

# **Entwicklung eines Management Systems für DocBook-Dokumente auf Basis von X- Technologien**

## **DIPLOMARBEIT**

zur Erlangung des akademischen Grades  
Diplom-Wirtschaftsinformatiker (FH)  
am Fachbereich Wirtschaft der Fachhochschule Nordostniedersachsen

Eingereicht von:  
Marko Petersen  
7. Fachsemester Wirtschaftsinformatik  
Matrikelnummer: 151012

Erstprüfer:  
Prof. Dr. Hinrich E. G. Bonin  
Zweitprüfer:  
Prof. Dr. Brigitte Urban

Eingereicht am: 30.01.2003

---

# Vorwort

Die vorliegende Diplomarbeit entstand im Rahmen meines Praxissemesters an der

Fachhochschule Nordostniedersachsen  
Volgershall 1  
21339 Lüneburg

in dem Forschungs- und Entwicklungsprojekt "*Digitales Publizieren im Selbstverlag*".

Die Diplomarbeit enthält somit den Praxisbericht.

Ich möchte mich auf diesem Wege bei allen bedanken, die mich während der Entstehungsphase dieser Arbeit sowie während meiner gesamten Studienzzeit begleitet und unterstützt haben. Insbesondere möchte ich mich bei den Betreuern dieser Arbeit, Herrn Prof. Dr. Hinrich E.G. Bonin und Frau Prof. Dr. Brigitte Urban, für Rat, Anregungen und die freundliche Unterstützung bei der Erstellung dieser Arbeit bedanken.

Besonderer Dank gebührt meinen Eltern, die mir sowohl durch finanzielle als auch moralische Unterstützung dieses Studium ermöglicht haben.

Tespe, den 30.01.2003

---

Marko Petersen

---

---

# Abstract

Die vorliegende Diplomarbeit beschreibt die Konzeption und Entwicklung des *DigiPub Document-Management-System* (DigiPub-DMS). Hierbei handelt es sich um ein webbasiertes System zur Verwaltung von Dokumenten, welches im DigiPub-System der Fachhochschule Nordostniedersachsen (FH NON) als Kernsystem eingesetzt wird. *DigiPub* steht in diesem Zusammenhang für *Digitales Publizieren*, das DigiPub-System ist das Ergebnis des Forschungsprojektes *Digitales Publizieren im Selbstverlag (DPS)* an der FH NON.

Das DigiPub-DMS ermöglicht das Publizieren von Dokumenten über das Internet. Dabei gilt ein besonderes Augenmerk den Dokumenten, die im XML-Format DocBook erstellt worden sind. Diese Dokumente werden in einer XML-Datenbank gespeichert, wodurch eine Volltextsuche oder auch eine Anfrage nach Teilen der Dokumente möglich ist, um aus diesen ein neues Dokument zu erstellen. Des Weiteren bietet das System die Möglichkeit, Anmerkungen zu Dokumenten direkt in diese einzufügen (*Annotation*). Das Präsentationslayout von DocBook-Dokumenten wird vom System vorgegeben und ein entsprechendes PDF-Dokument für die Auslieferung generiert, sodass ein Autor sich auf inhaltliche Aspekte konzentrieren kann.

Das DigiPub-System ermöglicht zudem die Veröffentlichung von Dokumenten mit einer *International Standard Book Number (ISBN)*. Im Rahmen des DPS-Projektes wurde neben dem DigiPub-DMS ein Zahlungs- und Versandsystem sowie ein System zur Sicherung von Autorenrechten entwickelt. Diese Systeme zusammen bilden das DigiPub-System, welches online erreichbar ist unter <http://digipub.fhnon.de> (Zugriff: 20.01.2003).

---

---

# Inhaltsverzeichnis

1. Einleitung .....	1
1.1. Aufbau dieser Arbeit .....	3
1.2. Konventionen .....	3
1.2.1. Konventionen zu diesem Dokument .....	3
1.2.2. Konventionen zur Programmierung .....	5
1.3. Erweiterungen des DigiPub-DMS .....	6
2. Systembeschreibung .....	7
2.1. Funktionale Leistungen .....	7
3. Umfeld .....	9
3.1. Einordnung des Systems .....	9
3.2. Systeme an der FH NON .....	10
3.3. Books on Demand .....	13
3.4. Slicing Books Technology .....	14
3.5. Zope Book .....	16
4. Eingesetzte Technologien .....	18
4.1. X-Technologien .....	18
4.1.1. XML .....	18
4.1.2. XSL .....	23
4.1.3. Umfang von XML .....	27
4.2. DocBook .....	28
4.3. Datenspeicherung .....	30
4.3.1. XML Datenspeicherung .....	30
4.3.2. Relationale Datenspeicherung .....	34
4.4. Jakarta Struts .....	35
5. Grundlegende Entscheidungen .....	40
5.1. MVC-Design .....	40
5.2. Präsentationsschicht .....	40
5.3. Internationalisierung .....	41
5.4. Zugriff auf XML-Daten .....	42
5.5. Sicherheitskonzept .....	44
5.6. Fehlerbehandlung .....	46
5.7. Datenbankdesign .....	47
5.7.1. Relationale Datenbank .....	47
5.7.2. XML-Datenbank .....	48
5.8. Konfiguration .....	50
6. Konzeption und Implementation .....	51
6.1. Mandantenmanagement .....	51
6.2. Benutzermanagement .....	53
6.3. Dokumentenmanagement .....	55
6.3.1. Einstellen von Dokumenten .....	55

---

6.3.2.Dokumentensuche.....	65
6.3.3.VirtuelleDokumente.....	68
6.3.4.Dokumentenabruf.....	70
6.3.5.Dokumentenverwaltung.....	71
6.4.Annotationen.....	71
7.ZusammenfassungundAusblick.....	77
A.Programmbausteine.....	80
A.1.JSP-Dateien.....	80
A.2.Java-Klassen.....	82
A.3.XSL-Stylesheets.....	87
B.SoftwareundLizenzen.....	88
B.1.EingebundeneFremdsoftware.....	88
B.2.Produktionsumgebung.....	89
Literaturverzeichnis.....	90
Glossar.....	93
Stichwortverzeichnis.....	101

---

# Abbildungsverzeichnis

2.1. Systemdesign .....	7
3.1. Inhalt, Struktur und Darstellung .....	9
3.2. Auswahl Slicing Book .....	15
3.3. Ergebnis Slicing Book .....	15
3.4. Zope Book .....	16
4.1. Navigationsachsen in XPath .....	25
4.2. Zusammenspiel von XSLT und XPath .....	26
4.3. Spezifikationen rund um XML .....	28
4.4. JSP Model 2 Architektur .....	36
4.5. Struts Architektur .....	38
5.1. DigiPub Präsentationsschablone .....	40
5.2. Vergleich von DOM und SAX .....	43
5.3. Relationale Datenbank "digipub" .....	48
5.4. XML-Datenbank "digipub" .....	49
5.5. Tamino Schema Editor .....	49
5.6. FoStyle.dtd .....	50
5.7. Mime Type.dtd .....	50
6.1. Konfiguration der Nutzergruppen .....	52
6.2. Dokumente einstellen - Übersicht .....	55
6.3. DocBook-Dokumente einstellen - Formular .....	57
6.4. Binäre Dokumente einstellen - Formular .....	64
6.5. Dokumentensuche - Formular .....	65
6.6. Suchergebnis .....	68
6.7. Annotation vornehmen .....	72
6.8. Beispielannotation in PDF .....	75
A.1. Klassendiagramm: de.fhnon.digipub.model .....	85

---

## Tabellenverzeichnis

3.1. Vergleich zwischen Systemen an der FH NON .....	12
5.1. Properties-Dateien zur Internationalisierung .....	42
5.2. Nutzerrollen .....	45
6.1. Nutzergruppen eines Mandanten .....	52
A.1. JSP-Dateien im DigiPub-DMS .....	80
A.2. Action-Klassen im DigiPub-DMS .....	82
A.3. Form-Klassen im DigiPub-DMS .....	83

---

## Beispiele

3.1. StructuredText im Zope Book .....	17
4.1. dokument.xml .....	19
4.2. dokument.dtd .....	20
4.3. dokument.xsd .....	23
4.4. Textbausteine_en_US.properties .....	39
4.5. Internationalisierung mit Struts .....	39
5.1. Einbindung lokalisierter HTML-Dateien .....	42
5.2. DigiPubException .....	46
6.1. Zugriffsliste für Nutzer .....	53
6.2. Metainformationen .....	60
6.3. PDF-Erzeugung .....	62
6.4. XPath-Query .....	66
6.5. Suchergebnis in XML .....	67
6.6. DocBook Element: chapter .....	73
6.7. Valides DocBook-Dokument .....	74
6.8. Annotation in Docbook .....	76

---

# Kapitel 1. Einleitung

Die jährliche Produktion an Informationen auf der Welt würde 1.5 Milliarden Gigabyte Speicherplatz benötigen. Dies entspricht ungefähr 250 Megabyte pro Person für jeden Mann, jede Frau und jedes Kind auf der Erde.

— Ergebnis einer Untersuchung der „University of California“<sup>1</sup>

Die Informationsflut im Internet wächst täglich, und elektronische Dokumente erhalten dabei eine immer wichtigere Rolle, seien es schriftliche Arbeiten oder auch optische Medien wie Bilder und Videoaufnahmen. Von den insgesamt verfügbaren Informationen liegen lediglich rund 0,003 Prozent in gedruckter Form vor. Der Anteil privater Informationen ist mehr als 2.600-mal so groß wie der Teil an veröffentlichten Informationen<sup>1</sup>. Ursache hierfür könnte zum einen sein, dass dieser Informationsanteil nicht veröffentlicht werden soll, zum anderen, dass den Urhebern Systeme zur einfachen Veröffentlichung elektronischer Dokumente nicht zur Verfügung stehen oder bekannt sind.

Die vorliegende Diplomarbeit ist im Rahmen eines Paxissemesters in dem Forschungsprojekt „Digitales Publizieren im Selbstverlag“ (DPS) der Fachhochschule Nordostniedersachsen (FH NON) entstanden. Dieses Projekt, unter der Leitung von Prof. Dr. Hinrich E.G. Bonin, wurde mit dem Ziel gegründet, ein XML<sup>2</sup>-basiertes Dokumenten-Management-Portal zu konzipieren und einzuführen, welches die Publikation von elektronischen Dokumenten sowohl in digitaler als auch in gedruckter Form ermöglicht. Dieses System wurde für den Einsatz als zentrales Publikationssystem an der FH NON entwickelt. Dort bietet es sowohl Studenten als auch Professoren sowie Organen der Fachhochschule eine einfache Möglichkeit zur Veröffentlichung von Dokumenten.

Ausgangsbasis für das DPS-Projekt sind die Ergebnisse des AWÖ-Projektes<sup>3</sup> der FH NON aus dem Sommersemester 2002, in welchem das LIVE-System<sup>4</sup> entwickelt wurde. Dieses System ermöglicht das Publizieren elektronischer Dokumente und ist speziell ausgelegt auf XML-Dokumente, die nach einer selbst entwickelten *Document Type Definition* (DTD) verfasst worden sind. Zur Realisierung des DigiPub-DMS wurden die mit dem LIVE-System gesammelten Erfahrungen eingesetzt, das System an sich jedoch in Anbetracht erheblicher Neuerungen von Grund auf neu durchdacht und implementiert.

<sup>1</sup> Siehe <http://www.sims.berkeley.edu/research/projects/how-much-info/summary.html>, (Zugriff: 19.01.2003)

<sup>2</sup> Extensible Markup Language

<sup>3</sup> DV-Anwendungen in der öffentlichen Verwaltung

<sup>4</sup> Lehrinhalte Interaktiv veröffentlichen

---

Das Ergebnis des DPS-Projektes ist das DigiPub-System. Dieses setzt sich zusammen aus dem DigiPub-DMS (als Kurzform für „DigiPub Document-Management-System“), welches in diesem Dokument beschrieben wird, sowie aus einem Zahlungssystem, einem System zur Sicherung von Autorenrechten und einem Versandsystem. Letztere wurden von Stephan Wiesner im Rahmen seiner Diplomarbeit entwickelt. Ein weiteres Ergebnis des DPS-Projektes ist *DocBookWriter*, ein Autorenwerkzeug zum Erstellen von DocBook-Dokumenten. Dieses Programm wurde von Oliver Kitzing im Rahmen seiner Diplomarbeit erstellt.

Das DigiPub-DMS stellt das Basissystem des DPS-Projektes dar. Es umfasst u.a. ein Webinterface, das Benutzermanagement sowie die Möglichkeit zur Speicherung und zum Abruf von Dokumenten. Hierbei stehen aktuelle X-Technologien, vor allem das XML-Datenformat DocBook, im Mittelpunkt. Für Dokumente in diesem Datenformat bietet das System zusätzliche Features, wie das Erstellen neuer Dokumente aus bestehenden Dokumententeilen (im Folgenden *virtuelle Dokumente* genannt) oder das Einfügen von Anmerkungen (*Annotationen*) direkt im Dokument. Durch den Einsatz von XML wird eine Trennung von Inhalt (XML-Dokument) und Layout (XSL<sup>5</sup>-Stylesheet) eines Dokumentes sowie dessen Durchsuchbarkeit gewährleistet, da es sich bei XML-Dokumenten um strukturierte Daten in einem reinen Textformat handelt (vgl. Abschnitt 4.1.1).

Ein Autor kann sich beim Schreiben von DocBook-Dokumenten auf den Inhalt konzentrieren. Durch die Verwendung von XSL-Stylesheets (vgl. Abschnitt 4.1.2) kann eine automatisierte Generierung des Präsentationsformates im Nachhinein zugewiesen werden. Ein Autor ist dabei nicht auf ein proprietäres Dateiformat beschränkt, das DocBook-System bietet bereits vorgefertigte Lösungen zur Erstellung von Dokumenten in der *HyperText Markup Language* (HTML) oder dem XSL-Formatting-Objects Format. Aus letzterem Format lassen sich anschließend beispielsweise einfache Textdateien, Postscript oder auch, wie im DigiPub-DMS, PDF-Dateien erstellen (siehe „XSL (Formatting Objects)“).

Diese Diplomarbeit wurde konsequenterweise im DocBook-Format erstellt, um selbst von den DocBook-spezifischen Funktionen des DigiPub-DMS Gebrauch zu machen.

Als Gesamtsystem ermöglicht das DigiPub-System die Veröffentlichung und Abrufung elektronischer Dokumente jeder Art, wobei DocBook-Dokumente sowohl in gedruckter als auch in digitaler Form abrufbar sind. Diese können mit einer ISBN versehen und bei Bedarf erzeugt werden, wie es die Philosophie von „Print on Demand“ ist, worauf detaillierter in Abschnitt 3.3 eingegangen wird. Zudem ist ein Zahlungs- und Drucksystem implementiert, um den Anforderungen eines internetbasierten Verlagssystems gerecht zu werden.

---

<sup>5</sup> Extensible Stylesheet Language

---

## 1.1. Aufbau dieser Arbeit

Im folgenden Teil der Einleitung werden zunächst die verwendeten Konventionen beschrieben. Anschließend wird auf weitere Teile des DigiPub-Systems eingegangen, die eng mit dem DigiPub-DMS verbunden sind.

Kapitel 2. *Systembeschreibung* gibt einen Überblick über die Architektur des DigiPub-Systems sowie die einzelnen Funktionalitäten des DigiPub-DMS.

In Kapitel 3. *Umfeld* wird ein Vergleich des DigiPub-Systems mit bereits bestehenden Systemen an der FH NON vorgenommen. Es werden weiterhin die Technologien *Books on Demand*, *Slicing Books Technology* und *Zope Book* vorgestellt und deren Kernfunktionalitäten mit denen des DigiPub-DMS verglichen.

Kapitel 4. *Eingesetzte Technologien* beschreibt die zur Realisierung des DigiPub-DMS eingesetzten Technologien und geht besonders auf den Bereich der X-Technologien ein.

Kapitel 5. *Grundlegende Entscheidungen* erläutert grundlegende und komponentenübergreifende Entscheidungen zur Implementation des DigiPub-DMS.

Kapitel 6. *Konzeption und Implementation* geht detailliert auf den systeminternen Ablauf der einzelnen Funktionalitäten ein und erklärt deren Implementation im DigiPub-DMS.

Kapitel 7. *Zusammenfassung und Ausblick* schließt die Arbeit ab, fasst die erreichten Ergebnisse zusammen und geht auf nahe liegende nächste Entwicklungsschritte ein.

## 1.2. Konventionen

Dieser Abschnitt beschreibt die in diesem Dokument eingehaltenen Konventionen zur Darstellung der unterschiedlichen Textpassagen. Außerdem werden die Konventionen zur Programmierung des DigiPub-DMS erläutert.

### 1.2.1. Konventionen zu diesem Dokument

Ein wesentlicher Grund für die Entscheidung für ein XML-Format wie DocBook ist die Trennung von Inhalt und Layout. Durch Zuweisung verschiedener Präsentationsschablonen (XSL-Stylesheets, vgl. Abschnitt 4.1.2) kann der Inhalt auf unterschiedliche Arten dargestellt werden. Dementsprechend ist auch die Darstellung von einzelnen Komponenten wie Quellcode oder einem Verweis abhängig von diesem Layoutformat. Eine übliche Beschreibung von typographischen Konventionen (wie beispielsweise „Programmcode ist durch die Schrift Courier dargestellt“) ist für diese Dokumente daher während der Erstellung nicht festzulegen.

---

Da dieses Dokument in DocBook erstellt ist, wurden zur Kennzeichnung von hervorzuhebenden Textpassagen entsprechende DocBook-Elemente genutzt. Um die einzelnen typographischen Komponenten zu beschreiben, werden im Folgenden Beispiele für die in diesem Dokument verwendeten DocBook-Elemente dargestellt. Auf diese Weise bleibt die Unabhängigkeit der Präsentationsform erhalten.

- Markup-Elemente werden durch das Element `<markup>` als solche gekennzeichnet und in diesem Dokument wie folgt dargestellt: `element1`, `element2`. Diese Notationsform wird ebenfalls bei Attributen von Elementen (vgl. Abschnitt 4.1.1) oder Komponenten einer DTD verwendet.
- Zur Darstellung weiterer verschiedener Bezeichner werden ebenfalls entsprechende Elemente verwendet:
  - `<classname>` zur Darstellung von Klassennamen (Beispiel: `Klasse1`, `Klasse2`)
  - `<methodname>` zur Darstellung von Methodennamen (Beispiel: `methodeEins()`, `methodeZwei()`)
  - `<filename>` zur Darstellung von Dateinamen (Beispiel: `Datei1.txt`, `Datei2.txt`)
- Quelltext wird durch das Element `<programlisting>` gekennzeichnet. An einigen Stellen werden einzelne Abschnitte des Quelltextes referenziert, diese sind dementsprechend gekennzeichnet. Ein Beispiel für eine Quelltextangabe (inklusive Referenzierung):

```
<p>Dies ist eine Quellcodeangabe</p>
```

❶

- ❶ An dieser Stelle wird der markierte Teil des Quelltextes referenziert.
- Zitate werden durch das Element `<blockquote>` gekennzeichnet, ein Beispiel hierfür:

Dies ist ein Zitat, welches aus der angegebenen Quellenangabe übernommen wurde. Ausgelassene Passagen sind durch die Zeichen [...] gekennzeichnet.  
—Dies ist die Quellenangabe des Zitates
- Literaturangaben werden aus dem Literaturverzeichnis über das Element `<xref>` referenziert, ein Beispiel für einen Verweis auf eine Literaturangabe: [Walsh1999].
- Begriffe, die auf einen Eintrag im Glossar oder einen anderen Dokumentenabschnitt verweisen, werden ebenfalls über das Element `<xref>` referenziert. Ein Beispiel für eine Referenz auf einen Glossareintrag: XML. Zwei Beispiele für eine Referenz auf einen anderen Dokumentenabschnitt: Abschnitt 1.2, Kapitel 2. *Systembeschreibung*.

- 
- Verweise auf externe Webquellen werden durch das Element `<ulink>` gekennzeichnet und erhalten auf Grund der Veränderbarkeit der Quelle ein Zugriffsdatum. Ein Beispiel: <http://digipub.fhnon.de> (Zugriff: 30.12.2002).
  - Wichtige Begriffe werden, soweit sie keinen Verweis auf das Glossar oder einen anderen Dokumentenabschnitt bilden, durch das Element `<emphasis>` gekennzeichnet und in diesem Dokument wie folgt dargestellt: *wichtiger Begriff*.
  - Weitere Begriffe, die im Kontext eines Satzes eine besondere Bedeutung haben und zum Verständnis beitragen sollen, sind durch das Element `<quote>` gekennzeichnet und in diesem Dokument wie folgt dargestellt: „Begriff“.

## 1.2.2. Konventionen zur Programmierung

Konventionen zur Programmierung wurden im DPS-Projekt von den beteiligten Teilnehmern gemeinsam erstellt.

Als zentrales Datenformat für Dokumente wurde XML gewählt, und zwar speziell das Dokumentenformat DocBook. Dieses ist ein etabliertes Format zur Darstellung strukturierter Dokumente und wird separat in Abschnitt 4.2 behandelt.

Als Programmiersprache wurde *Java* gewählt. Hierdurch wird eine Plattformunabhängigkeit sowie eine optimale Unterstützung für internetbasierte Anwendungen gewährleistet. Des Weiteren stehen diverse frei verfügbare Tools in Java zur Verfügung, die vor allem in Bezug auf XML-Verarbeitung eine gute Unterstützung bieten. Die Programmierung erfolgt in Englisch. Dies betrifft sowohl die Namensgebung für einzelne Bezeichner als auch die Kommentare im Quelltext. Als Konventionen zur Programmierung in Java wurden die von Sun Microsystems vorgeschlagenen Empfehlungen eingehalten<sup>6</sup> mit der Ausnahme, dass öffnende geschweifte Klammern nicht in derselben Zeile wie die vorangegangene Anweisung stehen, sondern in einer eigenen Zeile direkt unter der Anweisung. Diese Schreibweise wird von dem DPS-Team als übersichtlicher empfunden.

Kommentare im Java Quelltext sind im JavaDoc-Format erstellt und entsprechen den JavaDoc-Konventionen von Sun Microsystems<sup>7</sup>. Ein Kommentar direkt über einer Klasse, einem Interface, einer Methode, einer Variablendeklaration oder einer statischen Initialisierungsanweisung mit *static* bezieht sich genau auf diese Komponente. Kommentare innerhalb von Methoden beziehen sich genau auf den unmittelbar folgenden Block.

<sup>6</sup> Die empfohlenen Code-Konventionen von Sun Microsystems in Bezug auf Java sind online abrufbar unter <http://java.sun.com/docs/codeconv/> (Zugriff: 13.01.2003).

<sup>7</sup> Die empfohlenen Code-Konventionen von Sun Microsystems in Bezug auf JavaDoc sind online abrufbar unter <http://java.sun.com/j2se/javadoc/writingdoccomments/index.html> (Zugriff: 13.01.2003).

---

JSP-Dateien enthalten direkt nach ihren Direktiven<sup>8</sup> einen JSP-Kommentar, der die Verwendung der JSP-Seite erläutert. Sofern Erläuterungen zu speziellen Teilen innerhalb der JSP-Seiten für erforderlich angesehen wurden, sind dort Kommentare enthalten, die ihre Reichweite selbst beschreiben.

Alle Element- und Attributbezeichnungen in XML- und XSL-Dateien werden klein geschrieben. XSL-Dateien enthalten direkt nach ihrer XML-Deklaration einen Kommentar, der die Funktion der Datei erläutert. Zudem befindet sich direkt über jedem Template eine Erläuterung zu diesem in Form eines XML-Kommentares. Auch hier sind, falls Erläuterungen zu speziellen Teilen innerhalb der Datei für erforderlich gehalten wurden, Kommentare eingefügt, die ihre Reichweite selbst beschreiben.

### **1.3. Erweiterungen des DigiPub-DMS**

Im Rahmen des DPS-Projektes wurde entschieden, das DigiPub-System so zu konzipieren, dass u.a. eine Veröffentlichung von ISBN-Dokumenten ermöglicht wird. Das DigiPub-DMS stellt hierzu das Basissystem dar und bietet entsprechende Schnittstellen an, kann jedoch alleine diese Anforderungen nicht erfüllen.

Zur Publikation von Dokumenten mit ISBN wird sowohl ein Zahlungssystem benötigt als auch die Möglichkeit, ein Dokument in gedruckter Form abzurufen. Dieser Teil des Systems wurde von Stephan Wiesner konzipiert und beinhaltet neben den oben genannten Aspekten weitere Funktionalitäten wie zum Beispiel das Einfügen von Sicherheitseinstellungen in auszuliefernde PDF-Dokumente oder die Erstellung von Zugriffsstatistiken für abgerufene Dokumente.

Um die Schnittstellen für diese Erweiterungen bereitzustellen, sammelt das DigiPub-DMS bereits alle nötigen Informationen über Autoren und Dokumente wie etwa Kontoinformationen oder gewählte Sicherheitseinstellungen. Die endgültige Auslieferung der Dokumente erfolgt in der Implementation des DigiPub-Systems an der FH NON über das Sicherheits- und Zahlungssystem. Zum genauen Verständnis dieser Teile wird daher auf die Diplomarbeit von Stephan Wiesner verwiesen.

<sup>8</sup> Diese Direktiven sind durch das einleitende `<%@` zu erkennen.

# Kapitel 2. Systembeschreibung

Bevor in Kapitel 3. *Umfeld* ein Vergleich des DigiPub-Systems mit weiteren Systemen zur Veröffentlichung von Dokumenten erfolgt, bietet dieses Kapitel einen Überblick über das DigiPub-System und stellt anschließend die funktionalen Leistungen des DigiPub-DMS heraus.

In Abbildung 2.1. *Systemdesign* ist das Design des DigiPub-Systems dargestellt. Einzelne Funktionalitäten, wie die Suchmöglichkeit oder das Einfügen von Annotationen, wurden aus Gründen der Übersichtlichkeit herausgenommen. Diese Abbildung veranschaulicht den Kernablauf - das Einstellen und Abrufen von Dokumenten - anhand des an der FH NON implementierten Systems.

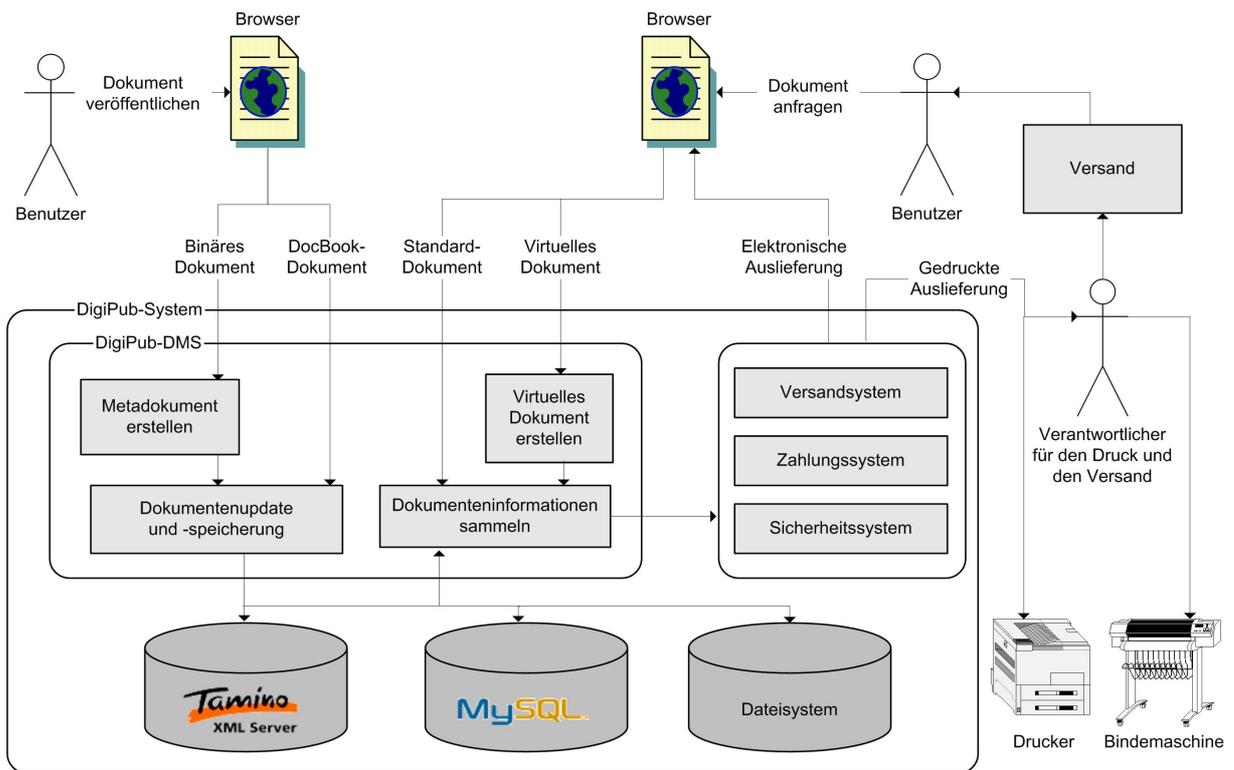


Abbildung 2.1. Systemdesign

Die Verwaltung der Dokumente wird im DigiPub-System vom DigiPub-DMS übernommen. Die Auslieferung erfolgt durch das Sicherheits-, Zahlungs- und Versandsystem. Für den Druck, die Bindung und den Versand von Dokumenten muss einer Person die Zuständigkeit übertragen werden; diese Vorgänge sind nicht automatisiert.

## 2.1. Funktionale Leistungen

Im Folgenden werden die besonderen Leistungen des DigiPub-DMS beschrieben. Einzelne Leistungen werden über eine alphanumerische Bezeichnung in der Form  $Lmn$  mit  $mn = 01, \dots, 99$  identifiziert.

- 
- L01 DigiPub-DMS ermöglicht das Veröffentlichen von Dokumenten unter einzelnen Mandanten. Diese können
    - L01.1 interne Benutzerrechte auf Basis von Emailadressen setzen
    - L01.2 mögliche interne Kategorien zur Dokumentenveröffentlichung festlegen
  - L02 DigiPub-DMS erlaubt das Veröffentlichen elektronischer Dokumente jeglicher Art.
    - L02.1 DocBook-Dokumente werden in einer XML-Datenbank gespeichert und automatisch durch ein systemeigenes XSL-Stylesheet für die Auslieferung zu einem PDF-Dokument transformiert.
    - L02.2 Für Nicht-DocBook-Dokumente (Binärdokumente) wird ein XML-Datensatz mit Metadaten erstellt.
  - L03 DigiPub-DMS ermöglicht eine mandantenübergreifende Suche nach Dokumenten.
    - L03.1 Für Binärdokumente ist eine Suchmöglichkeit anhand ihrer Metadaten gegeben.
    - L03.2 Für DocBook-Dokumente ist eine Suchmöglichkeit sowohl anhand ihrer Metadaten als auch über den kompletten Inhalt (Volltext) gegeben.
    - L03.3 Suchergebnisse für DocBook-Dokumente können auf einzelne gefundene Kapitel oder Abschnitte begrenzt werden.
  - L04 DigiPub-DMS ermöglicht das Erstellen neuer Dokumente (virtuelle Dokumente) aus Teilen bestehender DocBook-Dokumente.
  - L05 DigiPub-DMS ermöglicht das Abrufen von Dokumenten sowohl in elektronischer als auch in gedruckter Form<sup>9</sup>.
  - L06 DigiPub-DMS ermöglicht das Annotieren von DocBook-Dokumenten direkt im Dokument.
  - L07 DigiPub-DMS ist weltweit und internationalisiert über das Internet verfügbar.
  - L08 DigiPub-DMS ermöglicht die Veröffentlichung von ISBN-Dokumenten.

<sup>9</sup> Das DigiPub-DMS bietet hierzu eine Schnittstelle, da diese Funktionalität ausgegliedert wurde. Weitere Details sind in Abschnitt 1.3 beschrieben.

---

## Kapitel 3. Umfeld

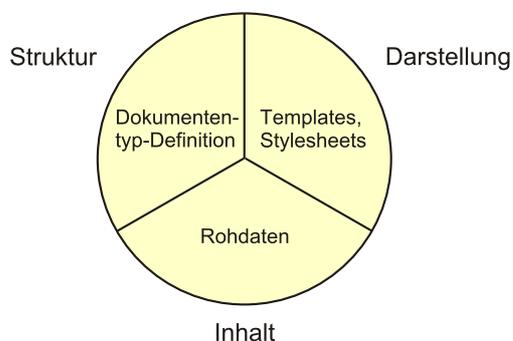
In diesem Abschnitt wird zunächst eine Einordnung des DigiPub-DMS in Bezug auf Content-Management- und Document-Management-Systeme vorgenommen. Im Anschluss daran wird das DigiPub-System mit bereits bestehenden Systemen zur Veröffentlichung von Dokumenten an der FH NON verglichen.

Des Weiteren wird das Konzept von *Books on Demand*, einem internetbasierten Verlagssystem, sowie die *Slicing Book Technology*, ein System zur Unterteilung von existierenden Dokumenten in einzelne semantische Einheiten, und das *Zope Book* vorgestellt. Bei dem Konzept von Books on Demand erfolgt die Erstellung eines gedruckten Dokumentes erst bei Anfrage, also „on demand“. Die Dokumente liegen hierzu in digitaler Form vor, genau wie es auch im DigiPub-System der Fall ist. Mit Hilfe der Slicing Book Technology lassen sich neue Dokumente aus Dokumententeilen erstellen. Diese Technik wird mit dem Konzept für virtuelle Dokumente des DigiPub-DMS verglichen. Zope Book erlaubt das Erstellen von Annotationen direkt innerhalb eines Dokumentes; auch diese Technik wird mit der des DigiPub-DMS verglichen.

### 3.1. Einordnung des Systems

Das DigiPub-DMS ist ein System zur Veröffentlichung von Dokumenten und besitzt Eigenschaften eines Document- bzw. Content-Management-Systems, wie datenbankgestützte Speicherung der Dokumente oder auch rollenbasierten Zugriff, weshalb es auch den Namen *DigiPub Document-Management-System* trägt.

Ein Content-Management-System verwaltet *Content*, wobei dieser Begriff als Summe der einzelnen wesentlichen Informationen zur Darstellung von Inhalt bezeichnet werden kann (veranschaulicht in Abbildung 3.1. *Inhalt, Struktur und Darstellung*). Diese Art der Verwaltung trifft für das DigiPub-DMS in Bezug auf DocBook-Dokumente zu.



**Abbildung 3.1. Inhalt, Struktur und Darstellung**

Quelle für Abbildung 3.1. *Inhalt, Struktur und Darstellung*: Eigener Entwurf in Anlehnung an [Schuster2000].

---

Document-Management-Systeme sind nach [Schuster2000] eine Ausprägung der Content-Management-Systeme, sie verwalten *Dokumente*. Unter einem Dokument wird hierbei ein Objekt bezeichnet, welches Informationen zu einem bestimmten Bereich trägt und in sich klar abgeschlossen ist.

Das DigiPub-DMS kann jedoch nicht eindeutig in diesen Bereich eingeordnet werden. Content-Management-Systeme, und somit auch Document-Management-Systeme, müssen als Funktion die Erstellung neuer Informationen (Dokumente) und deren Änderung enthalten<sup>10</sup>. Dies ist im DigiPub-DMS nicht gegeben. Das DigiPub-DMS stellt ein Archivierungssystem für Dokumente dar und ist speziell ausgelegt auf DocBook-Dokumente, für die es zusätzliche Kernfunktionalitäten bietet (vgl. Abschnitt 2.1).

Für die Erstellung von DocBook-Dokumenten existiert bereits eine große Anzahl an Autoren-systemen. Die Integration eines solchen Systems in das DigiPub-DMS wurde nicht in Betracht gezogen. Viele XML-Editoren bieten ebenfalls bereits eine spezielle Unterstützung zum Erstellen von DocBook-Dokumenten. Ein im Rahmen des DPS-Projektes erstelltes System ist der *DocBookWriter*, ein Plug-In für den XML-Editor *JEdit*.

Als Einsatzbereich des DigiPub-DMS ist das Publizieren von Dokumenten vorgesehen und nicht ein Unterstützungsprozess beim Erstellen dieser Dokumente. Aufgrund dessen wird es im Rahmen des DigiPub-Systems als „Publikationssystem“ bezeichnet.

## 3.2. Systeme an der FH NON

An der FH NON bestehen zum jetzigen Zeitpunkt (31.12.2002) mehrere Möglichkeiten, elektronische Dokumente im Internet zu veröffentlichen. Eine oft genutzte Variante ist die Veröffentlichung auf einer Webseite oder in einem FTP-Verzeichnis über einen entsprechenden Server. Diese Vorgehensweise bietet lediglich eine Archivierungs- und einfache Abrufmöglichkeit, keine weiteren Funktionalitäten wie beispielsweise eine Suchoption. Neben diesen einfachen Möglichkeiten existieren an der FH NON allerdings noch weitere Systeme, die zum Publizieren von Dokumenten genutzt werden können.

### 3.2.1. The City of Contropolis

*The City of Contropolis*, erreichbar unter <http://content.fhnon.de> (Zugriff: 31.12.2002), bietet die Möglichkeit zur Bildung virtueller Communities (Gemeinschaften), welche das Rückgrat von Contropolis bilden. Hierdurch wird eine Kommunikationsplattform angeboten, in der neben der Veröffentlichung von elektronischen Dokumenten weitere Funktionalitäten wie Nachrichtenaustausch zwischen einzelnen Benutzern oder Diskussionsforen angeboten werden. Primäres Ziel von Contropolis ist nicht das Publizieren von Dokumenten, sondern vielmehr das Angebot einer Plattform mit verschiedenen Services zur Unterstützung des Kommunikati-

<sup>10</sup> Vgl. [Schuster2000]

---

onsprozesses, basierend auf verschiedenen Communities und deren Mitgliedern. Contropolis ermöglicht das Veröffentlichen von Beiträgen, denen elektronische Dokumente beigelegt werden können.

### 3.2.2. ILIAS

*ILIAS* (Integriertes Lern-, Informations- und Arbeitskooperationssystem), erreichbar unter <http://rzserv2.fhnon.de/ilias/> (Zugriff: 31.12.2002), stellt eine Lernplattform zur Verfügung, die im Rahmen des VIRTUS-Projekts<sup>11</sup> an der Wirtschafts- und Sozialwissenschaftlichen Fakultät der Universität zu Köln entwickelt wurde. Primäres Ziel dieses Systems ist auch hier nicht die Publikation von einzelnen Dokumenten, sondern das Erstellen von virtuellen Kursen und webbasierten Lerneinheiten, um Studenten ein interaktives Lernen in einer einheitlichen Umgebung zu ermöglichen. Auf der Basis von Kursen können elektronische Dokumente veröffentlicht werden.

### 3.2.3. OPUS

Auf dem Publikations- und Multimediaserver der Universitätsbibliothek Lüneburg, erreichbar unter <http://kirke.ub.uni-lueneburg.de/> (Zugriff: 13.01.2003), werden Hochschulschriften im PDF-Format publiziert. Basis hierfür ist das System *OPUS*, welches ursprünglich an der Universität Stuttgart entwickelt wurde<sup>12</sup>. Hierbei handelt es sich um ein System speziell zum Veröffentlichen von Dokumenten.

### 3.2.4. Vergleich der Systeme

In Tabelle 3.1. *Vergleich zwischen Systemen an der FH NON* werden die vorgestellten Systeme der FH NON in Hinblick auf Funktionalitäten für das Publizieren elektronischer Dokumente miteinander verglichen. Die dargestellten Informationen sind den Hilfeseiten beziehungsweise den Dokumentationen entnommen, soweit diese über das jeweilige System abrufbar waren. Bei Funktionalitäten, die nicht in der jeweiligen Dokumentation beschrieben sind, wurden die Angaben anhand eigener Versuche spezifiziert. Bei nicht erfolgreichen Versuchen wurde angenommen, dass die Funktionalität nicht gegeben ist. Für Angaben über das *ILIAS*-System wurde zudem Stephan Wiesner befragt, der u.a. an der Einführung dieses Systems an der FH NON beteiligt war.

<sup>11</sup> Homepage des VIRTUS-Projektes: <http://www.virtus.uni-koeln.de/virtus/index.html> (Zugriff: 13.01.2003)

<sup>12</sup> Das OPUS-System der Universität Stuttgart ist erreichbar unter <http://elib.uni-stuttgart.de/opus/> (Zugriff: 13.01.2003)

**Tabelle 3.1. Vergleich zwischen Systemen an der FH NON**

<b>Funktionali- tät</b>	<b>DigiPub Sy- stem</b>	<b>Contropolis</b>	<b>ILIAS</b>	<b>OPUS</b>	<b>FTP/WWW</b>
Akzeptierte Dateiformate	Alle	Alle	PDF, HTML und Bilddateien	PDF	Alle
Abrufbare Form	Elektronisch oder gedruckt <sup>a</sup>	Elektronisch	Elektronisch	Elektronisch	Elektronisch
Stichwortsuche	Ja	Nein	Ja	Ja	Nein
Volltextsuche	Begrenzt auf Dokumente in DocBook	Nein	Begrenzt auf Textdateien	Nein <sup>b</sup>	Nein
Interaktion von Lesern	Durch Annotationen direkt im Dokument	Durch Nachrichtenaustausch oder Diskussionen im System	Durch Nachrichtenaustausch oder Diskussionen im System	Nein	Nein
Erstellen neuer Dokumente aus Dokumententeilen	Begrenzt auf Dokumente in DocBook	Nein	Begrenzt auf Textdateien	Nein	Nein
Dynamische Layoutzuweisung	Begrenzt auf Dokumente in DocBook	Nein	Begrenzt auf Textdateien	Nein	Nein
Internationalisierte Benutzeroberfläche	Ja (English und Deutsch implementiert)	Nein	Ja (English und Deutsch implementiert)	Nein	Nein

<sup>a</sup> Für das Konzept zur Auslieferung von Dokumenten in gedruckter Form siehe Abschnitt 1.3.

<sup>b</sup> Laut Dokumentation ist eine Volltextsuche im OPUS-System möglich, Tests blieben jedoch erfolglos. Antwort auf eine anschließende Anfrage bei der Universitätsbibliothek Lüneburg: „Eine Volltextsuche ist in unserem System leider nicht möglich“ (Dr. Christian Haenger, 09.01.2003 per Email).