

HUMBOLDT-UNIVERSITÄT ZU BERLIN
WIRTSCHAFTSWISSENSCHAFTLICHE FAKULTÄT
INSTITUT FÜR STATISTIK UND ÖKONOMETRIE
LEHRSTUHL FÜR STATISTIK

Diplomarbeit

zur Erlangung des Grades eines Diplom-Kaufmanns

MD*Book Online - eine universelle Schnittstelle zur Generierung netzbasierter interaktiver Bücher

Rodrigo Witzel

(Matrikel-Nr. 117816)

Prüfer: Prof. Dr. Wolfgang Härdle



12. April 2001

Inhaltsverzeichnis

Inhaltsverzeichnis	2
1 Einführung	5
1.1 Ziel der Arbeit	6
1.2 Gliederung, Layout, Begriffe	6
2 Ausgangspunkt	7
2.1 Was ist ein eBook ?	7
2.2 eBook Standards	8
2.3 Open eBook Forum	8
2.3.1 Open eBook Publication Structure	9
2.3.2 A Framework for the Epublishing Ecology	10
2.4 Electronic Book Exchange Working Group	11
2.4.1 Überblick über das EBX System	12
2.4.2 Anforderungen an das EBX System	13
2.4.3 Weitere Standardisierungsbemühungen	14
2.5 Kommerzielle Lösungen	15
2.6 Adobe Content Server und Acrobat eBook Reader	15
2.7 MicroSoft Digital Asset Server	16
2.8 Gemstar	18
2.9 Andere Ansätze	18
3 MD*Book im Vergleich	21
3.1 Anforderungen	21
3.2 MD*Book's Eigenschaften	25
3.3 Fazit	29
4 Architektur	31
4.1 Allgemeines Use Case Schema	31

4.2	Accounts und eBooks verwalten	31
4.3	eBooks erzeugen	33
4.4	Ablauf	33
5	Ausführung	37
5.1	Die Java Server Pages	37
5.2	Das Package services	38
5.2.1	Die Klassen FileManager und FTPConfigWriter	39
5.2.2	Die Klasse AuthService	40
5.2.3	Die Klasse DBAccess	40
5.3	Das Package server	40
5.4	Das Package client	41
5.4.1	Die Klassen MBOComponentBuilder und MBOComponent	41
5.4.2	Die Klasse Console	42
5.5	Das Package protocol	43
5.5.1	Die Klasse SignalSerializer	43
5.5.2	Die Klassen SignalType und Signal	43
6	Ausblick	47
6.1	Unterstützung Adobe's eBook Formats	47
6.2	Unterstützung MicroSoft's eBook Formats	48
6.3	Weitere Aufgaben	48
7	Glossar	51
	Bibliography	53
8	MD*Book Online	57
9	Kopien zitierter Quellen	59

Kapitel 1

Einführung

Die sprunghafte Entwicklung des Internets ermöglicht weltweiten Zugriff auf Informationen, die vorher nur in lokalen Fachzeitschriften oder gar nicht zugänglich waren. Autoren haben heute einen hohen Anreiz ihre Texte auch im Internet zu publizieren. Zum einen kann ein nahezu unmittelbarer Kontakt zwischen dem Autor und Leser geschaffen und der Bekanntheitsgrad des Autors und der Einfluß seiner Publikationen deutlich gesteigert werden. Zum anderen ermöglichen die im Internet vereinten Technologien eine neue Qualität der Publikationen. Elektronische Bücher, Publikationen im Internet allgemein, können mittels Querverweisen (Hyperlinks) direkt verbunden werden und einzelne Abschnitte können detaillierte Verweise zueinander beinhalten, so daß die Struktur eines Buches einem Netz von miteinander verknüpften Informationsknoten ähnelt. Eine weitere neue Qualität sind die interaktiven Komponenten in Büchern. Bücher bestehen nicht mehr aus reinem Text, sondern enthalten zusätzlich Algorithmen, die beschreiben auf welche Art und Weise auf Aktionen des Lesers reagiert werden soll. Gerade diese Technologie bewirkt eine Zäsur in der Entwicklung der Lehrbücher. Moderne Lehrbücher benutzen als grundlegendes Konzept modulare Lerneinheiten, die mit interaktiven Elementen (learning by doing) angereichert sind, um die Motivation der Lernenden und die Lernergebnisse zu steigern. Lehrbücher werden also zukünftig Beschreibungen von Interaktionen enthalten. Bisher entwickelte Software und laufende Projekte der Hersteller, besonders von Textverarbeitungssoftware (MicroSoft und Adobe (MSa 2000, ?)) und eBook-Geräten (GS 2000), betonen lediglich die Verfügbarkeit von elektronischen Büchern auf Handhelds und Palmtops und deren kommerzielle Verwertung, sie vernachlässigen aber die interaktive Komponente. Daher entwickeln heute Autoren eigene Lösungsmöglichkeiten für ihr jeweiliges Fachgebiet, um ihre Dokumente mit interaktiven Komponenten anzureichern. Am Lehrstuhl für Statistik ¹ wird seit drei Jahren an einem Konzept gearbeitet, um interaktive Komponenten in HTML Seiten einzubinden. Das Konzept beruht auf der Kommunikation zwischen Java Applets (*XploRe Client*) und der Statistiksoftware XploRe (*XploRe Server*), die in diesem Kontext als Dienstleister wirksam wird. Der *XploRe Client* erzeugt mittels der Server-Daten Dialogfelder und gibt grafische und textuelle Daten aus. Es steht nun

¹an der Wirtschaftswissenschaftlichen Fakultät der Humboldt Universität zu Berlin

das Werkzeug MD*Book zur Verfügung, mit dem man aus L^AT_EX-Dokumenten HTML Seiten mit diesen interaktiven Elementen erzeugen kann. Nachdem die grundlegenden technischen Probleme bei der Integration der interaktiven Komponenten in den statischen Text gelöst wurden, soll nun in einem zweiten Schritt eine moderne Benutzerschnittstelle für das Werkzeug entwickelt werden und geprüft werden, inwieweit MD*Book Teil einer allgemeinen eBook Entwicklungsumgebung sein kann.

1.1 Ziel der Arbeit

Diese Arbeit soll Lehrenden, vor allem der Fachbereiche Statistik und Mathematik, ein Werkzeug zur Verfügung stellen, mit dem sie auf einfache Art und Weise netzbasierte, interaktive Bücher erstellen können und auf diese Weise zukünftigen Lehrmittelstandards gerecht werden können. Dazu soll das bestehende MD*Book Werkzeug erweitert werden, so daß es den Bedürfnissen der Autoren wissenschaftlicher Publikationen als auch Entwicklern von Lehrmitteln gerecht wird. Der Schwerpunkt soll dabei auf der Entwicklung einer anwenderfreundlichen, internetweit einsetzbaren Schnittstelle liegen, die die bestehende MD*Book Funktionalität kapselt. Die Schnittstelle ist umfassend und WWW basiert zu dokumentieren.

1.2 Gliederung, Layout, Begriffe

Zu Beginn werden bestehende eBook-Standards und -Lösungen untersucht, darauf aufbauend die Stärken und Schwächen des MD*Book Werkzeugs analysiert und verschiedene Lösungsansätze verglichen, um seine Schwächen zu beheben. Im dritten Teil wird die erweiterte Architektur des MD*Book entwickelt, die die gefundenden Defizite ausgleicht und die Integration und Kompatibilität zu bestehenden eBook-Konzepten gewährleistet. Anschließend wird die konkrete Implementation der Architektur dokumentiert. Eine Plan für zukünftige Erweiterungen des MD*Book schließt die Arbeit ab.

Quellcode-Fragmente und Bezeichner setze ich in der Schriftart `typewriter`. Als Zitierformat für elektronische Dokumente verwende ich den *International Standard ISO 690-2* [(ISO690 2000)]. Absätze in englischer Sprache sind immer Zitate. Die Bedeutung oft verwendeter Begriffe erläutere ich, soweit das nicht im Text geschieht, im Glossar 51 (ohne im Text explizit auf das Glossar zu verweisen).

Kapitel 2

Ausgangspunkt

2.1 Was ist ein eBook ?

Anbieter von eBook Lösungen definieren eBooks oft mit der Menge der Leistungsmerkmale, die ihre Produkte umsetzen ¹⁵. Die beiden Standardisierungsorganisationen, die sich mit eBooks beschäftigen, haben jeweils einen Vorschlag zur Definition gemacht.

Die *EBX Working Group* betont den Transfer von statischem Text auf Papier zu seiner elektronischen Repräsentation:

„E-book: A digital object that is an electronic representation of a book. While an e-book can consist of a single page, it is normally thought of as an electronic analog of a multipage hardcover or paperback book. A e-book is the so-called 'content' in the EBX model. An e-book may come in a variety of formats, including, but not limited to, PDF, Open eBook Publication Structure 1.0, and various other textual and multimedial formats.

Die *OEBF's Digital Rights Management Strategy working group* betont jedoch den eigenständigen Charakter der eBooks:

- „Electronic Book (n.) *Sometimes ebook*. 1. A Literary Work in the form of a Digital Object, consisting of one or more standard unique identifiers, Metadata, and a Monographic body of Content, intended to be published and accessed electronically. *See also epublication*. 2. May also refer to the hardware device created for the purpose of reading ebooks (vide RocketBook, SoftBook, Franklin e-bookman).”
- Electronic Journal. (n.) *Sometimes ejournal*. A Literary Work in the form of a Digital Object, consisting of one or more standard unique identifiers, Metadata, and an ongoing Serial body of Content, intended to be published and accessed electronically. *See also epublication*.

- Electronic Publication. (n.) *Sometimes epublication.* A Literary Work disseminated in the form of a Digital Object and accessed electronically. These works can be multimedia, incorporating text, digitized images, animation, sound tracks, and functioning formulas. They may also use hypertext technology. (Modified CHI93.) Electronic Publishing. (n.) *Sometimes Epublishing.* The act of disseminating Literary Works in digital form. Ejournal. *See electronic journal.*

Ein eBook kann mit Grafiken, Animationen, Musik und funktionsfähigen Formeln angereichert werden und ist dann eine *electronic publication*, im Gegensatz zu einem *electronic journal*, das nur aus sequentiellen Text besteht. Ebooks sind in dieser Definition mehr als elektronische Repräsentationen von Büchern. Grundsätzlich kann man mit dem eBook-Format sogar digitalisierte Filme (animation) und Spiele (functioning formulas) abbilden, aber die Definition betont den Anteil des Textes im Buch (a literary work).

Bei der Analyse des eBook-Begriffes kommerzieller Anbieter werden wir sehen, daß die Definition des OEBF die wichtigere ist.

2.2 eBook Standards

Gegenwärtig entwickeln zwei Organisationen standardisierte eBook Formate und Standards für ihre Verbreitung. Beiden Organisationen gehören eine Vielzahl privater Unternehmen aber auch staatliche Institutionen an.

2.3 Open eBook Forum

Die erste Organisation ist das Open eBook Forum, eine internationale, gemeinnützige Organisation. Sie ist im Oktober 1998 ins Leben gerufen worden:

„The Open eBook Forum is committed to international participation and creating Publishing standards recognized worldwide.”¹

Ihr gehören u. a. Adobe Systems Incorporated (inklusive Glassbook Incorporated²), Versaware Technologies, Incorporated, Microsoft Corporation, Palm Computing, Gemstar eBook Group Limited (inklusive NuvoMedia und SoftBook Press³), Amazon.com, LLC und Bertelsmann AG an; weltweit insgesamt mehr als 130 Mitglieder und Partner.

Sie hat bisher die Spezifikation *Open eBook Publication Structure 1.0* (OEB 1999) und den Entwurf *A Framework for the Epublishing Ecology* (OEB 2000) veröffentlicht.

¹Open eBook Forum Homepage and Electronic Book '99 Conference Program

²Adobe hat Glassbook im August 2000 übernommen. Glassbook entwickelt PDF basierte eBook Reader

³Gemstar ist mit NuvoMedia und SoftBook Press fusioniert. NuvoMedia produziert das Rocket eBook, ein portables eBook Lesegerät, ähnlich den Handhelds. Softbook Press stellt ein ähnliches Gerät her. Beide zusammen dominieren den Markt für derartige Geräte.

2.3.1 Open eBook Publication Structure

Die Ziele der Spezifikation sind insbesondere:

- „The specification is intended to give content providers (e.g., publishers, and others who have content to be displayed) and tool providers minimal and common guidelines which ensure fidelity, accuracy, accessibility, and presentation of electronic content over various electronic book platforms.
- The specification seeks to reflect established content format standards.
- The goal of this specification is to provide the purveyors of electronic-book content (publishers, agents, authors et al.) a format for use in providing content to multiple reading systems.”

Die Spezifikation vereinbart zwei Dokumentformate, die beide durch XML 1.0 DTDs ([W3C 1998](#)) definiert sind.

OEB Packages

Ein Dokument des ersteren Typs (*OEB Packages*) beinhaltet

- die Dublin Core Metadaten der *OEB Publication* (*dc-metadata*^[4]) und
- Informationen über die Gesamtheit aller Dokumente, die zu einer OEB Publication (typischerweise ein Buch) gehören (*x-metadata*).

Dublin Core Metadaten sind unter anderem der Titel, der Autor, der Verleger, eine Beschreibung des Inhalts, die Namen der Mitwirkenden inklusive ihrer Rollen (Illustrator, Übersetzer, Autor des Nachworts usw.), die Sprache und das Erscheinungsdatum.

Die einzelnen Dokumente der Publikation werden mit einem eindeutigen Bezeichner versehen und durch ihre URI ^[5] und ihren MIME Typ ^[6] eindeutig beschrieben. Ein Fallback-Attribut verweist auf eine alternative Repräsentation des jeweiligen Dokuments. Diese Information soll von einem eBook Reader ausgewertet werden, wenn das Gerät den angegebenen MIME Typ nicht darstellen kann.

Mit weiteren Elementen wird angegeben,

- in welcher Reihenfolge die Dokumente am besten zu lesen sind (*spine*),

⁴Dublin Core Metadata Initiative. Dublin Core Metadata Element Set, Version 1.1: Reference Description [online] 02 Juli 1999 [zitiert am 24 März 2001]. Im World Wide Web verfügbar: <http://dublincore.org/documents/dces/>

⁵RFC 2396 Uniform Resource Identifiers (URI): Generic Syntax, T. Berners-Lee, MIT/LCS, R. Fielding, U.C. Irvine, L. Masinter, Xerox Corporation, <ftp://ftp.isi.edu/in-notes/rfc2396.txt>, August 1998

⁶N. Freed, Innosoft, N. Borenstein, RFC 2046, MIME Part Two: Media Types, <ftp://ftp.isi.edu/in-notes/rfc2046.txt>, November 1996

- welche Dokumente in kurzen Einführungen gezeigt werden sollen (*tours*) und
- welche Dokumente strukturelle Bedeutung für die Publikation haben (*guide*), dazu gehören z. B. das Inhaltsverzeichnis, das Abbildungsverzeichnis und das Vorwort.

OEB Document

Dokumente des zweiten Typs (*OEB Document*) können in OEB Packages referenziert werden, sind also Teil einer OEB Publication. Alle OEB Document Markups sind auch Elemente der HTML 4.0 Spezifikation. Im Unterschied zu HTML 4.0 müssen die Markups ohne schließendes End-Tag in der XML Syntax notiert werden (`` statt ``). OEB Documents können einen Teil der in CSS 1 Spezifikation definierten CSS Konstrukte verwenden. Zum Einbinden HTML-fremder Inhalte wird ausschließlich das Object-Tag benutzt, Grafiken ausgenommen. Das betrifft z. B. Java-Applets. Da der Java-Applet MIME Typ nicht in der *core OEB MIME media types list* (Objekte dieser MIME Typen müssen von jedem Reading Device interpretiert werden können) enthalten ist, muß zu jedem Java-Applet ein Fallback Attribut angegeben werden, das festlegt, welches andere Objekt angezeigt werden soll, wenn der eBook Reader Java-Applets nicht interpretieren kann.

Das Format der OEB Documents ist also ein reduziertes HTML Format. Es wurden nur die Tags übernommen, die auf allen gegenwärtig denkbaren Reading Devices interpretiert werden können. Jedoch ist wie im HTML Format die Angabe zusätzlicher Markups erlaubt, deren Semantik die Reading Devices jedoch nicht kennen und interpretieren müssen.

Die Open eBook Forum hat mit dieser Spezifikation ein Format entwickelt, das auf bestehenden Standards aufsetzt und für das bereits auf nahezu alle Plattformen frei verfügbare Parser und Interpreter existieren.

Es ist zu prüfen, inwieweit die von MD*Book erzeugten Bücher in OEB Publikationen überführt werden können.

2.3.2 A Framework for the Epublishing Ecology

Das Rahmenwerk besteht aus drei Teilen, die die Schaffung einer einheitlichen Begriffswelt in der Epublishing Welt forcieren sollen.

Der erste Teil ist ein Glossar, das auch die oben angeführte eBook-Definition enthält.

Der zweite Teil modelliert die Komponenten der Epublishing Welt (Referenzmodell). Es werden drei grundlegende Komponententypen identifiziert:

domains „The epubli­shing world is composed of multiple domains (e.g., commercial, non-profit, government, academic ,etc.).”

elements „Four types of elements populate a domain: Objects, Roles, Interactions and Authorities. Roughly speaking, Objects flow between players of Roles in Interactions that are governed by Authorities”

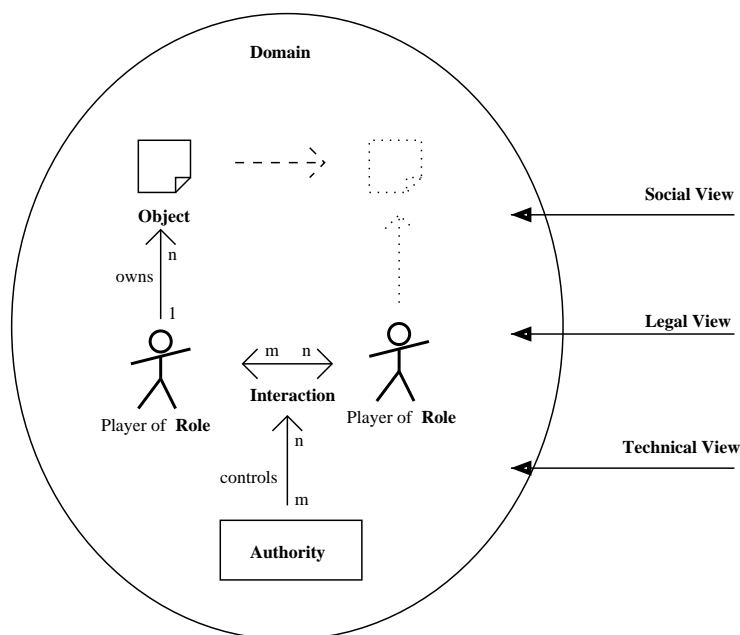


Abbildung 2.1: Elemente und ihre Beziehungen im Referenzmodell

perspektives „The domains and the elements within them can be viewed from three fundamental perspectives (social, legal and technical), each of which reveals a different view of the various elements”

Im Referenzmodell sollen am Epublishing beteiligten Parteien inklusive dem Endbenutzer und ihre Beziehungen zueinander modelliert werden können, wobei die Authorities ein Umfeld für Interaktionen darstellen, innerhalb dessen bestimmte Regeln eingehalten werden müssen.

Im dritten Teil werden alle Parteien im Epublishing analysiert und kategorisiert. Auf diese Weise ergänzt und verfeinert er das Glossar.

In dieser Arbeit verwendete Begriffe besitzen im Zweifel die Semantik, die ihnen im Rahmenwerk zugeordnet wird.

2.4 Electronic Book Exchange Working Group

Die zweite Organisation zur Standardisierung der eBook-Formate und ihrer Verbreitung ist die *Electronic Book Exchange Working Group*:

„The Electronic Book Exchange (EBX) Working Group develops open, freely available, and commercially viable standards for the secure transmission of electronic books (e-books) among rights holders, intermediaries, and users. EBX addresses such issues as the purchase, sale, lending, giving, printing, subscribing, and licensing of electronic books.”

Mitglieder der EBX Working Group und beteiligte Unternehmen sind u. a. Adobe Systems Incorporated, Versaware, Intel, Microsoft, Palm Computing, Gemstar eBook Group Limited usw.; insgesamt sind mehr als 30 Unternehmen und Organisationen beteiligt⁷.

Die bisher einzige Veröffentlichung der EBX Working Group ist das texttElectronic Book Exchange System(EBX 2000) in der Version 0.8 (Entwurf) im Juli 2000. In ihr werden der komplette Ablauf und alle Schnittstellen beschrieben, die benötigt werden, um eBooks zu vertreiben.

2.4.1 Überblick über das EBX System

Die drei Schlüsselideen des EBX System sind

- die Installation zertifizierter Software bei allen beteiligten Parteien,
- die Verwendung von Belegen (textbfvouchers, die für jede einzelne Buchinstanz generiert werden und immer mit dem Buchinhalt zusammen übertragen werden,
- die Verwendung von Public-Key-Verschlüsselungsverfahren bei allen Übertragungen zwischen den Parteien.

Jeder Verleger, Buchverkäufer und Kunde muß eine seiner Rolle entsprechende Software installiert haben, um am Epublishing-Prozeß teilzunehmen und die Software des Kunden muß auf den verschiedensten Geräten verfügbar sein.

Damit der Beleg als Schutz vor unkontrollierter Weitergabe benutzt werden kann, werden in ihm folgende Daten gespeichert (EBX 2000, 14-15):

- ein eindeutiger Bezeichner (ISBN) des Buches,
- der Schlüssel, mit dem der Inhalt des Buches dechiffriert werden kann,
- die Anzahl der Kopien, die der Halter des Buches verkaufen, verleihen, verschenken darf,
- spezielle Rechte auf den Kopien des Buches, d. h. ob man sie verleihen oder nur verschenken darf, wie lange man sie verleihen darf, wieviele Kopien und in welchem Umfang man Kopien für den persönlichen Gebrauch anlegen darf

Die Parteien kommunizieren mit XML-Dokumenten über das HTTP-Protokoll und speziellen HTTP Headern. Da zu einem eBook mehrere Dateien gehören können (z. B. OEB Publication) und die konsistente sichere Übertragung mehrerer Dateien die Komplexität des Protokolls erhöht, werden diese Dateien zu einer einzigen Datei verbunden (z. B. ZIP).

Das EBX System ist also ein komplexes System von genau spezifizierten Protokollen, Dokumenttypen und Schnittstellen, dessen Umsetzung erheblichen

⁷Das Open eBook Forum und die EBX Working Group planen zukünftig ihre Anstrengungen zu kombinieren, denn zum einen sind ein Großteil der Mitglieder in beiden Organisationen vertreten und zum anderen gibt es Überschneidungen bei den gesetzten Zielen.

Entwicklungsaufwand und den Willen zur Standardisierung von Hard- und Softwareherstellern erfordert. Es ist zu untersuchen, inwieweit MD*Book mit bestehenden Implementationen dieser Protokolle und Schnittstellen zusammenarbeiten kann und die MD*Book-Funktionalität erweitert werden muß.

2.4.2 Anforderungen an das EBX System

Im Hinblick auf die Bewertung des MD*Book ist das in dem Entwurf formulierte Anforderungsprofil, um den eBook Vertrieb zu organisieren und die Rechte der beteiligten Parteien zu sichern, besonders interessant (EBX 2000, 14-15), denn es gibt Anknüpfungspunkte zum ebook-Format (mit * versehen):

Consumer Needs	
<ul style="list-style-type: none"> • Interoperability* • Easy of Use • Transparent Performance • Giving and Lending (First Sale) • Copyright Law Help (Fair Use or Personal Use) 	<ul style="list-style-type: none"> • Backup • Backward Compatibility* • Privacy • Added Value* • Authenticity of Content • Preview • Refund
Publisher Needs	
<ul style="list-style-type: none"> • Copyright and Revenue Protection • Ease of Production • Elimination of Physical Manufacturing • Scalability 	<ul style="list-style-type: none"> • Accountability and Auditability • Bookseller Distrust • Marketing Data • Versioning • Super Distribution • Selling
Bookseller and Distributor Needs	
<ul style="list-style-type: none"> • Copyright Protection • Scalability 	<ul style="list-style-type: none"> • Liability Security • Publisher Distrust

Author Needs	
<ul style="list-style-type: none"> • Copyright and Revenue Protection 	<ul style="list-style-type: none"> • Self Publishing • Interactive Content*
Library Needs	
<ul style="list-style-type: none"> • Lending and Borrowing • Fair Use 	<ul style="list-style-type: none"> • Archiving • Plattform Independence*
Annahmen	
<ul style="list-style-type: none"> • EBook Leser wünschen die überwiegende offline-Nutzung der Bücher.* • Es ist keine Möglichkeit vorgesehen, die Transformation elektronischer Bücher in physische Repräsentationen zu unterstützen.* • Das Internet und das World Wide Web ist die bevorzugte Infrastruktur für alle beteiligten Parteien, um eBooks weiterzugeben. 	

2.4.3 Weitere Standardisierungsbemühungen

Es existieren weitere Ansätze, um Rechte auf elektronischen Medien zu schützen:

- Digital Property Rights Language (DPRL, XML Dialekt)
- Open Digital Rights Language (ODRL, XML Dialekt)
- Extensible Rights Markup Language (XrML, XML Dialekt)
- Payment API for v1.0 Internet Open Trading Protocol (IOTP), IETF Internet Draft (inklusive Electronic Commerce Modeling Language (ECML))

Den beiden erst genannten Ansätzen fehlt jedoch die breite Unterstützung der eBook-Industrie und sie werden daher hier nicht untersucht.

XrML wird von *ContentGuard, Incorporated* entwickelt und zahlreichen anderen Unternehmen unterstützt; darunter sind wieder Adobe und MicroSoft. Obwohl die XrML Spezifikation erst als Entwurf vorliegt, wird es bereits in der Microsoft eBook-Lösung verwendet [16](#).

Das Konzept der IETF umfaßt bereits zwei RFCs und einige Entwürfe. Es beruht ebenso wie das EBX System auf elektronischen Belegen (voucher), Public-Key-Verfahren, dem HTTP Protokoll und XML Protokolldaten. Es ist jedoch nicht eBook spezifisch, sondern allgemein auf den Schutz elektronischer Inhalte angepaßt. Es wird daher nicht zwangsläufig den Anforderungen des E-publishing [13](#) gerecht. Trotzdem ist die IETF ein wichtiges Standardisierungsgremium und daher sollte die Entwicklung dieses Ansatzes weiterverfolgt werden.

2.5 Kommerzielle Lösungen

Gegenwärtig bieten bereits mehrere Unternehmen eBook-Lösungen an. Die wichtigsten w: Adobe Systems Incorporated, , Microsoft Corporation, Gemstar eBook Group Limited und Versaware Technologies, Incorporated. Soweit es Veröffentlichungen zur internen Architektur dieser Produkte gibt, sind diese vergleichsweise ungenau und ihre Aussagen kann nicht ohne Weiteres überprüft werden; das betrifft vor allem die Einhaltung bzw. Umsetzung bestimmter Standards und Spezifikationen. Um so besser sind jedoch die Kundenschnittstellen beschrieben. Daher beruht der Vergleich der verschiedenen Lösungen vor allem auf der Analyse der Schnittstellen.

2.6 Adobe Content Server und Acrobat eBook Reader

Die Lösung besteht aus dem *Adobe Acrobat eBook Reader* und *Adobe Content Server Software*, die nicht für die Buchautoren selbst, sondern für Provider gedacht sind, die dem Autor das eBook generieren und den Vertrieb im Internet ermöglichen.

Beide Produkte zusammen setzen wesentliche Teile der Architektur des EBX Systems um. Das Belegkonzept, das den Schutz der Rechte sichert, ist implementiert. Andere grundlegende Merkmale aus dem EBX Anforderungskatalog sind unberücksichtigt geblieben:

Der Adobe Acrobat eBook Reader ist ausschließlich auf MicroSoft Windows und Apple Macintosh Systemen verfügbar. Eine Linux-Version ist nicht vorhanden und nicht geplant.

Adobe benutzt als Basisformat sein proprietäres *Portable Document Format* (PDF).

Buchautoren geht folgende drei Schritte, um ihre Bücher im Netz anzubieten:

1. Erzeuge aus den Buchquellen (z. B. Wort-Dokument) eine PDF-Datei (z. B. mit dem Adobe Acrobat Distiller).
2. Füge der PDF-Datei spezielle eBook-Merkmale hinzu. Dazu gehören im engeren Sinne Vorschaubilder (*thumbnails*), die im Schaufenster des eBook-Verkäufers und in der eBook-Bibliothek des Lesers als Deckblatt verwendet werden, und ein Inhaltsverzeichnis mit Verweisen. Im weiteren Sinne gehören Lesezeichen, Hypertext-Verweise und Vorschaubilder für einzelne Seiten dazu.
3. Lade die PDF-Datei auf einen Adobe Content Server und bestimme die Rechte, die Leser auf dem Buch haben (drucken, Text markieren, kopieren usw.).

Adobe sieht im Zusammenhang mit der Liste der Leistungsmerkmale des eBook Readers, folgende Merkmale als wesentlich für den Buchleser (ADb 2001, ?) an :

- Außerordentlich geringe Kosten bei der Aktualisierung des Buchinhalts im Vergleich zu herkömmlichen Neuauflagen (sowohl für Verleger und Autor, als auch den Buchleser)
- Universelles Format, das portabel ist, dynamischen und interaktiven Inhalt anzeigen kann
- Intuitive Navigation durch den Inhalt
- Volltextsuche über dem Inhalt, Text hervorhebung, Lesezeichen
- Texte können vorgelesen werden
- Eingebettete Links stellen Verbindungen zu anderen Inhalt im Internet her.
- Bücher gliedern sich in ein Bibliothekskonzept ein, das Funktionen zur Ablage, Übersicht und Suchfunktionen zur Verfügung stellt.

Das PDF enthält alle Informationen, um diese Funktionalität umzusetzen. Es ist zu prüfen, ob MD*Book Formate ebenfalls alle Informationen enthalten, um diese Funktionalität eines eBook Readers zu ermöglichen.

Adobe's Zielgruppen für die Anwendung des eBook Reader sind vor allem Schüler, Studenten und Berufstätige mit mobilen Arbeitsplätzen.

Der Adobe Content Server implementiert mehrere Rollen in der Epublishing-Kette:

- Er kann Bücher von Autoren entgegennehmen, packen, in das verschlüsselte eBook-Format bringen und speichern (*Publisher*).
- Er kann eBook-Anfragen von Einzelhändlern und Verlegern entgegennehmen und an sie verteilen (*Distributor*, Lieferant).
- Er kann selbst Kundenanfragen entgegennehmen und eBooks ausliefern. Ist das gewünschte eBook nicht in seiner Datenbank, holt er es zuerst von anderen eBook-Lieferanten und liefert es dann aus (*Seller*).

Sowohl als Distributor als auch als Seller implementiert der Server die EBX System Schnittstellen.

Adobe hat also bereits eine eBook Lösung (seit 29 Januar 2001 bekannt), die sich an bestehenden Standards orientiert und die bereits seit Anfang dieses Jahres eingesetzt wird.

Der Adobe Content Server läuft nur auf MicroSoft Windows Systemen.

2.7 MicroSoft Digital Asset Server

MicroSoft verwendet ein ähnliches Konzept wie Adobe: Zum Lesen von MicroSoft eBooks wird der MicroSoft eBook Reader benutzt (an Stelle des Adobe Acrobat eBook Reader). Die Aufgaben des Adobe Content Server übernimmt der *Digital Asset Server* (DAS). Die Aufgabe des Acrobat Distiller, ein PDF zu erzeugen, übernehmen kommerzielle Dienstleister (Overdrive, ContentGuard,

Lightning, Source, Reciprocal, Versaware). Sie bauen auf dem *Microsoft eBook Content Software Developer's Kit* (SDK) auf, mit dem man Publisher-Werkzeuge programmieren kann, die Microsoft eBooks erzeugen. Die bereits existierenden Werkzeuge sind zum Teil sogar in der Lage, alle anderen wichtigen eBook-Formate zu erzeugen (Adobe eBooks und Gemstar eBooks).

Zur Übertragung der Lizenzangaben zwischen den Digital Asset Servers und Microsoft eBook Readers wird XrML benutzt.

Der Microsoft eBook Reader ist nur auf Windows Systemen und Handhelds bestimmter Hersteller (Windows CE) verfügbar.

Jedes Buch besteht aus einem OEB Package (.opf Datei) und Dokumenten der OEB Core Media Types. Allerdings handelt es sich nicht zwangsläufig um standardkonforme OEB publications, denn es gibt im SDK eine Möglichkeit, Dokumente mit Microsoft spezifischen Erweiterungen einzubinden. Obwohl Microsoft's eBooks die inhaltliche Struktur von OEB publications besitzen, ist das Austauschformat das binäre lit-Format. Das lit-Format ist eine Entwicklung Microsofts und liegt nicht offen vor (nicht öffentlich dokumentiert und erweiterbar).

Microsoft's Verständnis von eBooks und dem (eigenen) eBook Format ist (MSa 2000):

- low cost publication
- authors can self-publish
- easy access - you can download them from the internet
- they are searchable
- they have a dictionary lookup feature
- they can be annotated
- the interface (the reading area) is customizable for specific needs and circumstances
- they are compact - you can carry many books in a small package
- they are easy to update
- Live text
- One book, many devices
- Smaller page size
- Fixed aspect ratio
- Page turns, rather than scrolling
- Sophisticated typography (Microsoft ClearType)

Microsoft's eBook Reader beherrscht, bis auf die Fähigkeit Texte vorzulesen, alle Leistungsmerkmale des Adobe eBook Readers [15](#).

Microsoft bietet also auch eine eBook Lösung an, die sich zwar an bestehenden Standards orientiert, aber gegenwärtig nur auf Windows Systemen benutzt werden kann.

2.8 Gemstar

Gemstar ist der führende Hersteller von eBook Hardware. Gemstar hat bis auf eine FAQ-Liste keine schriftlichen Veröffentlichungen zu der Architektur und den Schnittstellen seiner Geräte herausgegeben. Gemstar Geräte lesen OEB Publications und HTML-Dateien ein. Der Schutz der Autorenrechte ist mit Hilfe der Gemstar Windows Software eBook Librarian gelöst. Sie stellt die Verbindung zu eBook-Verkäufern her und lädt eBooks auf das Gerät. Gemstar benutzt allerdings ein weniger mächtiges Modell als das EBX System. So ist es z. B. nicht möglich, Bücher zu verschenken und in öffentlichen Bibliotheken zu verleihen.

Allerdings arbeiten Gemstar und Microsoft zusammen, so daß damit zu rechnen ist, daß der Microsoft eBook reader in zukünftigen Versionen Microsoft eBooks auf Gemstar Geräte laden kann und so die komplette Funktionalität des Microsoft Epublishing-Modell genutzt werden kann.

Gemstar verfügt über eine Vielzahl von Partnern, die Bücher in das eBook-Format konvertieren. Der Schwerpunkt wird dort nicht auf die automatische Erzeugung eines eBook durch ein Werkzeug gelegt, sondern auf die Betreuung des Autors und manuelle Optimierung des eBooks.

Gemstar nimmt im Übrigen an, daß die Entwicklung weiterer Standards, die bestehenden eBook-Lösungen kaum noch berühren wird. Ob Gemstar Geräte zukünftig PDF Dateien interpretieren können ist noch nicht entschieden.

2.9 Andere Ansätze

Es gibt eine Vielzahl weiterer Software-Produkte kleinerer Unternehmen, die eBook Lösungen anbieten. Sie bauen jedoch nicht auf den oben angeführten Standards auf, sondern verwenden wesentlich einfachere Architekturen, die nur einen Bruchteil der Funktionalität der obengenannten Lösungen umsetzen.

eBook Software Compiler und Lesegeräte

- Activ E-Book Best Buy
- eBook Edit Pro +Bonus
- WebCompiler
- NeoBook
- E-ditor 2.0 (Just updated!)
- Illuminatus v4.51
- E-Book Publishing Suite
- WinEbook
- HTML2EXE
- eBookoMatic
- E-Publisher Gold
- E-Book Publishing Wizard
- EBook Pro Compiler
- HTML Compiler Pro/Pro+
- MediaPacker 2.0
- HyperMaker HTML 2000
- Infinite Ink
- ePublish Now
- WebExe 1.3
- EbookPaper
- KeeBook
- SBookBuilder 3.0
- EBook Builder 4.0
- eBookIt
- HotSend
- Electronic Book Shell
- Gymnast
- GoHTM

Diese Werkzeuge spezialisieren sich auf zwei Gebiete. Zum einen fungieren sie als Konverter bestehender Quellen in typische eBook-Formate, wie HTML und PDF. Zum anderen kapseln sie diesen Inhalt in einem Reader, der den Zugriff auf den Text (Copy/Paste) beschränkt.

Nahezu alle Produkte sind an das Windows Betriebssystem gebunden. Das bringt allerdings den Vorteil, daß die komplette Funktionalität des Internet Explorers in eBooks eingebracht werden kann. Dazu gehört z. B. die Darstellung von Macromedia Flash-Videos und der Aufruf beliebiger Browser-Plugins.

Kapitel 3

MD*Book im Vergleich

Obwohl zur Zeit noch kein eBook Standard verabschiedet ist, haben die wichtigsten eBook Anbieter verschiedene Entwürfe der Standardisierungsgremien bereits aufgegriffen und in ihren Produkten umgesetzt. Sie stellen den Autoren Werkzeuge (z. T. von Drittanbietern) zur Verfügung, die die Autoren bei der Transformation ihrer Bücher in ein eBook-Format unterstützen. Das fertige eBook wird dann auf ihren Servern verwaltet, vertrieben und verkauft. Autoren sind in der Wahl ihres Quelldatei-Formats nur dann beschränkt, wenn keine Werkzeuge erhältlich sind, die das gewünschte Quelldatei-Format in das bevorzugte eBook-Format konvertieren.

MicroSoft, Adobe und GemStar haben erhebliche Entwicklungsarbeit (z. B. MicroSoft ClearType und Adobe CoolType) geleistet, um eBook Reader zu entwerfen, die auf verschiedensten Geräten lauffähig sind und den Schutz der Autorenrechte gewährleisten.

In diesem Umfeld braucht man für MD*Book keine eigenständige Lösung zum Vertrieb der Bücher und Schutz der Autorenrechte entwickeln. MD*Book ist in dieser Kette ein Werkzeug, das den Autor bei der Erzeugung des eBooks unterstützt, so daß dieser unmittelbar mit dem Vertrieb des Buches fortfahren kann. Das Format, auf dem ein eBook basiert, ist jedoch nicht festgeschrieben. Grundsätzlich ermöglicht die OEB Publication Structure mit dem Fallback-Attribut, Dokumente beliebigen Typs einzubinden, denn wenn der jeweilige eBook Reader dieses Format nicht interpretieren kann, greift er auf alternative Dokumente anderer Typen zurück. Die Wahl des Formats verbleibt also. Ebenso ist die Anwenderschnittstelle und die Architektur des eBook Werkzeugs zu optimieren.

3.1 Anforderungen

Die Vielfalt der Anforderungskataloge sowohl von Standardisierungsgremien als auch Anbietern kommerzieller eBook Lösungen erfordert es, ihre Gesamtheit zu einem projektspezifischen Anforderungskatalog zusammenzufassen, der bereits die Auslagerung des Vertriebes und des Schutzes der Autorenrechte berücksichtigt.

Einige zusätzlich angeführte Anforderungen ergeben sich direkt aus projektspezifischen Problemen, wie z. B. der Personalausstattung und der bereits gegebenen Architektur. Ein Teil der Anforderungen ergibt sich aus der Tatsache, daß MD*Book vor allem die spezielle Zielgruppe der Lehrenden und Lernenden ansprechen will.

Ein, diesen besonderen Einschränkungen genügendes, eBooks erzeugendes Werkzeug muß daher folgende Anforderungen erfüllen.

Flexibles Format

Elektronische Bücher liegen in einem möglichst flexiblem (mächtigem) Format vor, so daß das Buchformat den speziellen Bedürfnissen der Leser entgegenkommt:

Gedrucktes Buch Der Inhalt des e-books wird auch in gedruckter Form und mit gutem Textsatz angeboten.

Obwohl die interaktiven Elemente nicht ausgedruckt werden, enthält die gedruckte Version stattdessen an geeigneten Stellen Verweise auf diese Elemente, aus denen ersichtlich wird, welches Verhalten dort in der interaktiven Version zu erwarten ist. Auf diese Weise kann der Leser der gedruckten Version die interaktive Version gezielt als Ergänzung benutzen.

Freie Struktur Die Strukturierung des Inhalts und die Art der Navigation in diesem Inhalt ist möglichst frei gestaltbar.

Elektronische Bücher sind auf einer Vielzahl von elektronischen Geräten nutzbar, das betrifft zukünftig im besonderen Handhelds und PalmTops, obwohl jedes Gerät eine spezifische Art der Navigation und Strukturierung des Inhalts verwendet. Auf jedem dieser Geräte soll man intuitiv navigieren können.

Wiederverwendbar Einzelne Abschnitte sind in anderen Dokumenten wiederverwendbar.

Einzelne Wissen- und Lerneinheiten besitzen eine klare Schnittstelle zu anderen Einheiten, so daß sie einzeln abgelegt und wiederverwendet werden können. Optimal ist ein Ablagesystem für alle Wissenseinheiten, das automatisch Informationen über Querverweise zwischen den Wissens- und Lerneinheiten verwaltet.

Konvertieren Bestehende Bücher können leicht in das Format eines e-books konvertiert werden.

Bücher, die bereits in anderen Formaten vorliegen, sind leicht und ohne Informationsverlust in das e-book Quellformat zu konvertieren und problemlos um interaktive Elemente zu ergänzen.

Mächtige Sprache Die Beschreibungssprache der interaktiven Elemente ist mächtig genug, um den Anforderungen der Lehrenden, hier speziell der Statistiker und Mathematiker, zu genügen.

Im Idealfall kann man alle denkbaren Interaktionsmöglichkeiten mit der Sprache formulieren. Ist das nicht der Fall, kann der Autor fehlende Konstrukte einbinden.

Freies Layout Das Layout des Buches kann an fachspezifische Besonderheiten und persönliche Präferenzen des Autors angepasst werden.

Die grafische Gestaltung von Büchern ist aus didaktischer Sicht vom Fachgebiet und von der Zielgruppe abhängig. Hochschulstudenten haben andere Lesegewohnheiten als Fachhochschulstudenten und Gymnasiasten.

On/Offline Man kann e-books sowohl über das Internet als auch offline (CD oder Download vom Internet) benutzen.

Nur die Veröffentlichung im Internet ermöglicht Querverweise zu anderen e-books und die weltweite Verfügbarkeit des Buches. Weiterhin kann man e-books direkt in Lehrveranstaltungen einsetzen.

Zugriffsbeschränkung Der Autor kann den Zugriff auf Teile oder das gesamte Buch in Abhängigkeit verschiedener Bedingungen oder bedingungslos beschränken.

Diese Eigenschaft ermöglicht dem Autor den kommerziellen Vertrieb seines Buches.

Anwendergerechte Bedienung

Buchautoren können auf unkomplizierte, leicht verständliche Art und Weise, Bücher erzeugen:

Vertraute GUI Das Werkzeug wird über vertraute Benutzerschnittstellen bedient.

Auf den Plattformen UNIX/Linux, MS Windows und MacOS sind das grafische Benutzerschnittstellen, die mit dem Dialog/Fenster-Konzept arbeiten. Zur zweitwichtigsten Benutzerschnittstelle haben sich auf allen Plattformen WWW Seiten entwickelt.

Plattformunabhängigkeit Das Werkzeug ist auf allen Plattformen verfügbar, die von Buchautoren benutzt werden.

Akzeptanz Die Autoren müssen sich nicht in neue Programme und Sprachen einarbeiten, um das Werkzeug zu nutzen.

Gerade die Vielzahl der bereits existierenden Textbearbeitungsprogramme legt es nahe, bestehende Technologien zu verwenden. Autoren akzeptieren neue Werkzeuge nur bei geringen Einarbeitungszeiten.

Vertraute Programmiersprache Die interaktiven Elemente kann man schnell erstellen und einbinden.

Die Beschreibungssprache der interaktiven Elemente trägt im Gegensatz zu der gestaltungsorientierten Sprache des statischen Textes algorithmischen Charakter. Sie soll den üblichen Konventionen von algorithmischen Sprachen (Programmiersprachen) genügen.

Kurze *turn-around*-Zeit Die *turn-around*¹-Zeiten sind auch bei sehr grossen Textobjekten wie Büchern nur kurz.

¹Die *turn-around*-Zeit ist die Zeit, die der Autor benötigt, um den Text zu editieren, zu übersetzen, Rückmeldungen zu bekommen und wieder zu editieren. Diese Kennzahl spielt eine große Rolle beim Aktualisieren und Korrigieren bestehender Texte.

WYSIWYG²-Textprogramme überführen jede einzelne Texteingaben sofort in grafischen Output. Die *turn-around*-Zeit ist dort null. ASCII-Formate dagegen sind auf ein andere Programme angewiesen, die den ASCII-Text in die grafische Repräsentation überführen. Die *turn-around*-Zeit steht bei diesen Formaten in direkter Abhängigkeit von der Größe des Quelldokuments und der Menge und Komplexität der Transformationsregeln.

Teamarbeit Mehrere Autoren können zusammen an einem Buch arbeiten.

Autoren können gleichzeitig an einem Buch arbeiten, ohne sich zu behindern. Querverweise zwischen Kapiteln unterschiedlicher Autoren werden aufgelöst, wenn es möglich ist. Querverweise ohne auflösbare Ziele behindern nicht die Erstellung von Zwischenversionen.

Mehrsprachenfähigkeit Die Sprache der Anwenderschnittstelle ist länderspezifisch.

Buchautoren müssen nicht zwangsläufig Englisch Kenntnisse besitzen, um MD*Book Online zu bedienen.

Neben diesen e-book spezifischen Anforderungen an eine anwendergerechte Bedienung sind die allgemeinen Grundprinzipien der Benutzerführung umzusetzen (SMI 1988):

- Selbsterklärungsfähigkeit,
- Aufgabenangemessenheit (Problemangemessenheit),
- Steuerbarkeit (Kontrollierbarkeit),
- Fehlerrobustheit (Fehlertoleranz), und
- Erwartungskonformität (Verlässlichkeit).
- Konsistenz, Transparenz (Durchschaubarkeit), Handlungsdirektheit und Individualisierbarkeit

Software-Qualität

Das Werkzeug ist leicht zu warten, erweiterbar, zukunftssicher und portierbar:

Bestehende Software verwenden Das Werkzeug baut in einem möglichst hohem Maß auf bestehender Software auf.

Zum einen wird der Entwicklungsaufwand so gering wie möglich gehalten und zum anderen die Pflege des Werkzeugs auf den eigenen Teil beschränkt. Auf diese Weise können Fehler besser zugeordnet und die Fehleranfälligkeit verringert werden.

Standards einhalten Es werden ausschließlich Standardformate, -Protokolle und -Schnittstellen verwendet,

- die in einer stabilen Version vorliegen,

²What You See Is What You Get

- plattformübergreifend verbreitet sind und
- deren zukünftige Unterstützung über mehrere Jahre hinweg gesichert ist.

Modulare Architektur Die Architektur des Werkzeugs soll modular sein.

Veränderungen, Aktualisierungen einzelner Komponenten sind sowohl für Entwickler als auch Benutzer wesentlich einfacher umzusetzen bzw. klarer wahrzunehmen.

Gängige Programmiersprache Das Werkzeug ist in einer Sprache implementiert, mit der eine große Zahl von Entwicklern vertraut ist.

Auf diese Weise kann man sichern, daß Personal für die Weiterentwicklung gefunden wird.

Programmierkonventionen einhalten Bei der Implementierung werden professionelle Programmierkonventionen eingehalten.

Eine herausragende Rolle spielt dabei die ausführliche Dokumentation.

3.2 MD*Book's Eigenschaften

Flexibles Format

Gedrucktes Buch

L^AT_EX ist als Textsatz-System konzipiert und daher erzielen die PDF- und PostScript-Ausgabeformate des MD*Book optimale Druckergebnisse. Die Makros `label` und `pageref` unterstützen Verweise im Text.

Freie Struktur

MD*Book-Bücher sind aus einer Abfolge von Kapiteln und Abschnitten aufgebaut, so wie sie in L^AT_EX unterstützt werden. Die Verweisteknik erlaubt es jedoch, die Struktur des Buches der HTML-tyischen Informationsnetz-Struktur (*Web*) anzunähern. Die Navigation ist in den drei Ausgabeformaten PDF, HTML und HTML mit Javascript besonders gut umgesetzt. In den PDF Repräsentationen ist das Inhaltsverzeichnis mit Verweisen versehen und es wird eine Leiste mit Lesezeichen erzeugt. In der HTML Repräsentation mit Frames steht ebenfalls eine solche Leiste zur Verfügung. In der HTML Repräsentation mit Javascript wird neben dem mit Verweisen gefüllten Inhaltsverzeichnis ein Karteikartensystem umgesetzt, das die Navigation im Inhalt des Buches weiter vereinfacht (nützlich für didaktische Zwecke). Ausserdem werden dem Benutzer automatisch gesetzte Lesezeichen auf die zuletzt gelesenen Abschnitte angeboten. Handhelds und Palms, die nur über einfache Browser-Technologien verfügen, benötigen jedoch HTML-Seiten mit Verweisen auf vorhergehende und nachfolgende Seiten. Derartige Seiten werden von MD*Book nicht generiert.

Wenn auf diesen Geräten jedoch eBook Reader eingesetzt werden können, dann werten diese das Inhaltsverzeichnis des PDF-Dokuments aus oder das als Inhaltsverzeichnis deklarierte Dokument der HTML Repräsentation mit Frames.

Wiederverwendbar

\LaTeX -Abschnitte können natürlich in anderen \LaTeX -Dokumenten wiederverwendet werden, aber Querverweise gehen verloren oder müssen manuell angepaßt werden. Die Struktur der \LaTeX -Dokumente beinhaltet, im Gegensatz zu Markup-Sprachen, keine Semantik, sondern lediglich Informationen für den Textsatz. Allerdings hat sich bisher kein Textverarbeitungs- und Textsatzsystem durchsetzen können, das vorwiegend die semantischen Bestandteile des Buches beschreibt. Nur mit einer solchen Sprache liessen sich Querverweise auflösen bzw. automatisch Querverweise zwischen Dokumenten erzeugen lassen.

Konvertieren

Die Konvertierung von einem mit MD*Book erzeugten PDF-Buch in ein Adobe eBook ist nahezu trivial und ohne Informationsverlust zu bewerkstelligen. Allerdings werden Verweise auf HTML Seiten (Einbindung der XploRe Quantlet Dokumentationen) werden gegenwärtig nicht unterstützt. Damit entfällt auch die Ausführung der Quantlets.

Die Konvertierung in das MicroSoft eBook Format ist etwas komplizierter. Obwohl der Dokumenttypen, die in OEB publications angegeben werden können, grundsätzlich frei wählbar sind, muss immer eine Fallback-Repräsentation jedes Dokuments verfügbar sein. Der Dokumenttyp der Fallback-Repräsentation muß ein **OEB Core Media Type**³ sein. Die Generierung dieser Repräsentation ist natürlich um so einfacher, je näher der ursprüngliche Dokumenttyp einem OEB Core Media Type gleicht. OEB Core Media Types sind PNG and JFIF(JPEG) (nicht GIF) und OEB Document, wobei OEB Document ein XML Dokumenttyp ist, der eine Teilmenge der HTML4.0/XHTML1.0-Markups und CSS 1-Anweisungen ist. Daher ist das MicroSoft eBook entweder als Grafik oder HTML-ähnliches Dokument zu erzeugen. Da das, von MD*Book eingesetzte, \LaTeX 2HTML HTML3.2-Dateien erzeugt, ist der Weg von der HTML Repräsentation zum MicroSoft eBook am kürzesten.

Die Konvertierung eines bestehenden Buches in das MD*Book Quellformat (\LaTeX), um dann mittels MD*Book ein eBook zu erzeugen, ist jedoch wesentlich komplizierter. Grundsätzlich stehen zwar Werkzeuge für die Konvertierung von HTML und MicroSoft Word nach \LaTeX zur Verfügung, aber der Umweg über ein solches Zwischenformat stellt für den Autor eher ein Hindernis dar und ist in der Regel mit Informationsverlust behaftet.

Mächtige Sprache

MD*Book ist gegenwärtig auf XploRe festgelegt, um interaktive Komponenten in das eBook zu integrieren. Allerdings ist XploRe als matrix-orientierte Sprache einerseits mächtig genug um komplizierte mathematische und statistische Probleme zu lösen und andererseits auch in der Lage dem Benutzer Dialoge

³OEB Core Media Type: A MIME media type that all reading systems must support (OEB 1999).

anzubieten und grafische Ergebnisse auszugeben. Sollten weitergehende Interaktionsmöglichkeiten gewünscht werden, kann mit Hilfe der `rawhtml`-Umgebung HTML-Code für beliebige Plugins eingefügt werden. Diese eingebundenen *Objects* (im OEB package-Kontext) müssen immer mit einem Fallback-Attribut versehen werden, damit der MicroSoft Reader oder andere OEB publication Lesegeräte diese Objekte in jedem Fall interpretieren können.

Freies Layout

Allein \LaTeX bietet vielfältige Gestaltungsmöglichkeiten, aber insbesondere CSS-Dateien können die Gestaltungsmöglichkeiten der HTML-Repräsentationen erweitern.

On/Offline

MD*Book legt die erzeugten Ausgabeformate einfach in Unterverzeichnissen ab. Es ist jedoch wünschenswert die erzeugten Bücher bereits im WWW lesen zu können und auch in einer Datei zusammengefaßt herunterladen zu können.

Zugriffsbeschränkung

Diese Funktionalität wird von MD*Book nicht unterstützt. Das ist aber auch nicht notwendig, denn im oben angeführten Konzept übernehmen die Epublising Server diese Aufgabe.

Anwendergerechte Bedienung

Vertraute GUI

MD*Book kann nur innerhalb eines LANs über die Kommandozeile benutzt werden, d. h. Autoren müssen sich erst einen UNIX-Zugangaccount im Netz des Statistik-Instituts besorgen und dann mit dem `telnet` MD*Book bedienen. Optimal ist es, alle MD*Book Funktionen im WWW nutzen zu können (mit Java Applet), anstatt mit Kommandozeilen und UNIX-Shell-Skripten zu arbeiten.

Plattformunabhängigkeit

MD*Book läuft gegenwärtig nur auf UNIX Systemen, genauer Linux. Mit einer webbasierten Schnittstelle kapselt man die Beschränkung auf dieses Betriebssystem.

Akzeptanz

\TeX / \LaTeX ist im Grunde die Sprache der Wahl, wenn es notwendig ist, mathematische Formeln zu Papier zu bringen. Aus diesem Grund ist es in den Fachrichtungen Mathematik und Statistik weit verbreitet und stellt keine Hemmschwelle für Autoren dar.

Vertraute Programmiersprache

Die interaktiven Elemente werden in der Programmiersprache XploRe formuliert. Diese Sprache ist noch nicht weit verbreitet. Sie lehnt sich aber immerhin an gängige Programmiersprachen an und ist umfangreich dokumentiert. Allerdings ist die Programmierung von Dialogen in anderen Sprachen noch aufwendiger, so daß XploRe für diesen Anwendungsfall gut geeignet ist.

Kurze *turn-around-Zeit*

Die turn-around-Zeiten sind bei grossen Büchern unzumutbar (\LaTeX -Technologie) und liegen schon bei kleinen Dokumenten im Minuten-Bereich. Die einzige Möglichkeit, die Zeiten zu verkürzen, ist, das Buch temporär zu zerlegen und immer nur einzelne Teile des Buches zu übersetzen.

Teamarbeit

MD*Book selbst unterstützt noch keine Teamarbeit. Wird MD*Book von zwei verschiedenen Konsolen gleichzeitig, können DeadLocks⁴ und undefinierte Ausgaben auftreten. Ein einfacher Lock-Mechanismus behebt dieses Problem.

Mehrsprachenfähigkeit

MD*Book soll zukünftig internetweit eingesetzt werden. Daher ist die webbasierte Schnittstelle mehrsprachenfähig zu machen.

Software-Qualität

Bestehende Software verwenden

Im MD*Book kapselt im wesentlichen anderen Werkzeuge, so daß der Entwicklungsaufwand für MD*Book selbst so gering wie möglich gehalten wird. Bei dem Vertrieb der eBooks können Lösungen von Adobe oder MicroSoft eingesetzt werden.

Standards einhalten

\LaTeX liegt in einer stabiler Version vor und ist plattformübergreifend. Die erzeugten Formate und verwendeten Protokolle sind entweder Quasistandards (z.B. PDF) oder echte Standards (z.B. HTTP, HTML, CSS). Die Interaktivitätssoftware XploRe ist allerdings nicht frei verfügbar (weder die Quellen noch die Binaries).

⁴Mehrere Prozesse belegen die gleichen Ressourcen und blockieren sich dadurch.

Modulare Architektur

Wegen dem hohen Verwendungsgrad anderer Werkzeuge, ist die Architektur automatisch modular. Die webbasierte Schnittstelle sollte so weit wie möglich

3.3 Fazit

MD*Book leistet bereits ein großen Teil, der in der ersten Anforderungsliste angegeben Funktionalität. Interaktivität ist auf die Fähigkeiten von Xplore beschränkt. Jede Fachrichtung verwendet eigene hochspezialisierte Termini und Sprachen, um Problemstellungen zu beschreiben und zu lösen. Insofern ist es nicht sinnvoll, die Unterstützung auf eine allmächtige Sprache zu beschränken. Alle Problemlösungen müssten neu implementiert werden. MD*Book ermöglicht durch die Unterstützung der Umgebung `rawhtml` die Einbindung jeder HTML-Interaktionstechnologie. Dazu gehören Plugins, Applets, ActiveX Controls. Die Schwächen des Werkzeug liegen in der Umsetzung der zweiten Anforderungsliste. Abhilfe schafft hier eine webbasierte Schnittstelle, die zum einen die eigentliche MD*Book Funktionalität kapselt und zum anderen kleine Hilfsdienste anbietet, die MD*Book ergänzen. Die dritte Anforderungsliste setzt vor allem Maßstäbe für die Erweiterung des MD*Book.

Kapitel 4

Architektur

Die hier vorgestellte Architektur soll die im vorherigen Kapitel ermittelten Schwächen ganz oder teilweise beheben.

Wie bereits hergeleitet wurde, ist die optimale Schnittstelle WWW basiert. Interaktive Komponenten werden mit einer serverseitig interpretierten Sprache umgesetzt. PHP und JSP sind auf diesem Gebiet die verbreitetsten Konzepte. JSPs sind eng an die Sprache Java gekoppelt und haben daher ihren Rückhalt in der großen Java Entwicklergemeinschaft. Die Wartung der JSPs ist günstiger als die Wartung der PHP Seiten, weil Java als Programmiersprache wesentlich verbreiteter ist und entsprechendes Know How schneller zur Verfügung steht. Darum wurden JSPs verwendet.

4.1 Allgemeines Use Case Schema

Grundsätzlich lassen sich die Aktivitäten der Buchautoren in zwei Gruppen einteilen. Zum einen gibt es Aktivitäten, die zur Verwaltung des Benutzerkontos und der eigenen Bücher dienen und zum anderen gibt es Aktivitäten, die direkt der Erzeugung der Bücher dienen.

4.2 Accounts und eBooks verwalten

Die Verwaltung der eBooks und Accounts besteht darin die Erzeugung der Bücher vorzubereiten. Dazu benötigt man einen Arbeitsplatz auf dem MD*Book Server und Verzeichnisse in denen sowohl die Quelldateien als auch die erzeugten Bücher abgelegt werden können. Alle diese Tätigkeiten können durch insgesamt drei Dienste bewältigt werden. Ein Datenbankdienst speichert Benutzerdaten in einer Datenbank und liest sie aus. Ein Dateisystemverwaltungsdienst verwaltet Verzeichnisse an, deren Struktur konsistent mit den Daten in der Datenbank sind. Ein dritter Dienst authentifiziert die Anfragen der Autoren. Die Kombinationen dieser drei Dienste leisten alle erforderlichen Aufgaben für diesen Use case. Sie werden alle drei direkt über JSPs angesprochen.

Use cases

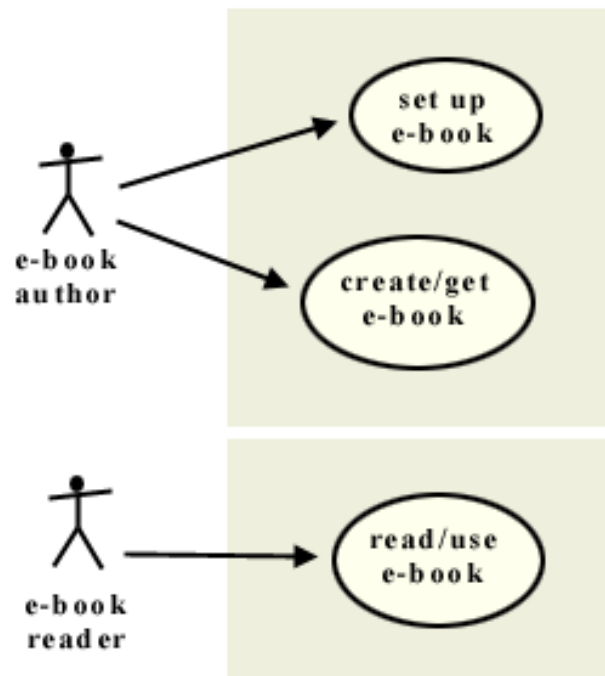


Abbildung 4.1: *use case diagram* Use MD*Book Online

Set up e-book

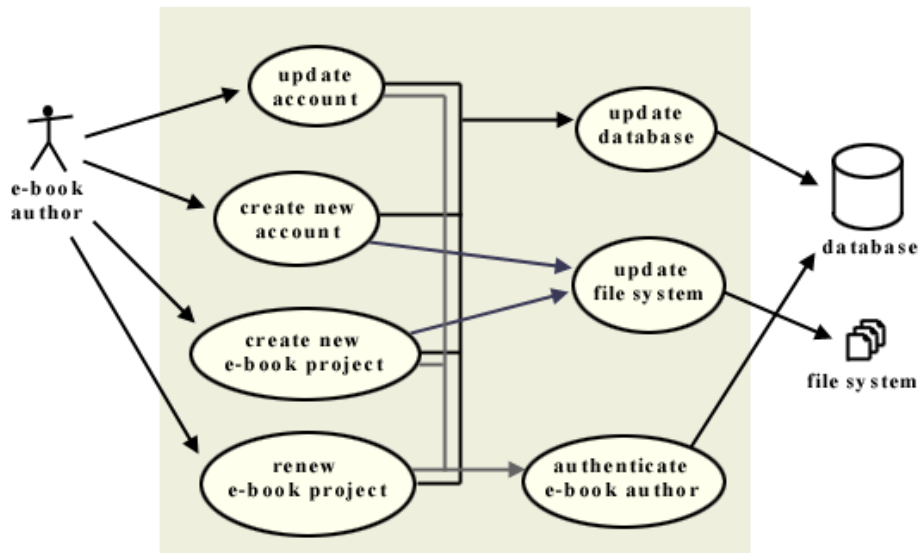


Abbildung 4.2: *use case diagram* Setup new eBook

4.3 eBooks erzeugen

Die Erzeugung der eBooks erfordert zum einen den Aufruf des MD*Book und zum anderen das Hochladen der Quelldateien und Herunterladen der Ergebnisse. Diese Aufgaben können zwei Dienste übernehmen, die jedoch eine direktere Interaktion mit dem Benutzer erfordern, als das mit JSPs möglich ist. Der verbreitetste Standard für Dateitransfers im Internet ist das File Transfer Protokoll (FTP). Also habe ich einen FTP Server so konfiguriert, daß die MD*Book Benutzer Zugriff auf ihre eigenen Verzeichnisse haben können ohne auf dem darunterliegenden UNIX-System über ein Benutzerkonto zu verfügen. Der zweite Server kann MD*Book starten. Er wird über ein Java Applet angesprochen.

4.4 Ablauf

Die folgenden Diagramme veranschaulichen das Zusammenspiel der einzelnen Komponenten des MD*Book Online Systems. Die linke Seite in der Komponentensicht zeigt ein mögliches Geschäftsmodell für die Abrechnung von Rechenzeiten auf XploRe Servern. Solange jedoch in der Open eBook Publication Structure-Spezifikation keine Angaben über die Behandlung externer Objekte (Browser-Plugins und Java Applets) gemacht werden, kann über die Authentifizierung der Buchleser und damit über das verbundene Geschäftsmodell nur spekuliert werden.

Create/get e-book

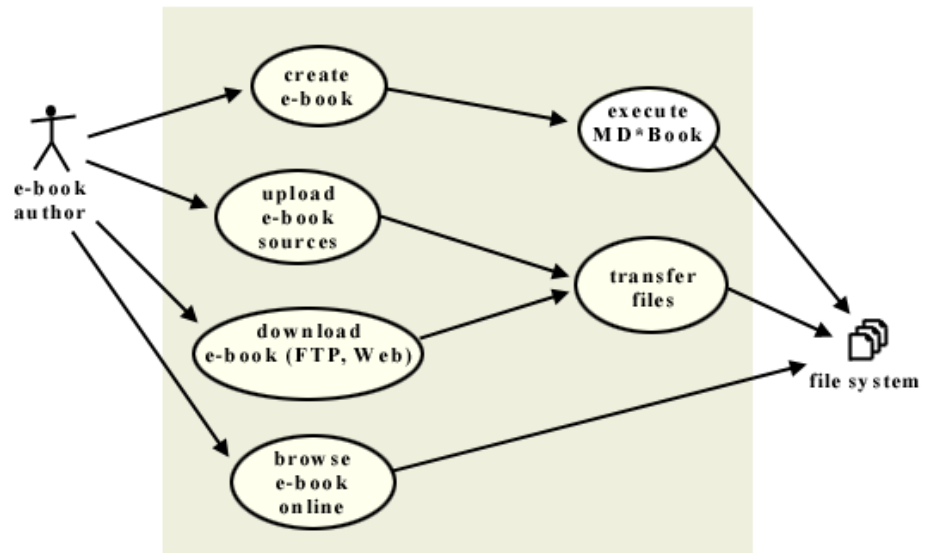


Abbildung 4.3: use case diagram Create ebook

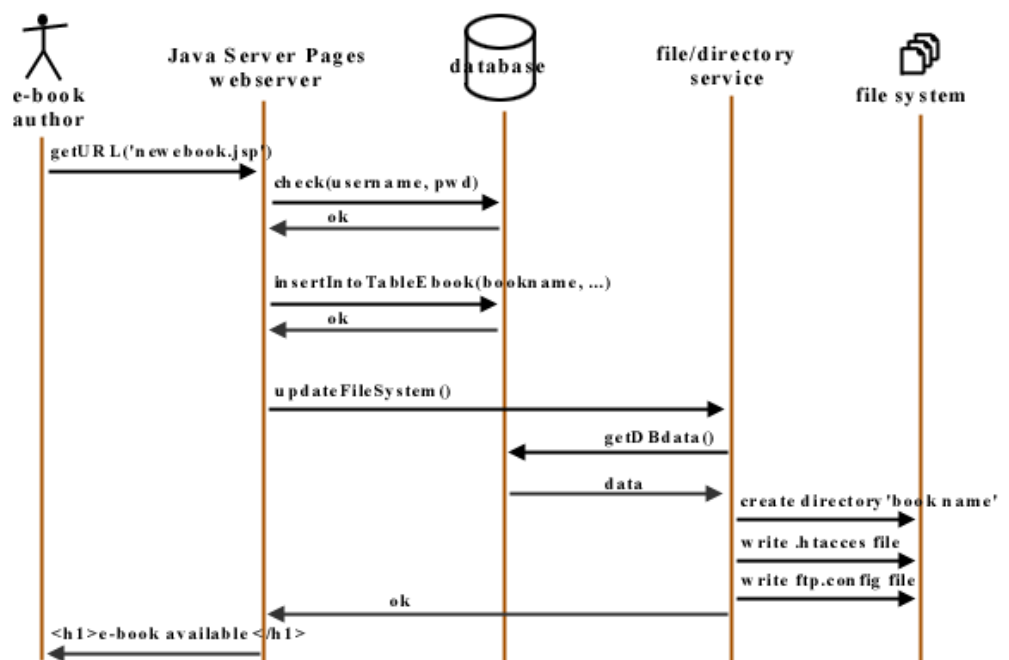


Abbildung 4.4: sequence diagram Create eBook

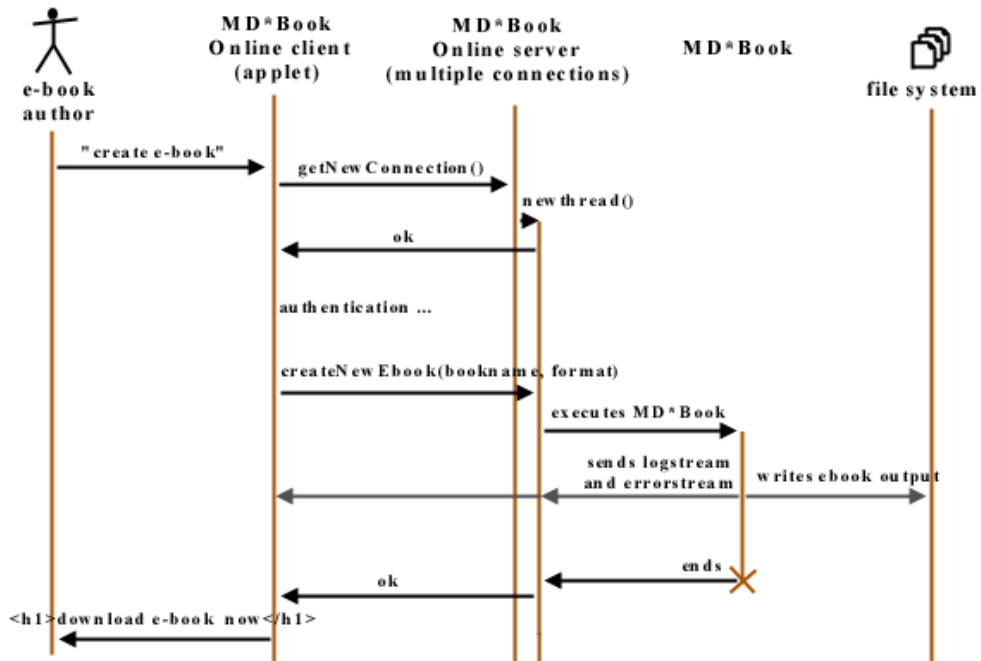


Abbildung 4.5: *sequence diagram* Execute MD*Book

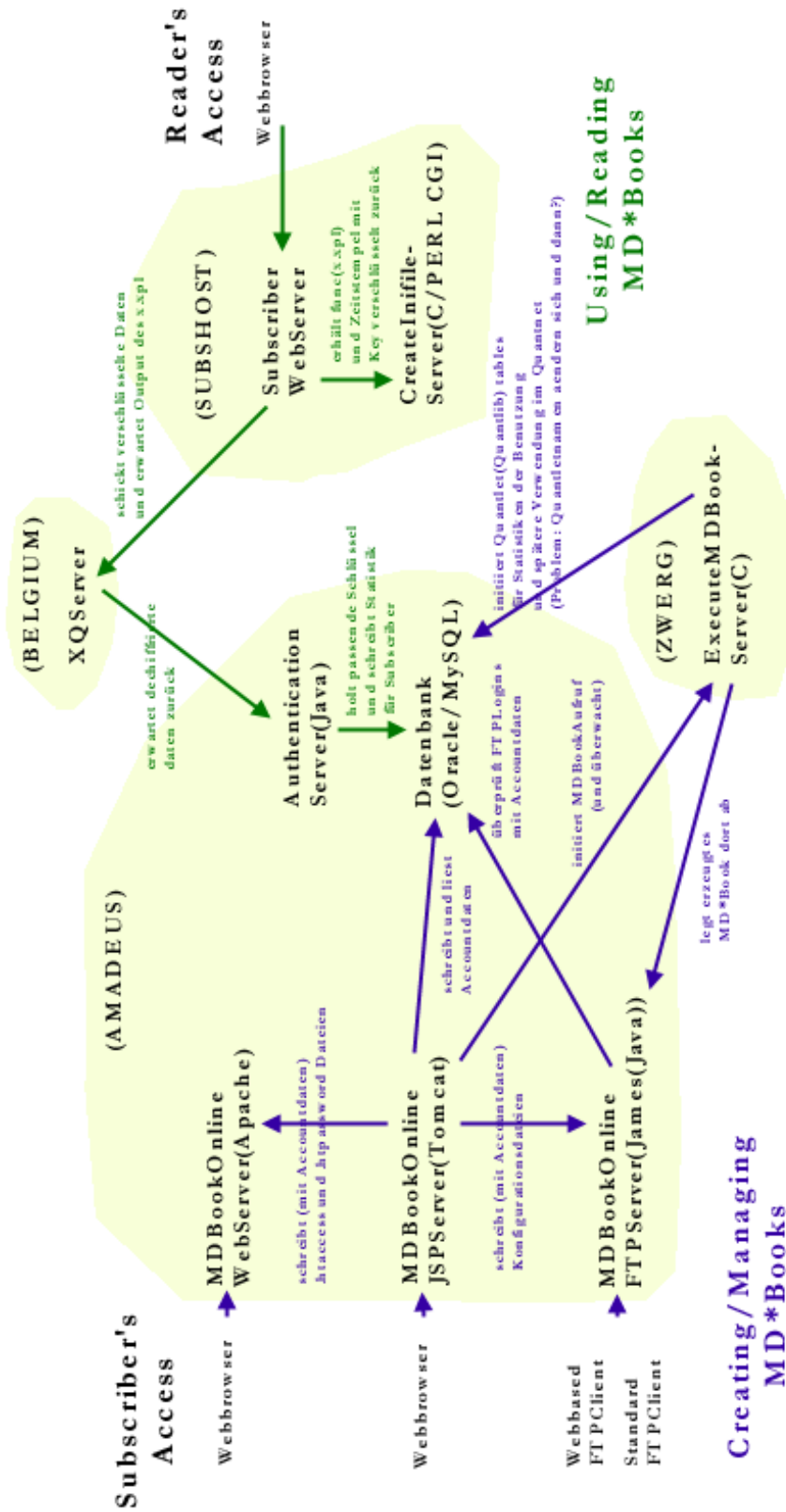


Abbildung 4.6: Komponentensicht

Kapitel 5

Ausführung

In diesem Kapitel werden die konkrete Ausführung im Überblick und einige ausgewählte Probleme, die bei der Umsetzung der Architektur, auftraten, beschrieben. Die exakte Signatur der Java-Klassen ist im Anhang angegeben¹.

5.1 Die Java Server Pages

Im Wesentlichen entspricht jede JSP einem der im obigen Kapitel identifizierten Use Cases.

- **bookcreate.jsp**: liefert eine HTML Seite, in die der MD*Book Online Client eingebunden ist.
- **booktransfer.jsp**: liefert eine HTML Seite, in die der webbasierte FTP Client eingebunden ist. Die Daten für das Farbschema des FTP Client werden einer Instanz der MBOHelper-Klasse entnommen.
- **bookonline.jsp**: liefert dem Benutzer eine Liste aller seiner Bücher, die Formate, in denen sie erzeugt wurden, das Datum, an dem sie generiert wurden und Verweise auf die Online-Versionen der generierten Bücher und auf mit ZIP gepackte Versionen der Bücher.
- **booknew.jsp**: ermöglicht dem Benutzer, ein neues Buch einzurichten.
- **bookprolong.jsp** ermöglicht dem Benutzer, die Verlängerung der Nutzungsdauer des Buches. Die begrenzte Nutzungsdauer von Büchern habe ich eingeführt, damit man zum einen zukünftige Verlängerungen an Bedingungen knüpfen kann und zum anderen, um stillgelegte Accounts zu erkennen.
- **accountnew.jsp** ermöglicht dem Benutzer, einen neuen Account einzurichten. Der Besitz eines Accounts ist Voraussetzung, um MD*Book Online zu benutzen.

¹Die Dokumentation der Java-Klassen ist noch nicht abgeschlossen. Daher geben einige der API-doc-Seiten im Anhang nur einen Überblick. Die Dokumentation wurde mit JavaDoc und dem L^AT_EX 2_ε-Doclet (erstellt

- **accountedit.jsp**: ermöglicht dem Benutzer seine Accountdaten zu aktualisieren.
- **accountnbookdata.jsp**:

Die JSP `datafromdb.jsp` erfüllt genau die gleichen Aufgaben wie `accountnbookdata.jsp`, jedoch werden die Daten aller Benutzer und eine Liste der in der Mailingliste eingetragenen Benutzer angezeigt. Sie ist für statistische Zwecke gedacht und kann nur vom MD*Book Online Provider eingesehen werden.

Jede JSP hat ihre eigene Authentifizierungsabfrage, denn das Nutzungsschema der JSPs weist keine (Verweis-)Ketten aufeinanderfolgender JSPs auf. Würden solche Ketten existieren, würde es Sinn machen, für jede HTTP-Session genau eine Authentifizierungsabfrage vorzunehmen.

Die JSP benutzen die Klassen `MBOHelper`, `AuthService`, `DBAccess` und `FileManager` aus dem Package `services`.

5.2 Das Package services

Das Package `services` wird vor allem von den JSPs benutzt.

Die Klassen `DBAccess`, `AuthService` und `FileManager` setzen die namensverwandten Dienste um, die sich aus den Use Case-Diagrammen ableiten. Die Klasse `MBOHelper` bietet Hilfsdienste an, die von mehreren Komponenten im MD*Book Online-Konzept gebraucht werden.

Die Klasse `MBOHelper` stellt installations- und sprachabhängige Daten zur Verfügung. Wenn zum Beispiel MD*Book Online die Datenbank wechselt, so wird diese Änderung der `MBOHelper`-Klasse bekannt gemacht und alle anderen Dienste, die auf diese Datenbank zugreifen, benutzen automatisch die neue Datenbank.

Folgende Daten lassen sich auf diese Weise in Abhängigkeit von der verwendeten Installation ermitteln:

- die URLs der JSP Seiten und damit indirekt das Wurzelverzeichnis des JSP Servers,
- die URL der, für das Layout der JSPs verwendeten, CSS Datei,
- die Wurzel-URL aller Benutzerverzeichnisse,
- die MD*Book Kontakt Adresse und die Adresse der Mailingliste,
- die Dateisystempfade, an denen folgende Dateien und Programme zu finden sind:
 - Perl (wird benutzt, um Passwörter zu verschlüsseln),
 - Schablonen für die Konfigurationsdatei des FTP Servers,
 - temporäre Konfigurationsdatei und permanente Konfigurationsdatei des FTP Servers,
 - Passwortdatei für den FTP Server,

- MD*Book,
- TeX2Sk,
- die mehrsprachigen Titel der JSPs,
- die mehrsprachigen Fehlermeldungen des MD*Book Online Servers,
- der Port und die Adresse (IP-Adresse oder DNS-Name) des FTP Servers,
- das Farbschema des FTP Servers,
- die .passwd-Daten des Benutzers unter dem der FTP Server läuft,
- der Port des FTPConfigWriter,
- die eingestellte Sprache,
- der Port und die Adresse (IP-Adresse oder DNS-Name) des MD*Book Online Server,
- das Farbschema für den MD*Book Online Client

Die Klasse `MBOHelper` stellt weiterhin Methoden zur Validierung der, in den Formularen einzugebenden, Daten zur Verfügung. Dabei werden die eingegebenen Daten auf die Beschränkungen getestet, die in der MD*Book Online Dokumentation beschrieben werden.

5.2.1 Die Klassen `FileManager` und `FTPConfigWriter`

Die Klasse `FileManager` übernimmt zwei Aufgaben.

Die erste Aufgabe ist es, das MD*Book Online Dateisystem zu pflegen. Das Dateisystem umfaßt die Organisation der Benutzerverzeichnisse und der Rechte auf diesen Verzeichnissen. Die Methode `updateFileSystem()` liest die Benutzer- und Buchdaten aus der Datenbank, erzeugt die entsprechenden Verzeichnisse, aktualisiert die Konfigurationsdateien des FTP Servers und schreibt Dateien, die die Zugriffsrechte auf die Benutzerverzeichnisse über den Apache Webserver beschränken.

Die zweite Aufgabe ist es, die vom MD*Book erzeugten Dateien zu verbinden und zu komprimieren (ZIP Archiv).

Mit dem `FileManager` werden ebenfalls Lock-Dateien geschrieben und gelöscht. Solange MD*Book auf einem bestimmten Buch ausgeführt wird, befindet sich eine Lock-Datei in dem Buchverzeichnis. Wird inzwischen versucht ein weiteres Mal, MD*Book aufzurufen, wird das abgelehnt.

Die Klasse `FTPConfigWriter` stellt einen `SocketServer` zur Verfügung, der die temporäre Konfigurationsdatei in die permanente Konfigurationsdatei des FTP Servers schreibt. Dieser Zwischenschritt ist nur dann notwendig, wenn der FTP Server auf einem anderen Rechner läuft als der MD*Book Online Server.

5.2.2 Die Klasse AuthService

Die Klasse `AuthService` verschlüsselt Passwörter und überprüft, ob gegebene Benutzerdaten (Benutzername und unverschlüsseltes Passwort) mit den Daten in der Datenbank übereinstimmen. Zur Verschlüsselung wird die Funktion `crypt()` benutzt, die auch Unix verwendet.

5.2.3 Die Klasse DBAccess

Die Klasse `DBAccess` stellt ein umfangreiches Repertoire von Zugriffsmethoden auf die Datenbank zur Verfügung. Allen Methoden ist jedoch gemeinsam, daß kein direkter Zugriff, im Sinne von SQL Anweisungen, auf die Datenbank möglich ist. So kann die gekapselte Datenbank gewechselt und das Tabellendesign geändert werden, ohne daß die Komponenten, die auf die Daten der Datenbank zugreifen, angepaßt werden müssen.

5.3 Das Package server

Das Package `server` ermöglicht zusammen mit den Packages `client` und `protocol` die Kommunikation zwischen dem MD*Book Online Server und dem MD*Book Online Client (Java Applet).

Der MD*Book Online Server ist ein `MultiSocketServer` (umgesetzt in der Klasse `MDBookOnlineServer`). D. h. er erzeugt einen Socket und weist dann jedem neuen Client ein eigenes Socket und Thread² zu. Diesen Client-spezifischen Verbindungen wird von der Klasse `SignalServer` behandelt.

Analog zum `MDBookOnlineClient` kann sich der `SignalServer` bezüglich der Verbindung zum Client in genau vier Zuständen befinden:

- `SERVER_DISCONNECTED`: Es gibt keine Datenverbindung zum Client. Im Gegensatz zum `MDBookOnlineClient` bedeutet der Verlust der Verbindung auch gleichzeitig das Ende dieses Sockets, d. h. der Thread wird gestoppt und der Java Garbage Collector wird explizit gerufen³, um die Ressourcen des `SignalServer` freizugeben.
- `SERVER_CONNECTED_TO_CLIENT`: Es existiert eine Datenverbindung zum Client, aber sie ist nicht authentifiziert. In diesem Zustand ist es nicht möglich, MD*Book zu starten.
- `SERVER_AUTHENTICATED_CLIENT`: Es existiert eine authentifizierte Datenverbindung zum Client.
- `SERVER_EXECUTING_MDBOOK`: TeX2Sk/MD*Book wird gerade ausgeführt.

²Java Threads werden in Unix auf Kindprozesse abgebildet. Hier werden also Kindprozesse erzeugt.

³In der Praxis habe ich beobachtet, daß der Garbage Collector Prozess unter Linux eine so geringe Priorität hat, daß er nie gerufen wird. Dies führte bei längerer Laufzeit des Servers zu ständig steigendem Speicherbedarf.

Die Methoden, die Signale an den Client schicken, mußten threadsicher gemacht werden, denn es kümmern sich zwei Threads um die Weiterleitung der MD*Book Ausgaben (Standardausgabe und Fehlerausgabe).

Der SignalServer protokolliert alle Aufrufe von MD*Book in der Datenbanktabelle `tasks` und ordnet sie dort den Benutzern und ihren Büchern zu.

5.4 Das Package client

Das Package *client* enthält den MD*Book Online Client. Er verwendet Java 1.2 Swing Klassen, weil zum einen das Look and Feel leicht angepaßt werden kann und zum anderen in der MD*Book Konsole ohne großen Programmieraufwand verschiedenfarbiger Text dargestellt werden. Das ist wichtig, um Fehlermeldungen des MD*Book von den Standardausgaben zu trennen. Nur mit dem Java-Plugin 1.2/1.3 von Sun Microsystems können Objekte dieser Klasse (und damit MD*Book Online) in jedem Browser und auf den verbreitetsten Plattformen⁴ benutzt werden. Der Nachteil, Software zusätzlich installieren zu müssen, um MD*Book Online zu benutzen, ist nur gering, denn Buchautoren, die die interaktiven Komponenten, also XploRe Makros benutzen, sind ohnehin auf dieses Plugin angewiesen⁵.

Dem Java Applet kann die gewünschte Sprache als Parameter übergeben werden. Der Parameter wird vom `MBOComponentBuilder` ausgewertet.

Der `MDBookOnlineClient` kann sich bezüglich des Servers in genau vier Zuständen befinden:

- `CLIENT_DISCONNECTED`: Es gibt keine Datenverbindung zum Server.
- `CLIENT_CONNECTED_TO_SERVER`: Es existiert eine Datenverbindung zum Server, aber sie ist nicht authentifiziert. In diesem Zustand ist es nicht möglich, MD*Book zu starten.
- `CLIENT_AUTHENTICATED`: Es existiert eine authentifizierte Datenverbindung zum Server.
- `CLIENT_EXECUTING_MDBOOK`: TeX2Sk/MD*Book wird gerade ausgeführt.

5.4.1 Die Klassen `MBOComponentBuilder` und `MBOComponent`

Die Klasse `MBOComponentBuilder` ist eine Factory-Klasse⁶, die Formulare (Instanzen der Klasse `MBOComponent`) erzeugt, die man direkt in Dialoge einfügen kann. Die `MBOComponents` zeichnen sich gegenüber normalen AWT Components durch den einfachen Zugriff (mittels Schlüssel) auf die Formulardaten aus.

⁴Sun Solaris, Linux auf x86 Systemen, MicroSoft Windows Systeme. Macintosh Systeme können mit der aktuellen MacOS Java X Runtime Environment auch ohne Java-Plugin auf Java 1.2 APIs zugreifen.

⁵XploRe Clients nutzen ebenfalls die Java 1.2 APIs

⁶Factory-Klassen und Factory-Methoden werden benutzt, wenn das Instantiieren von Objekten mit dem `new`-Operator nicht die gewünschte Funktionalität bietet

Ein weiteres Kennzeichen des `MBOComponentBuilder` ist seine Mehrsprachigkeit. Bei seiner Konstruktion kann man (indirekt) die Sprache auswählen, in der er Formulare verfassen soll und Warnungen an den Benutzer ausgeben soll. Alle Zeichenketten, die von der Sprache abhängig sind, werden in genau einer Methode (Konstruktur) initialisiert und können so leicht an weitere Sprachen angepaßt werden. Bisher sind englische und deutsche Zeichenketten angegeben. Die Voreinstellung ist deutsch. Auf der MD*Book Online Seite kann man eine der beiden Sprachen auswählen.

5.4.2 Die Klasse `Console`

Die private Klasse `Console` implementiert die (zeit)kritischste Komponente des MD*Book Online Client. Es sind zwei Probleme zu lösen.

Zum einen muß die Konsole gleichzeitig Tastatureingaben und Signale vom MD*Book Online Server verarbeiten. Da Swing Komponenten nicht thread-sicher sind, versuchen im ungünstigsten Fall zwei Threads gleichzeitig in die Konsole zu schreiben. In diesem Fall werden Ausnahmen geworfen und es treten Darstellungsfehlern auf. Durch die Verwendung des `synchronized`-Konzepts von Java kann diese Fehlerquelle beseitigt werden.

Die andere Fehlerquelle war schwieriger zu beheben. MD*Book und vor allem `Tex2Sk` erzeugen in sehr kurzer Zeit sehr viel Ausgabertext. Dieser Text muß nun über das Internet transportiert und angezeigt werden. Es ist wichtig, den beiden Programmen ihre Texte so schnell wie möglich abzunehmen, denn die Puffergröße für Ausgabeströme ist nur gering und wenn er gefüllt ist, blockiert das Programm, da es seine Ausgabetexte nicht mehr los wird. Daher werden serverseitig moeglichst grosse Datenpakete gelesen und verschickt. Damit wird auch dem Problem begegnet, daß TCP/IP-Pakete eine Mindestgröße haben (Headerdaten), d. h. sehr kleine Pakete deutlich mehr Bandbreite verbrauchen.

Die Konsole muß diese Daten nun aber auch in dieser hohen Geschwindigkeit verarbeiten können. Die normale Herangehensweise ist zur Anzeige des Textes wäre jedes einzelne Datenpaket zu lesen, in der Konsole Text hinzuzufügen und dann die Konsole nach unten zu scrollen. Die Tatsachen, daß Java-Applets vergleichsweise langsam laufen und die Algorithmen, die den Text hinzuzufügen und die Konsole scrollen, nicht trivial sind, führen dazu, daß sich Signale stauen. Das wiederum führt zu Anzeigefehlern, zum Absturz oder Verbindungsabbruch. Ein verbesserter Algorithmus liest zuerst alle im Datenstrom vorliegenden Signale ein, kombiniert den Text aller Signale zu einer Zeichenkette und fügt ihn erst dann der Konsole hinzu. Nur wenn inzwischen immer noch keine neuen Signale anliegen, scrollt er die Konsole nach unten. Das Ergebnis dieser Herangehensweise war, das der Text erst dann nach unten gescrollt wurde, wenn MD*Book schon fast fertig war und auf diese Weise der Benutzer keine Informationen über den aktuellen Zustand des MD*Book erhielt. Daher ist als letzte Verbesserung ein Zähler eingefügt worden, dessen Auswertung dazu führt, daß in regelmäßigen Abständen die Konsole nach unten gescrollt wird, auch wenn wieder Signale im Datenstrom anliegen.

5.5 Das Package protocol

Das Package *protocol* ist ein Rahmenwerk für streambasierte Kommunikation zwischen zwei Objekten.

Es stellt die Datenstrom-Klasse `SignalSerializer` für die Kommunikation zwischen Client (`MDBookOnlineClient`) und Server (`MDBookOnlineServer`) zur Verfügung, den Datentyp `SignalType` und die Klasse `Signal` als Datencontainer.

5.5.1 Die Klasse `SignalSerializer`

`SignalSerializer` implementiert eine dem Java Output/InputStream-Konzept ähnliche Schnittstelle:

- `availableSignal()`
- `readSignal()`
- `writeSignal(Signal signal)`

Der Datenstrom transportiert ausschließlich serialisierte Instanzen der Klasse `Signal`. Auf diese Weise ist die Typsicherheit der übertragenen Daten gewährleistet. Fehler im Datenstrom werden bereits durch den zugrundeliegenden `ObjectStream` erkannt und identifiziert. Aus diesem Grund kann der `SignalSerializer` auch nicht von `Output` oder `InputStream` abgeleitet werden, denn die `byte` schreibenden und lesenden Methoden könnten zu Inkonsistenzen im Datenstrom führen.

5.5.2 Die Klassen `SignalType` und `Signal`

Alle übertragenen Signale werden anhand ihres Typs (`getType()`) identifiziert. Alle gültigen Typen sind Klassenkonstanten der Klasse `SignalType`. Die Erzeugung weiterer Typen wird durch die Privatisierung des Konstruktors verhindert (ähnlich dem Singleton-Konzept). Dabei tritt jedoch ein Problem auf.

Eine Klassenkonstante, die in einer Java Virtual Machine instantiiert wird und dann mit dem Standard-Serialisierungsmechanismus zu einer anderen Virtual Machine übertragen wird, kann mit der dortigen Klassenkonstante nicht auf Gleichheit verglichen werden. Es gibt zwei Wege, um den Vergleich trotzdem zu ermöglichen.

Die erste Möglichkeit besteht darin, eigene (De-)Serialisierungsfunktionen zu implementieren. Java stellt dazu die Schnittstelle `java.io.Serializable` zur Verfügung. Die zweite Möglichkeit redefiniert den Vergleichsalgorithmus. Um die Struktur der übertragenen Daten zukünftig erweitern zu können, ohne neue (Serialisierungs)methoden implementieren zu müssen, habe ich die zweite Möglichkeit gewählt.

Jedoch existiert in Java keine Operatorüberladung (hier: Gleichheitsoperator `==`) wie in C++, so daß die Gleichheit mit einer explizit als Vergleichsfunktion deklarierten Methode festgestellt werden muß: `equals(SignalType`

`signaltype`). Diese Methode benutzt die Funktion `getInnerType()`, die auch benutzt werden kann, um Signaltypen als Text auszugeben, z. B. in log-Dateien.

Ein Teil der Signale enthalten neben ihrem Typ weitere Daten.

Signale, die ausschließlich vom Server empfangen werden dürfen, sind:

- `OPEN_CONNECTION`: das erste Signal nach der erfolgreichen Eröffnung des Datenstroms
- `NOT_OPENED_CONNECTION`: das erste Signal nach der fehlgeschlagenen Eröffnung des Datenstroms
- `AUTHENTICATION_DATA`: überträgt Anmeldedaten zum Server, erwartet als Antwort `AUTHENTICATION_SUCCESSFULLY` oder `AUTHENTICATION_FAILED`
 - `VK_USERNAME`: der Benutzername
 - `VK_PASSWORD`: das unverschlüsselte Passwort
- `START_MDBOOK`: fordert den Server auf, TeX2Sk und anschließend MD*Book zu starten
 - `VK_BOOKNAME`: der Name des zu übersetzenden Buches
 - `VK_FORMAT`: das Zielformat des Buches
 - `VK_SRCFORMAT`: `LATEX`
- `USER_INPUT`: überträgt Tastatureingaben des Benutzers, erwartet keine Bestätigung durch den Server
 - `VK_PRESSEDKEY`: Java (Unicode) Zeichen, entspricht dem Zeichen, das der Benutzer in der MD*Book Online Konsole eingegeben hat
- `CANCEL_EXECUTING_MDBOOK`: fordert den Server auf, die Ausführung von TeX2Sk bzw. MD*Book zu beenden

Signale, die ausschließlich vom Client empfangen werden dürfen, sind:

- `AUTHENTICATION_FAILED`: informiert über den fehlgeschlagenen Vergleich der Benutzerdaten mit den Daten in der Datenbank
- `AUTHENTICATION_SUCCESSFULLY`: bestätigt den erfolgreichen Vergleich der Benutzerdaten mit den Daten in der Datenbank
 - `VK_ALLBOOKNAMES`: die Buchnamen aller Bücher des Benutzers, separiert mit Kommata
- `MDBOOK_OUTPUT_STD`:
 - `VK_TXT`: Text, den TeX2Sk und MD*Book auf dem Standardausgabestrom ausgeben
- `MDBOOK_OUTPUT_ERR`:
 - `VK_TXT`: Text, den TeX2Sk und MD*Book auf dem Fehlerausgabestrom ausgeben

Signale, die sowohl vom Client auch vom Server empfangen werden dürfen, sind:

- `NO_VALID_SIGNAL`: das Signal `NO_VALID_SIGNAL` kann, aber muß nicht, vom Datenstrom erzeugt worden sein.

Das Signal `NO_VALID_SIGNAL` wird vom Datenstrom erzeugt, wenn

1. der Datenstrom unterbrochen wird,
2. ein Signal übertragen wird, daß der Empfänger nicht auswerten kann⁷,

Das Signal `NO_VALID_SIGNAL` wird vom Client oder vom Server erzeugt, wenn

1. ein Signal gesendet wurde, das dem regulären Ablauf des Protokolls widerspricht
2. ein Fehler aufgetreten ist, der eine weitere Kommunikation zwischen Server und Client sinnlos macht

Das Signal `NO_VALID_SIGNAL` führt immer zum Abbruch der Kommunikation.

Die Signale `OPEN_CONNECTION` und `NOT_OPENED_CONNECTION` werden ausschließlich vom Datenstrom selbst erzeugt.

⁷Damit das funktioniert muß dem Datenstromobjekt mitgeteilt werden, ob es serverseitig oder clientseitig verwendet wird.

Kapitel 6

Ausblick

6.1 Kompatibilität mit Adobe's eBook Formats

Bei der Konvertierung von \LaTeX nach PDF werden bereits einige eBook-spezifischen Eigenschaften des PDF Dokuments erzeugt:

Inhaltsverzeichnis mit Verweisen wird bei MD*Book standardmäßig erzeugt.

Lesezeichen werden bei MD*Book standardmäßig für jedes Kapitel und jeden Abschnitt erzeugt.

Hyperlinks , die bereits im \LaTeX -Code mit `label` und `pageref` gesetzt wurden, werden in das PDF-Dokument übernommen.

Seitennummerierung I kann bereits im \LaTeX -Code bestimmt werden.

Geeignete Seitengröße

Geeignete Seitenränder

Geeignete Schriftart und -größe

Vorschaubilder auf einzelne Seiten

GIF Vorschaubild für das Deckblatt

JPEG Vorschaubild für die Anzeige im Acrobat eBook Reader-Ordner

Seitennummerierung II

Die drei ersten Eigenschaften sind über \LaTeX -Stiloptionen steuerbar. Es gilt herauszufinden, welche genauen Parameter im \LaTeX -Code zu setzen sind, um für das bevorzugte eBook-Lesegerät, die am besten lesbaren eBooks zu erzeugen.

Vorschaubilder sind prinzipiell möglich, aber eher als Alternative zu den Lesezeichen zu betrachten.

Die Vorschaubilder werden nicht direkt im PDF-Dokument eingebunden, sondern zusammen mit ihm auf den Adobe Content Server geladen. MD*Book muß an dieser Stelle also keine Vorarbeit leisten.

Die Seitennummering, die der Autor gewählt hat, muß zum Teil durch eine eBook-spezifische Seitennummerierung ersetzt werden. Darüberhinaus muß immer eine Seitennummerierung für ein doppelseitiges Format eingestellt werden.

Es gibt zwei Wege, um diese eBook-Eigenschaft zu kontrollieren. Adobe's Vorschlag, die Änderungen in der Seitennummerierung im PDF-Dokument von Hand vorzunehmen, ist im MD*Book-Kontext kaum geeignet, denn sobald mit MD*Book eine neue Version des Buches erzeugt wird, müssen die Änderungen wieder manuell vorgenommen werden. Die einfachste Lösung für die MD*Book Entwickler wäre, den Buchautoren die eBook-Seitennummerierungskriterien mitzuteilen und ihnen die Aufgabe zu überlassen, mittels geeigneten \LaTeX -Befehlen im PDF-Dokument eine gültige Seitennummerierung zu veranlassen. Die aufwendigere Methode für die Entwickler, aber für den Buchautor einfachere Methode, besteht darin, den \LaTeX -Quellcode zu parsen und alle Seitennummerierungsbefehle des Autors zu entfernen und durch andere zu ersetzen.

Da das, standardmäßig von MD*Book, erzeugte PDF-Dokument eher den ursprünglichen Absichten des Autors entgegenkommt, aber trotzdem ein PDF-Dokument für ein PDF basiertes eBook angeboten werden soll, sollte MD*Book (und damit auch MD*Book Online) optimalerweise um eine Option `-pdfebook` ergänzt werden, ohne die Option `-pdf` zu streichen.

6.2 Kompatibilität mit MicroSoft's eBook Formats

Es gibt mehrere Wege MD*Book zu veranlassen, MicroSoft's eBooks zu erzeugen. Grundsätzlich erlaubt die OEB Publication Structure-Spezifikation beliebige Dokumente in das eBook einzubinden, solange entsprechende fallback-Attribute angegeben sind. In der Praxis nützt das nicht viel, denn der Reader muß die gewünschten Formate interpretieren können. Also greift man am besten gleich auf OEB documents zurück. Die von MD*Book erzeugten HTML-Dateien sind beinahe schon OEB documents. Mit frei verfügbaren HTML-DOM-Parsern können auf einfache Art und Weise die letzten, nicht OEB document-konformen, Markups herausgefiltert werden. Als einzig verbleibende Aufgabe ist die OEB package-Datei zu generieren und mit MicroSoft's eBook SDK die lit-Datei zu erzeugen. Alle Informationen, die man benötigt, um diese Datei zu erzeugen, befinden sich in der .sk-Datei. Trotzdem ist die Erzeugung elektronischer Bücher für den MicroSoft eBook reader insgesamt etwas aufwendiger als für den Adobe Acrobat eBook Reader.

6.3 Weitere Aufgaben

Solange die eBook Standards noch nicht verabschiedet sind, muß die Entwicklung der Entwürfe und der dazugehörigen eBook Server-Produkte weiter ver-

folgt werden. In der OEB Publication Structure-Spezifikation wird zukünftig die Einbindung weiterer Dokumenttypen (HTML und Objects in OEB Documents) genauer beschrieben werden.

Sobald die eBook-Formate von MD*Book unterstützt werden, ist der MD*Book Online-Dokumentation eine Anleitung zum Vertrieb und Verkauf der Bücher über eBook Server-Partner und eine kurze Einführung über das Konzept der eBooks hinzuzufügen.

Alle MD*Book Online Seiten sollten in Englische übertragen werden.

Kapitel 7

Glossar

eBook *auch e-book.* Elektronisches netzbasiertes interaktives (Lehr)Buch, hier besonders als Ausgabeformat des MD*Book.

E-book Reading System - The combination of an e-book reading device with software or hardware that enforces copyrights and permissions. The system contains a voucher engine.”(EBX 2000)

JSP *Akronym für Java Server Page.* ASCII-Textdokument mit eingebettetem Java Code. Durch die Interpretation des Java Codes wird der statische Text mit dynamischen Text ergänzt.

MD*Book ist ein Werkzeug zum Erzeugen von e-books, entwickelt am Lehrstuhl für Statistik der Wirtschaftswissenschaftlichen Fakultät der Humboldt Universität zu Berlin.

MD*Book Online ist ein WWW basiertes Frontend zum MD*Book Werkzeug, kapselt die komplette Funktionalität des MD*Book.

OEB Package A file that describes an OEB publication. It identifies all other files in the publication and provides descriptive and access information about them (OEB 1999).

OEB Document An XML document which conforms to this specification (OEB 1999).

OEB Publication A collection of OEB documents and other files, typically in a variety of media types, including structured text and graphics, that constitutes a cohesive unit for publication (OEB 1999).

OEB Core Media Type A MIME media type that all reading systems must support (OEB 1999).

Reading Device The physical platform (hardware and software) on which publications are rendered. (OEB 1999).

Quantlets kleine Programme, die in der XploRe spezifischen Programmiersprache verfaßt werden

Use Case Eine Abfolge von Transaktionen in einem Dialog zwischen Benutzer und dem System. Ein Use Case kann einen Basisablauf und verschiedene Alternativabläufe haben. Im Rahmen der Analyse als ersten Schritt der (objektorientierten) Softwareentwicklung werden Use Cases identifiziert und klassifiziert.

Voucher Engine - The software or hardware that creates, imports, modifies and reads vouchers. The voucher engine enforces copyrights and transfer and usage permissions.”(EBX 2000)

Voucher - A digital object that describes an e-book’s transfer and usage permissions and copyrights. These are also called intellectual property rights. A voucher can be passed from one entity in the system to another entity. For example, a publisher can use a voucher to pass the permission to sell multiple copies of an e-book to a bookseller, or a consumer can use a voucher to pass the permission to use a copy of an e-book for a specified period of time (i. e. lend the e-book).”(EBX 2000)

XploRe Statistik Software, Client/Server-Architektur, Client kann als Java-Applets in HTML Seiten eingebunden werden

Literaturverzeichnis

- Adobe Systems Incorporated. PDF Reference, second edition, Adobe Portable Document Format 1.3 [online] 22 März 2001 [zitiert am 24 März 2001]. PDF Format. Im World Wide Web verfügbar: <http://partners.adobe.com/asn/developer/acrosdk/docs/PDFRef.pdf>. ISBN: 0201615886. [696 Seiten]
- Adobe Systems Incorporated. Adobe eBook Solutions (Adobe eBook Lösungen) [CD] März 2001. Enthält: Adobe Acrobat eBook Reader 2.0, At a Glance. Januar 2001. PDF Format. [eBook Reader2.0 AAG.pdf](#) erhältlich bei Adobe Systems Incorporated, San Jose, USA. [2 Seiten] Vollständig im Anhang.
- Adobe Systems Incorporated. Adobe eBooks Central [online] Ausgewählte Seiten: Adobe Content Server 2, How to sell eBooks (Wie man eBooks verkauft), How secure are eBooks? (Wie man eBooks schützt), Adobe eBook workflow; Adobe Acrobat eBook reader 2, Key features (Wesentliche Leistungsmerkmale), Frequently Asked Questions (Häufig gestellte Fragen) [zitiert am 30 März 2001]. Im World Wide Web verfügbar: <http://www.adobe.com/epaper/ebooks/main.html>. Vollständig im Anhang.
- Adobe Systems Incorporated. How to Create Adobe PDF Files for eBooks (Wie man Adobe PDF Dateien für eBooks erzeugt) [online] S.23, 66-76 [zitiert am 27 März 2001]. Im World Wide Web verfügbar: <http://www.adobe.com/products/acrdis/pdfs/pdf4eBooks.pdf> (im Anhang: S23, 58-76).
- Electronic Book Exchange Working Group. Electronic Book Exchange System, 0.8, Entwurf (System für den Austausch elektronischer Bücher) [online] Juli 2000 [zitiert am 24 März 2001]. PDF Format. Im World Wide Web verfügbar: <http://www.ebxwg.org/pdfs/spec.pdf>. [109 Seiten] Im Anhang S6-18, 21-22, 38-39, 51-53, 101-104.
- Gemstar eBook Group Limited. About eBook: Frequently Asked Questions (Häufig gestellte Fragen zum eBook) [online] 2000 [zitiert am 24 März 2001]. Im World Wide Web verfügbar: <http://www.ebook-gemstar.com/ebookrdrs/faq2.asp>. Vollständig im Anhang.
- ISO Technical Committee (TC) 46, Subcommittee (SC) 9, Excerpts from International Standard ISO 690-2. Information and documentation, Bibliographic references, Part 2: Electronic documents or parts thereof (Auszüge

des Internationalen Standards ISO 690-2, Information und Dokumentation, Bibliographische Referenzen, Teil 2: elektronische Dokumente und Teile davon) [online] 06 November 1997 [zitiert am 24 März 2001]. Im World Wide Web verfügbar: <http://www.nlc-bnc.ca/iso/tc46sc9/standard/690-2e.htm>.

Microsoft Corporation. Reader Content SDK 1.5 [online] Ausgewählte Themen: Understanding eBooks (eBooks verstehen lernen), Markup Guide und Microsoft Reader Layout (Markup Leitfaden und Microsoft Reader Architektur), Package File and Cover Graphics (Package Datei und Deckblatt-Grafik), Markup Overview (Markup Übersicht) [zitiert am 27 März 2001]. Microsoft CHM Format. Im World Wide Web verfügbar: readersdk.chm. Vollständig im Anhang.

Open eBook Authoring Group. Open eBook Publication Structure, 1.0 (Struktur offener eBook Publikationen) [online] 16 September 1999 [zitiert am 24 März 2001]. Im World Wide Web verfügbar: <http://www.openebook.org/OEB1.html>. [Im PDF Format 84 Seiten] Im Anhang S1-22, 39-47, 48-50.

OEBF's Digital Rights Management Strategy working group. A Framework for the Epublishing Ecology, version 0.78 (Rahmenwerk für das Verlegen elektronischer Bücher) [online] 25 September 2000 [zitiert am 24 März 2001]. PDF Format. Im World Wide Web verfügbar: [http://www.openebook.org/framework/A Framework for the Epublishing Ecology.pdf](http://www.openebook.org/framework/A%20Framework%20for%20the%20Epublishing%20Ecology.pdf). [25 Seiten] Vollständig im Anhang.

PKWARE, Inc. PKZIP Application Note, 4.0 (Anmerkung zur Verwendung von PKZIP) [online] 22 März 2001 [zitiert am 24 März 2001]. Im World Wide Web verfügbar: <http://www.pkware.com/appnote.html>.

R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee. RFC2616: Hypertext Transfer Protocol – HTTP/1.1, Draft Standard (Entwurf eines Standards) [online] Juni 1999 [zitiert am 24 März 2001]. In PostScript, PDF and ASCII Versionen im Internet verfügbar: <http://ftp.isi.edu/in-notes/>. [Im PDF Format 114 Seiten]

Smith, S.L. Standards versus guidelines for designing user interface software (Standards im Vergleich mit Leitfäden für die Gestaltung von Benutzerschnittstellen). In M. Helander (Ed.), Handbook of human-computer interaction (S.45-65). 1988, Amsterdam: North-Holland Publishing Company.

World Wide Web Consortium. Extensible Markup Language (XML), 1.0, W3C Recommendation [online] 10 Februar 1998, aktualisiert am 6 Oktober 2000 (second edition) [zitiert am 24 März 2001]. Im World Wide Web verfügbar: <http://www.w3.org/TR/2000/REC-xml-20001006>. [Im PDF Format 59 Seiten]

World Wide Web Consortium. HTML 4.01 Specification, W3C Recommendation [online] 24 Dezember 1999 [zitiert am 24 März 2001]. Im World Wide Web verfügbar: <http://www.w3.org/TR/1999/REC-html401-19991224>. [Im PostScript Format 389 Seiten]

World Wide Web Consortium. Cascading Style Sheets, level 1, W3C Recommendation [online] 17 Dezember 1996, überarbeitet am 11 Januar 1999 [zitiert am 24 März 2001]. Im World Wide Web verfügbar: <http://www.w3.org/TR/1999/REC-CSS1-19990111/>.

Kapitel 8

MD*Book Online

Dieser Abschnitt enthält die von mir entworfene und umgesetzte WWW Präsenz des MD*Book. Soweit es sich um statischen Inhalt handelt, wird die PDF-Repräsentation der \LaTeX -Quelle angegeben. Bei dynamischen (serverseitig generierten) WWW-Seiten werden beispielhafte HTML-Repräsentationen angegeben.

Kapitel 9

Kopien zitierter Quellen

Der Anhang ist so umfangreich weil fast ausschließlich Internetquellen benutzt wurden, die nicht in Bibliotheken und öffentlichen Archiven zu finden sind.

Dokumente, die nicht , sondern ausschließlich im Internet vorliegen, und wesentlich für den Inhalt der Arbeit sind, müssen im Anhang untergebracht werden, denn ihr Inhalt kann sich ändern, gerade wenn sie jetzt erst als Entwurf vorliegen. (AAAA) Die Reihenfolge der angefügten Dokumente ist durch die Reihenfolge der Dokumente in der Bibliographie bestimmt. Der Inhalt der Dokumente, deren zukünftige Verfügbarkeit gesichert ist, ist nur über die in der Bibliographie angegeben Quellen zu erreichen.