhe, G. (2002) Systematic courseware development using an integrated engineering style method. In: Proceedings of the World Congress "NETWORKED LEARNING IN A GLOBAL ENVIRONMENT: Challenges and Solutions for Virtual Education". Technical University of Berlin, Germany
- [GT06] Grützner, I.; Thomas, L.: Systematische Entwicklung von Medien zur Benutzerunterstützung und -schulung mit einem XML basierten Single-Source-Ansatz. Multikonferenz Wirtschaftsinformatik 2006
- [LL97] Lee, N.; Litecky, C. R.: An Empirical Study of Software Reuse with Special Attention to Ada, IEEE Trans. on Software Engineering, Vol. 23(9), September 1997
- [LTV02] Lucke, U.; Tavangarian, D.; Vatterrott, H.: The Use of XML for the Development of an Adaptive Multimedia Teaching and Learning System. World Congress on Networked Learning in a Global Environment (NL2002), Berlin, 2002
- [Ma01] Masie, E.: Tools for Developing e-Learning, 2001. ed at a distance journal, Vol. 5 <http://www.usdla.org/html/journal/MAY01_Issue/article07.html> (13.03.06)
- [MET02] Morisio, M.; Ezran, M.; Tully, C.: Success and Failure Factors in Software Reuse, IEEE Trans. on Software Engineering, Vol. 28(4), April 2002
- [P90] Prieto-Diaz, R.: Implementing Facetted Classification for Reuse, Proc. Int. Conf. on Software Engineering 1990, 1990
- [POS05a] Pankaratius, V.; Oberweis, A.; Stucky, W.: Product Lines in E-Learning. Technical Report 501, University of Karlsruhe, Institute AIFB, 76128 Karlsruhe, Germany. August 2005.
- [POS05b] Pankaratius, V.; Oberweis, A.; Stucky, W.: Lernobjekte im E-Learning - Eine kritische Beurteilung zugrunde liegender Konzepte anhand eines Vergleichs mit komponentenbasierter Software-Entwicklung. 9. Workshop Multimedia in Bildung und Wirtschaft, 2005, Technische Universität Ilmenau
- [RB91] Basili V. R.; Rombach, H. D.: Support for Comprehensive Reuse, Software Engineering Journal, 1991
- [RMW05] Ras, E.; Memmel, M.; Weibelzahl, S.: Integration of E-Learning and Knowledge Management - Barriers, Solutions and Future Issues. In (Althoff, K.-D. et al., Hrsg.): Professional Knowledge Management (WM2005), LNAI 3782, Springer-Verlag, Berlin-Heidelberg.
- [TR05] Thomas, L.; Ras, E.: Courseware Development Using a Single-Source Approach. In: Proceedings of the World Conference on Education Multimedia, Hypermedia and Telecommunications, Ed-Media 2005; S. 4502-4509
- [WM99] Walsh, N.; Muellner, L.: DocBook. O'Reilly, 1999
- [Za03] ZapThink: XML in the Content Lifecycle Foundation Report. 2003 <<http://www.zapthink.com/report.html?id=ZTR-CL100>> (14.03.06)
- [ZK03] Zimmermann, V.; Küchler, T.: Architekturen und Prozesse für die Entwicklung von eLearning-Content. DeLFI 2003; S. 205-214