

Ein Framework für die Publikation von Multimediadokumenten in digitalen Bibliotheken

Sebastian Schick, Holger Meyer, Andreas Heuer
Universität Rostock
{schick,hme,ah}@informatik.uni-rostock.de

30. März 2007

Zusammenfassung

Autoren stehen immer häufiger vor der komplexen Aufgabe, multimediale Lehrmaterialien oder komplex strukturierte multimediale Dokumente zu erstellen und diese in digitalen Bibliotheken zur Verfügung zu stellen. Hier wird oft nur eine Unterstützung bei der Erstellung von Teildokumenten angeboten. Für das Zusammenfassen der Teildokumente in Multimediadokumenten fehlt die Unterstützung durch entsprechende Werkzeuge. Ein weiteres Problem stellt die Kontrolle und Ablaufsteuerung von Publikationsprozessen durch Workflow-Management-Systeme dar, die oft nicht angeboten wird. Starre Abfolgen von Arbeitsschritten geben den Verlauf der Publikation vor. Der hier gezeigte Ansatz stellt einen Publikationsprozess von Multimediadokumenten in digitalen Bibliotheken als Lösung für die oben genannten Probleme vor. Es werden Anforderungen an den Prozess untersucht. Aufbauend auf diesen Anforderungen wird der Publikationsprozess in digitalen Bibliotheken definiert. Als Ergebnis wird ein Java Framework für Publikationsprozesse in digitalen Bibliotheken vorgestellt, das auf den Konzepten von Service-Orientierten Architekturen (SOA) aufbaut. Die SOA ermöglicht es Komponenten einfach hinzuzufügen oder auszutauschen. Um den Publikationsprozess zu unterstützen, wurde das Framework durch die Integration einer Workflow-Komponente erweitert. Das Framework ist prototypisch für das digitale Bibliothekssystem MyCoRe umgesetzt, welches an der Universitätsbibliothek Rostock zum Einsatz kommt.

1 Motivation

Autoren werden bei der Publikation von Multimediadokumenten in digitalen Bibliotheken oft nur bei der Erstellung von Teildokumenten unterstützt. Die Unterstützung für das Zusammenfassen der Teildokumente in Multimediadokumente durch entsprechende Werkzeuge fehlt. Um diese Lücke zu schliessen, wird im folgenden ein Publikationsprozess für Multimediadokumente in digitalen Bibliotheken vorgestellt. Dafür werden in einem ersten Schritt Anforderungen an Multimediale Dokumentenmodelle vorgestellt. Im zweiten Schritt wird ein allgemeiner Erstellungsprozess für Multimediadokumente in digitalen Bibliotheken definiert. Zentrale Punkte sind dabei die Integration existierender Werkzeuge, die [semi-] automatische Ableitung von Metadaten und eine medienspezifische Deskribierung. Ausserdem sind passende Suchmöglichkeiten sowie eine flexible Sicht auf die zu publizierenden Inhalte wichtig. Im folgenden Abschnitt dient der vorgestellte Publikationsprozess dann als Grundlage für ein Java Framework, welches die Wiederverwendbarkeit des Publikationsprozesses erhöhen soll. Im letzten Teil wird dann die Proof-of-Concept Implementierung des Frameworks vorgestellt.

2 Publikationsprozesse in digitalen Bibliotheken

Ein Dokumentenmodell beschreibt nach [EF00] die logische Struktur eines Dokumentes. Für die Modellierung von Multimediadokumenten müssen entsprechende Dokumentenmodelle bestimmte Eigenschaften erfüllen. In [Bol01] werden eine Menge von minimalen Anforderungen vorgestellt, die ein Modell unterstützen muss um in Multimedia Applikationen verwendet werden zu können. Dazu gehören *einfache Anforderungen* wie ein zeitliches und räumliches Modell sowie Interaktionen. Zusätzlich wurden erweiterte Anforderungen wie Wiederverwendbarkeit, Granularität Auswahl und Identifikation, Anpassbarkeit und eine darstellungsneutrale Beschreibung definiert. Diese Eigenschaften werden hier speziell auf die Anforderungen von digitalen Bibliotheken erweitert.

Attributbeziehungen ermöglichen das Zuordnen von Multimedia Content zu einem Element.

Komponentenbeziehungen ermöglichen das Modellieren von zusammengesetzten Objekten.

Mittels **Substitutionsbeziehungen** können Informationen von verschiedenen Medientypen dargestellt werden.

Semantische Äquivalenz beschreibt, dass Objekte den gleichen Inhalt haben, obwohl es sich um verschiedene Varianten des selben Objektes handelt [HS].

Vererbung von Struktur und Verhalten ermöglicht den Aufbau von Hierarchien.

Das Erzeugen von komplex strukturierten Dokumenten in digitalen Bibliotheken, erfolgt auf der Grundlage eines Publikationsprozesses. Da die Autoren, wie oben bereits erwähnt, häufig nur bei der Erstellung von Teildokumenten unterstützt werden, wird im Folgenden ein Publikationsprozess für Multimediadokumente in digitalen Bibliotheken vorgestellt. Ausgehend von der Definition, dass digitale Bibliothekssysteme „alle Prozesse einer digitalen Bibliothek [unterstützen], die auch ein Content-Management-System anbietet“[HD02], wurden die einzelnen Phasen des Publikationsprozesses aus den Phasen der Publikationsprozesse in Content-Management-Systemen (siehe [Boi01]) und den Phasen aus digitalen Bibliothekssystemen kombiniert. Zusammen mit den vorgestellten Anforderungen wurde der in Abbildung 1 dargestellte Publikationsprozess für Multimediadokumente in digitalen Bibliotheken entwickelt. Im Folgenden werden die Phasen vorgestellt.

Das Erstellen von Content erfolgt entweder durch vorhandene Autorenwerkzeuge, die durch eine Erweiterung entsprechende Ausgabeformate erzeugen. An dieser Stelle können Metadaten, die das Autorenwerkzeug selbständig verwaltet, mit an das System übergeben werden. Oder die Daten werden in der nächsten Phase in das System geladen.

Das Erfassen ermöglicht das Importieren von Content über verschiedene Schnittstellen wie zum Beispiel WebDav, FTP oder Web Interfaces.

Das Konvertieren nimmt ein Mapping der Binärdaten und der Struktur auf das jeweils verwendete interne Format vor. Dies stellt eine einheitliche Verarbeitung der Daten in den folgenden Phasen sicher.

Das Aggregieren unterteilt sich in die Phasen Segmentierung, bei der komplex strukturierter Content automatisch, halbautomatisch oder manuell in Komponenten unterteilt wird

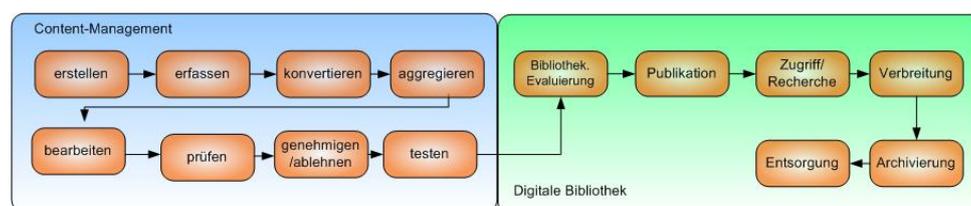


Abbildung 1: Publikationsprozess für Multimediadokumente

und in die Angabe von Metadaten, was eine spätere Auswahl und Identifikation durch entsprechende Suchtechniken ermöglicht.

Das Bearbeiten ist die Phase, in der die Modellierung des Multimediadokumentes auf der Basis des zugrunde liegenden Dokumentenmodells vorgenommen wird. Wichtig ist hier die Umsetzung der gestellten Anforderungen. Zusätzlich muss das neue Multimediadokument hier mit beschreibenden Metadaten erweitert werden. Eine neue Deskribierung ist nicht notwendig, da dies bereits für die Teildokumente durchgeführt wurde.

Das Prüfen beschreibt in Abhängigkeit des Content Typs eine Kontrolle durch den Autor.

Genehmigen / Ablehnen und Testen beschreibt eine Qualitätssicherung durch ein Organ ausserhalb der Bibliothek. An dieser Stelle kann ein Gutachter- und Lektor-Konzept wie in [EF00] vorgestellt, angesetzt werden.

Bibliothekarische Evaluierung ermöglicht nach der Freigabe des Dokumentes, eine Kontrolle und eine bibliothekarische Evaluation.

Publikation gibt das Dokument frei. Da das Dokument für den gesamten Publikationsprozess im Repository gehalten wurde, wird es erst jetzt für die Suchmechanismen zugänglich gemacht.

Die Phasen **Zugriff/Recherche**, **Verbreitung**, **Archivierung** und **Entsorgung** werden in der Regel vom verwendeten digitalen Bibliothekssystem zur Verfügung gestellt und daher hier nicht weiter betrachtet.

3 Ein Framework für den Publikationsprozess

Aufbauend auf dem in Abschnitt 2 vorgestellten Publikationsprozess wurde eine Java Framework entwickelt. Das Framework wird in Abbildung 2 gezeigt. Durch die Verwendung des Frameworks soll eine hohe Wiederverwendbarkeit des erstellten Codes ermöglicht werden. Weiterhin soll der Publikationsprozess schnell an neue Anforderungen angepasst werden können. Grundsätzlich wird davon ausgegangen, dass das Framework keine Ausgabefunktionalität und keinen eigenen Daten-Store für die Verwaltung von Datenobjekten bietet. Beides soll durch vorhandene Anwendungen zur Verfügung gestellt werden.

3.1 Architektur

Das Framework baut auf den Konzepten des in [Wu04] vorgestellten X18p Frameworks auf. Für die Instanziierung wird das Framework Spring verwendet. X18p bietet alle Funktionalitäten in Form von Services an. Das Konzept wird unter den folgenden Annahmen gemacht:

- Eine Service-Vermittler-Schicht vermittelt zwischen der Anwendung und der Geschäftslogik
- Anfragen an die Geschäftslogik werden Grundsätzlich an einen Service-Router gestellt. Dieser leitet die Anfrage an ein entsprechenden Service-Provider-Controller weiter. Die dafür benötigten Informationen befinden sich in der bean.xml Datei.
- Ein Service-Provider-Controller kann Services direkt einbinden

Anlehnend an das X18p Framework wird das hier vorgestellte Framework in einen Service-Requester, Service-Router, Service-Controller, und Service-Module unterteilt. Abbildung 2 zeigt das entwickelte Framework für den Publikationsprozess von Multimediadokumenten in digitalen

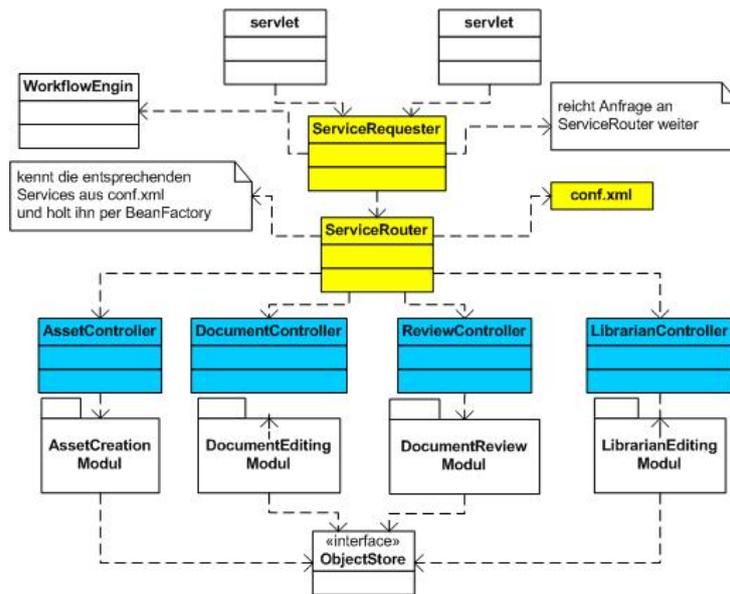


Abbildung 2: Framework für den Publikationsprozess in digitalen Bibliotheken

Bibliotheken.

Der **Service-Requester** ist als zentraler Teil des Frameworks für die Verarbeitung von Anfragen verantwortlich. Entsprechend [Wu04] wird ein *ServiceRequest-Objekt* zurückgeliefert, welches zum Transport von Daten verwendet wird. Diese Daten setzen sich aus dem angeforderten Service, den auszuführenden Operationen des Services, und Daten die der Service verarbeiten soll zusammen.

Der **Service-Router** wertet das *ServiceRequest-Objekt* aus und kann aus den enthaltenden Informationen und den Informationen aus der *bean.xml* Datei einen entsprechenden *Service-Controller* auswählen. Anders als in [Wu04] vorgegeben, wird hier der Spring Service-Controller direkt aufgerufen.

Ein **Service-Controller** steuert die Verarbeitung des *ServiceRequest-Objektes*. Dabei kontrolliert es die Abarbeitung durch die ihm zur Verfügung stehenden Service-Module.

Die **Service-Module** fassen Komponenten zusammen, die die eigentliche Funktionalität in Form von Services bereitstellen. Die Komponenten werden durch einen *Service-Locator* ausgewählt.

3.2 Service-Module

Die in Abbildung 2 gezeigten Service-Module entsprechen der Einteilung aus Abschnitt 2. Wie bereits erwähnt, werden einzelne Komponenten in Service-Module zusammengefasst. Diese Komponenten bieten jeweils nur einen speziellen Service an (siehe [Wu04]).

Der **AssetController** und das Modul *AssetCreationModul* fassen die in Abschnitt 2 vorgestellten Komponenten zum Datei-Upload, Datei-Konvertierung und zur Aggregation zusammen.

Der **DocumentController** und das Modul *DocumentEditingModul* fassen die Komponenten für das Erstellen von Multimediadokumenten zusammen. An dieser Stelle ist zu erwähnen, dass nur Dokumente für die Modellierung verwendet werden können, die bereits im Repository vorhanden sind.

Der **DocumentController** muss weiterhin das Erstellen verschiedener Beziehungstypen unterstützen.

Der **ReviewController** und das *DocumentReviewModul* fassen die Komponenten für das Kontrollieren von erzeugten Dokumenten zusammen.

Der **LibrarianController** und das *LibrarianEditingModul* bieten die Funktionalität für das bibliothekarische Evaluieren von erzeugten Dokumenten an.

Das **ObjektStore** Modul bietet eine Schnittstelle zu einem Repository, in dem Datenobjekte gespeichert werden. Es wird hier kein eigener Controller bereitgestellt, da Datenobjekte nur von Komponenten und Modulen direkt genutzt werden können.

Um die Modularität und damit auch eine leichte Austauschbarkeit zu unterstützen, müssen für die einzelnen Service-Module Und Komponenten Java Interfaces und abstrakte Klassen implementiert werden.

Um den Publikationsprozess von Multimediadokumenten in einer digitalen Bibliothek mit Hilfe eines Workflow- Management-Systems (WFMS) zu steuern, wird eine Schnittstelle eingeführt. Da die konkreten Anforderungen von Anwendungen schwer zu verallgemeinern sind, gestaltet sich die Anbindung eines WFMS schwierig. Der hier gewählte Ansatz geht davon aus, dass über ein Interface, welches für ein konkretes WFMS implementiert werden muss, die Service-Controller und der Service-Requester mit dem WFMS interagieren. So kann sichergestellt werden, dass der Publikationsprozess immer entsprechend eines vorgegebenen Workflows abgearbeitet wird. Zusätzlich können Aufgaben an Rollen verteilt werden. Ausserdem besteht die Möglichkeit bestimmte Komponenten direkt vom WFMS ausführen zu lassen. Dies steht dann nicht mehr unter der Kontrolle des Frameworks.

Das oben vorgestellte Framework wurde in einer Proof-of-Concept Implementierung an der Universitätsbibliothek Rostock mit dem digitalen Bibliothekssystem MyCoRe (www.mycore.de) umgesetzt. Es wurde ein Multimedia Dokumentenmodell für Bildserien erstellt. Das Modell beschreibt einen Container für Bilder. Eine Bildserie kann Metadaten und n Bilder enthalten. Diese Bilder müssen vom Datentyp Bild sein, der ebenfalls definiert wurde. Als Ausgangspunkt für eine neue Bildserie dient die Bildverwaltungssoftware Google Picasa. Sie bietet die Möglichkeit eine Auswahl von Bildern in einer Ordnerstruktur mit beschreibender XML Datei zu exportieren. Der Publikationsprozess unterstützt den Autor beim Import dieser Daten in die digitale Bibliothek.

4 Ausblick

Da das verwendete Datenmodell nicht vollständig den hier vorgestellten Anforderungen genügt, ist als nächster Schritt die Umsetzung des in [IB05] vorgestellten Modells vorgesehen. Es unterstützt die Anforderungen an ein Multimedia Dokumentenmodell und ist durch seine UML Repräsentation einfach zu erweitern. Eine stärkere Integration eines Workflow-Management-Systems ist ebenfalls angedacht, da die prozessorientierten Ansätze hier oft zu Problemen führen. Dies kann zum Beispiel die Änderung des Workflows aufgrund von Änderungen der Dokumentstruktur sein. Ein dokumentenzentrierter Ansatz ist zu untersuchen.

Literatur

- [Boi01] Bob Boiko. *Content Management Bible*. John Wiley & Sons, Inc., New York, NY, USA, 2001.
- [Bol01] Susanne Boll. *ZYX – Towards flexible multimedia document models for reuse and adaptation*. PhD thesis, Vienna University of Technology, 2001.
- [EF00] Albert Endres and Dieter W. Fellner. *Digitale Bibliotheken*. dpunkt-Verl., 1. aufl. edition, 2000.
- [HD02] Andreas Heuer and Klaus R. Dittrich. Schwerpunktthema: Content Management und digitale Bibliotheken. *Datenbank-Spektrum*, 2(4):7–8, 2002.
- [HS] Thomas Heimrich and Günther Specht. Struktur-, verhaltens- und ausgabeschemata für multimedia-daten in objektrelationalen datenbanksystemen. <http://www.old.netobjectdays.org/pdf/03/papers/ws-mik/314.pdf>.
- [IB05] Temenushka Ignatova and Ilvio Bruder. Utilizing a multimedia uml framework for an image database application. In *ER (Workshops)*, pages 23–32, 2005.
- [Wu04] Fangjian Wu. Design a simple service-oriented j2ee application framework. <http://www.javaworld.com/javaworld/jw-10-2004/jw-1004-soa.html>, 10 2004.