
Beispiel 6.6. DocBook Element: chapter

```
<!ELEMENT chapter (beginpage?,chapterinfo?,
(title,subtitle?,titleabbrev?),
(toc|lot|index|glossary|bibliography)*,
tocchap?,
(((calloutlist|glosslist|itemizedlist|orderedlist|segmentedlist|
simplelist|variablelist|caution|important|note|tip|warning|
literallayout|programlisting|programlistingco|screen|screenco|
screenshot|synopsis|cmdsynopsis|funcsynopsis|classsynopsis|
fieldsynopsis|constructorsynopsis|destructorsynopsis|
methodsynopsis|formalpara|para|simpara|address|blockquote|
graphic|graphicco|mediaobject|mediaobjectco|informalequation|
informalexample|informalfigure|informaltable|equation|example|
figure|table|msgset|procedure|sidebar|qandaset|productionset|
constraintdef|anchor|bridgehead|remark|highlights|abstract|
authorblurb|epigraph|indexterm|beginpage)+,
❶
(sect1*|
(refentry)*|
simplesect*|
(section)*))|
(sect1+|
(refentry)+|
simplesect+|
(section)+),
(toc|lot|index|glossary|bibliography)*>
```

- ❶ An dieser Stelle soll das neue `para`-Element eingefügt werden.

Wie aus der Definition des `chapter`-Elementes in Beispiel 6.6. *DocBook Element: chapter* abzuleiten ist, kann der Inhalt eines Kapitels relativ flexibel aufgebaut werden. Sobald ein Kapitel einen Unterabschnitt (Elemente `sect1`, `refentry`, `simplesect` und `section`) enthält, kann ein neues `para`-Element direkt vor diesem eingefügt werden, um die Validität des Dokumentes zu erhalten.

Problematischer hingegen ist die Wahl der Platzierung bei Kapiteln, die keinen Unterabschnitt enthalten. Hier muss das `para`-Element als letztes Element des Kapitels eingefügt werden, lediglich gefolgt von den optionalen Elementen `toc`, `lot`, `index`, `glossary` und `bibliography`. Diese Elemente können jedoch sowohl vor den eigentlichen inhaltlichen Elementen wie Paragraphen (`para`) existieren als auch danach, wodurch eine eindeutige Feststellung der Position zum Einfügen des neuen Elementes nicht allein anhand von Elementnamen erfolgen kann, sondern anhand des Kontextes entschieden werden muss.

Der Quelltext in Beispiel 6.7. *Valides DocBook-Dokument* stellt ein valides DocBook-Dokument dar, welches die soeben beschriebene Problematik aufzeigt.

Beispiel 6.7. Valides DocBook-Dokument

```
<book>
  <chapter>
    <beginpage pagenum="1"/>
    <title>Testkapitel</title>
    <index>
      <title>Index</title>
    </index>
    <simpara>Hier eigentlicher Inhalt</simpara>
    <beginpage pagenum="2"/>
    ❶
    <index>
      <title>Zweiter Index</title>
    </index>
  </chapter>
</book>
```

❶ An dieser Stelle soll das neue `para`-Element eingefügt werden.

Das Problem zur eindeutigen Feststellung der Position für das neue `para`-Element wurde durch zwei `select`-Anweisungen gelöst. In einem XSL-Stylesheet werden zunächst alle Elemente abgearbeitet, die nicht vom Typ `simplesect`, `refentry`, `sect1`, `section`, `index`, `toc`, `lot`, `glossary` oder `bibliography` sind oder aber vom Typ `index`, `toc`, `lot`, `glossary` oder `bibliography` sind und ein nachfolgendes Element auf gleicher Ebene besitzen, welches nicht von einem dieser fünf genannten Typen ist. Es folgt das Einfügen eines neuen `para`-Elementes, und abschließend werden alle Elemente abgearbeitet, die vom Typ `simplesect`, `refentry`, `sect1` und `section` sind sowie alle Elemente vom Typ `index`, `toc`, `lot`, `glossary` oder `bibliography`, die kein nachfolgendes Element auf gleicher Ebene besitzen, welches nicht von einem dieser fünf genannten Typen ist. Dieser Prozess wird beim Einstellen von Dokumenten in das System abgearbeitet, sodass neu eingestellte Dokumente sofort einen Link zum Einfügen von Annotationen erhalten. Aus diesem Grund befindet sich dieser Teil in dem Stylesheet `DocUpdate.xsl`, welcher für die Aufbereitung von Dokumenten zur Einstellung in das Digi-Pub-DMS zuständig ist (siehe Abschnitt 6.3.1).

Für jede einzufügende Annotation wird vom System ein neues `itemizedlist`-Element als letztes Element in das referenzierte `para`-Element eingefügt. Dieses Element stellt in DocBook ein Listenelement dar, welches den Vorteil hat, dass es im fertig erstellten PDF-Dokument bei Verschachtelungen automatisch eingerückt wird. Damit ist im Dokument leicht ersichtlich, worauf sich eine Annotation bezieht. Dies wird in der nachfolgenden Abbildung veranschaulicht, welche ein Kapitel mit zwei Annotationen darstellt, wobei sich die zweite Annotation auf die erste bezieht.

Kapitel 1. Beispiel für Annotationen

Etwas einleitender Text ...

Annotationen [*Annotation einfügen*]

- [Annotation von mp \(mp@bitset.de, 03.01.2003 19:36:07\)](#)
[*Annotation einfügen*]

Dies ist die erste Annotation.

- [Annotation von test \(lg002237@rzserv2.fhnon.de, 03.01.2003 19:39:11\)](#)
[*Annotation einfügen*]

Eine Antwort auf die erste Annotation.

- [Annotation von test \(lg002237@rzserv2.fhnon.de, 03.01.2003 19:38:07\)](#)
[*Annotation einfügen*]

Eine zweite neue Annotation.

Abbildung 6.8. Beispielannotation in PDF

Innerhalb des `itemizedlist`-Elementes muss sich gemäß der DocBook-DTD ein `listitem`-Element befinden. In dieses Element wird dann ein `para`-Element eingefügt, welches wiederum zur Referenzierung für neue Annotationen dienen kann und somit ein rekursives Verhalten ermöglicht. Neben dem vom Benutzer eingegebenen Text für die Annotation werden vom System der Login-Name sowie die Emailadresse des Benutzers, welche beide bei einer Registrierung angegeben werden müssen (siehe Abschnitt 6.2), in die Annotation eingefügt, um verifizieren zu können, durch wen diese Annotation eingefügt wurde. Gleichzeitig wird der aktuelle Zeitpunkt für die Erstellung der Annotation in diese mit eingefügt. Der eigentliche Annotationstext wird in einen `CDATA`-Block eingefügt, um zu verhindern, dass die Validität des Dokumentes verletzt wird, wenn der Text beispielsweise Zeichen wie `&` enthält. Der XML-Quelltext für die in Abbildung 6.8. *Beispielannotation in PDF* illustrierte Ausgabe vom System ist exemplarisch in Beispiel 6.8. *Annotation in Docbook* dargestellt.

Zur Erzeugung der Ausgabe im PDF-Format wurden einige Erweiterungen zu den DocBook XSL-Stylesheets vorgenommen, welche aus folgenden Teilen bestehen:

- `common_props.xsl`: Hier wurde der Parameter `digipub` eingefügt, der die URI zur Annotationsseite der Applikation enthält. Mit Hilfe dieser Angabe werden die einzelnen Links zum Einfügen von Annotationen erstellt.
- `annotation.xsl`: Diese Datei ist für die eigentliche Verarbeitung der annotationspezifischen Elemente und deren Aufbereitung zuständig.

- annotationtext.xml: Diese XML-Datei enthält feste Textpassagen für verschiedene Sprachen (im Rahmen dieser Arbeit wurden Deutsch und Englisch implementiert), welche bei der Erstellung des auszugebenden Dokumentes dynamisch anhand der festgelegten Sprache des Dokumentes von dem Stylesheet annotation.xsl eingebunden werden.

Beispiel 6.8. Annotation in Docbook

```

<chapter id="testdoc_N1001F">
  <title chapter_id="testdoc_N1001F" id="testdoc_N10021"
    parent_id="testdoc_N10001">Beispiel für Annotationen</title>
  <para>Etwas einleitender Text ...</para>
  <para id="testdoc_ANNN1001F" role="annotation.new">
    <itemizedlist role="annotation">
      <listitem>
        <para condition="03.01.2003 19:36:07" id="testdoc_ANN_1041618967765"
          role="annotation.reply">
          <author role="annotation">
            <othername>mp</othername>
            <email>mp@bitset.de</email>
          </author>
          <![CDATA[ Dies ist die erste Annotation. ]]>
          <itemizedlist role="annotation">
            <listitem>
              <para condition="03.01.2003 19:39:11" id="testdoc_ANN_1041619151890"
                role="annotation.reply">
                <author role="annotation">
                  <othername>test</othername>
                  <email>lg002237@rzserv2.fhnon.de</email>
                </author>
                <![CDATA[ Eine Antwort auf die erste Annotation. ]]>
              </para>
            </listitem>
          </itemizedlist>
        </para>
      </listitem>
    </itemizedlist>
  </para>
</chapter>

```

Kapitel 7. Zusammenfassung und Ausblick

In den vorangegangenen Kapiteln wurde die Konzeption und Implementierung des DigiPub-Systems dargestellt. Mit diesem System ist ein Publikationssystem entstanden, durch das der Vorgang der Dokumentenveröffentlichung von DocBook-Dokumenten automatisiert wurde. Von der Einstellung des Dokumentes über die Zuweisung eines Layouts bis hin zur Erstellung einer PDF-Datei wird der Verarbeitungsprozess vom DigiPub-DMS unterstützt, welches die XML-Daten für einen späteren Zugriff und eine Durchsuchbarkeit in einer XML-Datenbank ablegt. Auch die Auslieferung von Dokumenten in elektronischer Form ist über das Zahlungs- und Versandsystem im DigiPub-System automatisiert.

Neben DocBook-Dokumenten können ebenfalls binäre Dokumente durch das DigiPub-DMS verwaltet werden. Für Dokumente im DocBook-Format sind spezielle Funktionalitäten wie das Erstellen virtueller Dokumente oder das Einfügen von Annotationen realisiert. Über ein Mandantenkonzept ist die Verwaltung von Dokumenten in unterschiedlichen Bereichen bzw. Verlagen möglich. Das DigiPub-System ist online zu erreichen unter <http://digipub.fhnon.de> (Zugriff: 22.01.2003).

Als Basistechnologie zur Umsetzung des DigiPub-DMS wurden neben Java aktuelle X-Technologien eingesetzt. Die Spezialisierung auf das Datenformat XML bringt deutliche Vorteile mit sich. Durch die selbstbeschreibende Eigenschaft dieses Formats können virtuelle Dokumente geschaffen werden, in denen Strukturinformationen implizit vorhanden sind. Bei der Transformation mit einem XSL-Stylesheet werden somit Verzeichnisse und Seitenzahlen anhand des neu erstellten Dokumentes und dessen Struktur generiert, das gesamte Dokument erhält ein einheitliches Layout. Ein Vergleich zu dem Konzept der *Slicing Books Technology* wurde in Abschnitt 3.4 vorgenommen, dort wird die Struktur des Originaldokumentes beibehalten.

Auch zum Einfügen von Annotationen in vorhandene Dokumente ist die Verwendung von XML als Dokumentenformat sehr gut geeignet. Eine Annotation kann direkt in die Struktur des Dokumentes eingefügt werden und wird durch die Verwendung einer XML-Datenbank ebenfalls bei einer Suchabfrage berücksichtigt. Sogar aus virtuellen Dokumenten heraus ist eine Annotation zu einem Abschnitt möglich, welche dann in dem jeweiligen Quelldokument hinterlegt wird.

Das Konzept der dynamischen Erstellung des Präsentationsformates hat hingegen auch Nachteile. Ein Autor kann bei der Erstellung eines Dokumentes lediglich die Struktur und den Inhalt festlegen, nicht das Layout⁷². Die Erstellung von Dokumenten in XML unterscheidet sich somit stark von der herkömmlichen Erstellung mit einem *What You See Is What You Get*-Editor, wie beispielsweise Microsoft Word.

⁷² Ob dies als Vor- oder Nachteil angesehen werden kann, hängt von dem Autor ab; im DPS-Projektteam wurde dieser Punkt für vorteilhaft gehalten

Für Autoren, die nicht im Umgang mit XML vertraut sind, ist die Umstellung sicherlich kein einfacher und schnell zu meisternder Weg. Hier bedarf es guter Autorenwerkzeuge⁷³ und der Akzeptanz des Nutzers. Des Weiteren ist die dynamische Erzeugung, vor allem die der PDF-Dateien, bei dem im DigiPub-System beschrittenen Weg ein erheblicher Zeitfaktor. Auch hier ist die Akzeptanz des Nutzers gefragt, beispielsweise bei der Generierung virtueller Dokumente.

7.1. Weitere Entwicklung

Die weitere Entwicklung des DigiPub-Systems und somit auch des DigiPub-DMS hängt stark von der Implementation in der FH NON sowie der Akzeptanz der einzelnen Nutzer und Autoren ab. Bis zum Abschluss dieser Arbeit konnten noch nicht alle administrativen und rechtlichen Fragen geklärt werden, so steht zum Beispiel die Verteilung der für das System benötigten Rollen noch nicht exakt fest. Aus diesen Gründen konnte eine offizielle Testphase des Systems noch nicht gestartet werden, durch die sich die Akzeptanz der Nutzer analysieren ließe und abhängig davon Prioritäten für bestimmte Erweiterungen festgelegt werden könnten.

Unabhängig von diesen Fragen sind jedoch u.a. folgende Erweiterungen des DigiPub-Systems nahe liegend:

- Untersuchung der XML:DB-API- sowie XPath- und XQuery-Implementationen verschiedener XML-Datenbanken, um auch in diesem Bereich eine Unabhängigkeit zu erreichen.
- Realisierung eines Warenkorbsystems. Dieses müsste mit dem Zahlungssystem des Gesamtsystems abgestimmt werden.
- Einbindung externer Mechanismen zur Volltextsuche in proprietären Dateiformaten wie beispielsweise PDF.
- Im Rahmen einer Publikationsplattform sind weitere externe Erweiterungen denkbar wie zum Beispiel ein Rezensionssystem, welches Angaben außerhalb der eigentlichen Dokumente speichert.

Ein zentraler Erweiterungspunkt in Bezug auf die Implementation in der FH NON ist das Präsentationsformat für DocBook-Dokumente. Im Rahmen des DPS-Projektes wurden zwei unterschiedlich ausgerichtete Anpassungen an den DocBook XSL-Stylesheets vorgenommen. In diesem Bereich ist eine Erweiterung um speziell auf die Fachhochschule ausgerichtete Formate sinnvoll.

⁷³ Die International Data Corporation (IDC) prognostiziert eine zusammengesetzte jährliche Wachstumsrate von 41.5% im Bereich der Entwicklung von XML-Autorenwerkzeugen bis zum Jahre 2006, siehe http://www.idcresearch.com/getdoc.jhtml?containerId=pr2002_08_28_154542 (Zugriff: 20.01.2003)

7.2. Abschlussbemerkung

Durch das Schreiben dieser Arbeit nach der DocBook-DTD konnte ausgiebige Erfahrung über dieses Format gewonnen werden, die sich als sehr positiv darstellt. DocBook wird in vielen Bereichen vor allem für technische Dokumentationen eingesetzt (siehe Abschnitt 4.2), und gerade hier kann eine Erstellung von virtuellen Dokumenten sehr sinnvoll sein. So können beispielsweise Dokumentationen für einzelne Softwaremodule in dem DigiPub-DMS abgelegt und eine komplette Dokumentation der gewählten Module je nach Anwendungsfall automatisch generiert werden. Eine Untersuchung weiterer Zielgruppen für dieses System bietet eine mögliche Grundlage für eine separate Forschungsarbeit.

Anhang A. Programmbausteine

A.1. JSP-Dateien

In diesem Abschnitt sind die erstellten JSP-Dateien und deren Bedeutung im DigiPub-DMS erläutert. Zur besseren Trennung wurden die einzelnen Dateien in verschiedene Pakete unterteilt, diese befinden sich im Hauptverzeichnis für JSP-Dateien (*view*).

Tabelle A.1. JSP-Dateien im DigiPub-DMS

JSP-Datei inklusive relativem Pfad	Aufgabe im System
index.jsp	Globale Startseite zur Weiterleitung an die Applikation
admin/changeclient.jsp	Ändern von Mandantendaten
admin/clientchangeadmin.jsp	Ändern des Administrators für einen Mandanten
admin/clientregistration1.jsp	Schritt 1 bei einer Mandantenregistrierung: Angabe eines Administrators
admin/clientregistration2.jsp	Schritt 2 bei einer Mandantenregistrierung: Angabe von weiteren Mandantendaten
client/changeaccess.jsp	Ändern von Zugriffsrechten für einen Mandanten
client/changeclassification.jsp	Ändern von Kategorien für einen Mandanten
client/clientinfo.jsp	Anzeigen von Informationen über einen Mandanten
common/about.jsp	Anzeigen einer lokalisierten <i>about</i> -Seite
common/blank.jsp	Seite zur Behandlung von Fehlern im Menübereich
common/clients.jsp	Anzeigen einer Übersicht über vorhandene Mandanten im System
common/contact.jsp	Anzeigen einer lokalisierten <i>contact</i> -Seite
common/datachanged.jsp	Anzeige für Meldungen bei erfolgreicher Datenänderung
common/errorpage.jsp	Seite zur Behandlung von Fehlern im zentralen Inhaltsbereich
common/help.jsp	Anzeigen einer lokalisierten <i>help</i> -Seite
common/search.jsp	Anzeigen eines Suchformulars
common/security.jsp	Anzeigen einer lokalisierten <i>security</i> -Seite

JSP-Datei inklusive relativem Pfad	Aufgabe im System
common/start.jsp	Anzeigen der Startseite
common/terms.jsp	Anzeigen einer lokalisierten <i>terms</i> -Seite
template/digipub.jsp	Zentraler JSP-Controller, setzt einzelne Seiten zur Präsentation zusammen
template/digipubTemplate.jsp	Zentrale JSP-Schablone zum Zusammensetzen der Präsentationsseiten
template/footer.jsp	Anzeigen des Fußbereiches
template/header.jsp	Anzeigen des Kopfbereiches
template/menu_left.jsp	Anzeigen des linken Menus
template/menu_right.jsp	Anzeigen des rechten Menus
transform/annotation.jsp	Eintragen von Annotationen
transform/annotationdone.jsp	Anzeigen einer Meldung bei erfolgreicher Annotation
transform/docinfo.jsp	Anzeigen von Informationen über ein Dokument
transform/docresult.jsp	Anzeigen eines Suchergebnisses
upload/binaryupload.jsp	Formular zum Einstellen von binären Dokumenten
upload/upload.jsp	Anzeigen von Informationen über verschiedene Uploadmöglichkeiten
upload/uploadresult.jsp	Anzeigen einer Meldung bei erfolgreichem Upload
upload/xmlupload.jsp	Formular zum Einstellen von DocBook-Dokumenten
user/userchangedata.jsp	Ändern von Benutzerdaten
user/userchangepass.jsp	Ändern des Benutzerpasswortes
user/userinfo.jsp	Anzeigen von Informationen über einen Benutzer
user/userlogin.jsp	Anmelden eines Benutzers am System
user/usermailpass.jsp	Senden des Passwortes an einen Benutzer
user/userpassmailed.jsp	Anzeigen einer Meldung bei erfolgreichem Emailversand
user/userregistration.jsp	Registrierung eines Benutzers

A.2. Java-Klassen

Die einzelnen Java-Klassen sind nach ihrer Bedeutung in unterschiedliche Pakete aufgeteilt. Dieser Abschnitt gibt einen Überblick über die erstellten Klassen des DigiPub-DMS. Als Basispaket dient *de.fhnon.digipub*. Für eine vollständige Dokumentation der Klassen wird auf die JavaDoc-Dokumentation verwiesen.

A.2.1. Package action

Das Paket *action* enthält die einzelnen *Action*-Klassen des Systems. Aufgrund gegenseitiger Unabhängigkeit werden diese Klassen, genau wie die verschiedenen JSP-Dateien, anhand einer Tabelle erläutert.

Tabelle A.2. Action-Klassen im DigiPub-DMS

Klasse	Aufgabe im System
AnnotationAction	Einfügen von Annotationen
BinaryUploadAction	Upload von Binärdateien
ChangeAccessListAction	Ändern von Zugriffsrechten für einen Mandanten
ChangeClassificationListAction	Ändern von Kategorien für einen Mandanten
ClientChangeAdminAction	Ändern des Administrators für einen Mandanten
ClientRegistrationAction	Registrierung eines Mandanten
ClientRemoveAction	Löschen eines Mandanten
ClientUpdateAction	Ändern von Mandantendaten
GetDocumentAction	Auslieferung eines Dokumentes - gibt die benötigten Daten an das Versandsystem weiter
LockDocumentAction	Sperren eines Dokumentes
RemoveDocumentAction	Löschen eines Dokumentes
SearchDocumentsAction	Suche nach Dokumenten
SelectLanguageAction	Auswahl einer Sprache für die Benutzungsoberfläche
ShowClientInfoAction	Anzeigen von Mandantendaten
ShowClientsAction	Anzeigen einer Mandantenübersicht
ShowDocumentInfoAction	Anzeigen von Dokumenteninformationen
ShowPageAction	Anzeigen einer gewählten JSP-Seite
TransformVirtualAction	Erstellung eines virtuellen Dokumentes
UserChangePassAction	Ändern eines Benutzerpasswortes

Klasse	Aufgabe im System
UserLoginAction	Anmelden eines Benutzers
UserLogoutAction	Abmelden eines Benutzers
UserMailPassAction	Senden des Passwortes an einen Benutzer
UserRegistrationAction	Registrierung eines Benutzers
UserUpdateAction	Änderung von Benutzerdaten
UserVerifyAction	Überprüfung von Nutzerdaten (auf Vorhandensein von Vor- und Nachnamen)
XMLUploadAction	Upload von DocBook-Dokumenten

A.2.2. Package config

Das Paket *config* enthält lediglich eine Klasse: `DigiPubConfig`. Diese verwaltet allgemeine Konfigurationseinstellungen, die in der Datei `digipub.conf` eingetragen sind. Auch die Konfigurationsdatei ist in diesem Paket abgelegt.

A.2.3. Package controller

Das Paket *controller* enthält die Klasse `PageController`. Diese wird zur Zuweisung von verschiedenen JSP-Seiten zu einzelnen Bereichen der Präsentationsschicht genutzt (siehe Abschnitt 5.2).

A.2.4. Package form

Das Paket *form* enthält die einzelnen `ActionForm`-Klassen des Systems, welche zur Aufnahme von gesendeten Formulardaten dienen (vgl. Abschnitt 4.4). Auch diese Klassen werden anhand einer Tabelle erläutert.

Tabelle A.3. Form-Klassen im DigiPub-DMS

Klasse	Zur Haltung und Prüfung von:
AnnotationForm	Annotationsdaten
BinaryUploadForm	Daten des Uploads von Binärdateien, Oberklasse ist <code>DocumentUploadForm</code>
ClientDataForm	Mandantendaten
DocumentUploadForm	Daten eines Dokumentenuploads
UserChangePassForm	Daten zur Passwortänderung eines Benutzers
UserDataForm	Benutzerdaten

Klasse	Zur Haltung und Prüfung von:
UserLoginForm	Daten zum Anmelden eines Nutzers am System
UserMailPassForm	Daten zum Senden eines Passwortes per Email
UserVerifyForm	Überprüfung von Nutzerdaten
XMLUploadForm	Daten des Uploads von DocBook-Dateien, Oberklasse ist DocumentUploadForm

A.2.5. Package jdbc

Das Paket *jdbc* enthält die Klassen `JDBCConnectionPool` sowie `JDBCException`. `JDBCConnectionPool` dient zum Verwalten von Datenbankverbindungen zu der relationalen Datenbank, `JDBCException` als Exception für mögliche auftretende Fehler in Bezug auf diese Verbindungen.

A.2.6. Package model

Das Paket *model* enthält die einzelnen Model- sowie deren Zugriffsklassen des Systems. Aufgrund der Abhängigkeiten der verschiedenen Komponenten zueinander ist in Abbildung A.1. *Klassendiagramm: de.fhnon.digipub.model* ein Überblick über dieses Paket illustriert. Dieses Klassendiagramm wurde in Anlehnung an die Unified Modeling Language (UML 1.4) erstellt, Attribute und Methoden wurden jedoch aus Gründen der Übersichtlichkeit entfernt. Diesbezüglich wird auf die JavaDoc-Dokumentation verwiesen.

`DocumentAdministration` fundiert als zentrale Zugriffsklasse für Dokumenteninformationen. Mandanten werden durch die Klasse `Client` abgebildet, die zentrale Zugriffsklasse hierfür ist `ClientAdministration`. Zur Verwaltung von Benutzern, die durch die Klasse `User` dargestellt werden, dient `UserAdministration`.

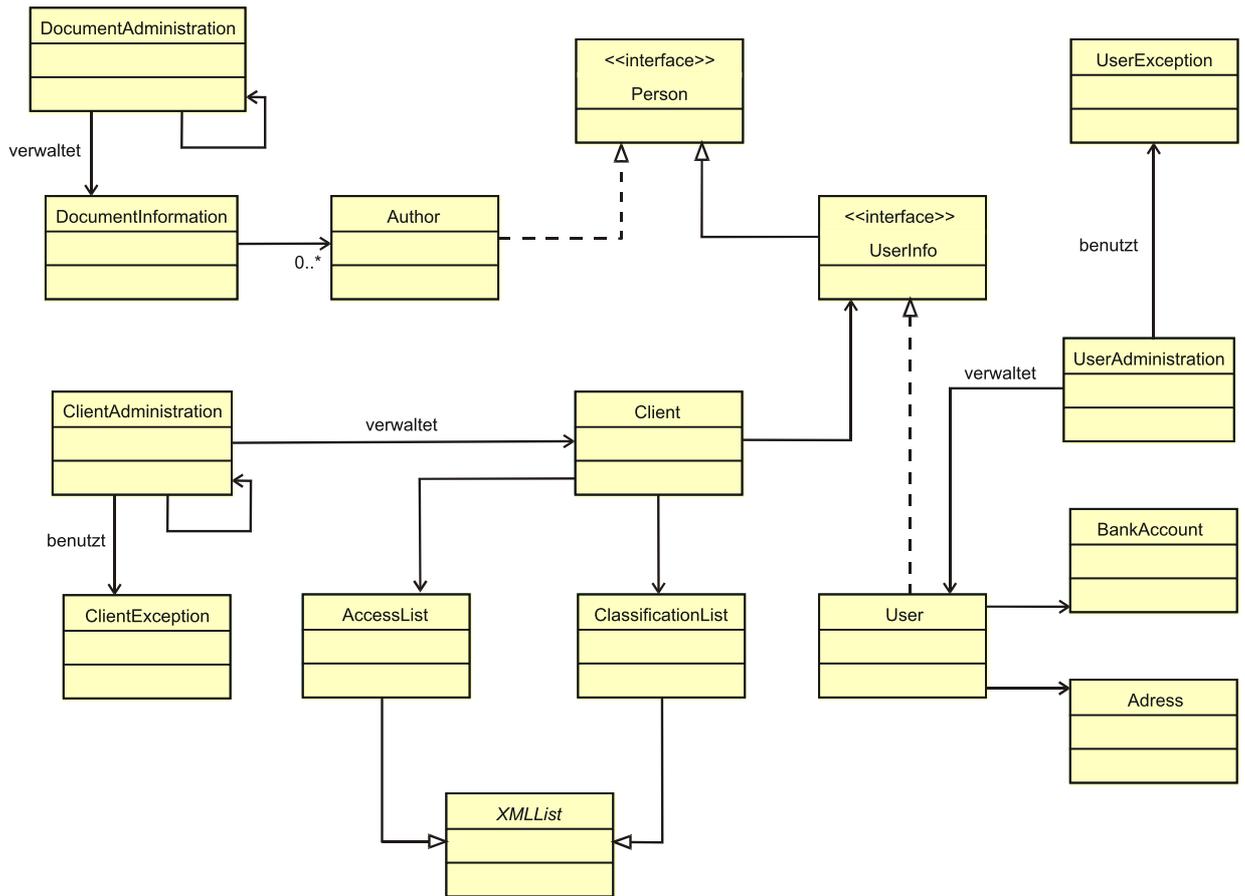


Abbildung A.1. Klassendiagramm: de.fhnon.digipub.model

A.2.7. Package resources

Das Paket *resources* enthält die verschiedenen `Properties`-Dateien zur Internationalisierung (Abschnitt 5.3).

A.2.8. Package tools

Das Paket *tools* enthält Hilfsklassen zur Verarbeitung.

Die Klasse `DigiPubException` stellt die Oberklasse für weitere `Exception`-Klassen im System dar und implementiert den Mechanismus zur Internationalisierung von Fehlermeldungen. Die Klasse `ExceptionFormat` dient dem Formatieren von Fehlermeldungen.

Die Klasse `DocFinder` wird zum Auffinden gespeicherter PDF-Dateien anhand der Dokumenten-ID genutzt.

Der Versand von Emails wird durch die Klasse `Mailer` durchgeführt, für mögliche auftretende Fehler bei diesem Versand dient die Klasse `MailException`.

`MimeMapping` repräsentiert die Zuweisung eines MIME-Typen zu einer Dateiendung. Als zentrale Verwaltungsklasse für Instanzen dieses Typs dient `MimeTypeAdministration`.

Mit der Klasse `PasswordGenerator` werden zufällige Benutzerpasswörter generiert. Die Ver- und Entschlüsselung von Passworten erfolgt durch die Klasse `PasswordEncryptor`.

`StyleAdministration` dient der Verwaltung von verschiedenen XSL-Stylesheets, die für den FOP-Prozess wählbar sind.

A.2.9. Package transform

Das Paket *transform* enthält Klassen, die zur XML/XSL bzw. FOP-Transformation benutzt werden.

Die Klasse `DigiPubDocumentBuilder` implementiert die JAXP-API und dient dem Erzeugen eines DOM-Dokumentes.

`GetBinaryDataServlet` fundiert als Schnittstelle zwischen dem FOP-Prozess und der XML-Datenbank (siehe „Dokumentenspeicherung und PDF-Erzeugung“).

XSL- bzw. FOP-Transformationen werden von der Klasse `ResultTransformer` durchgeführt, mögliche auftretende Fehler durch die Klasse `TransformationException` behandelt.

A.2.10. Package upload

Das Paket *upload* enthält Klassen für den Upload von Dateien in das System.

Die Klasse `FileUploadHandler` ermöglicht die Verarbeitung und Speicherung der Dateien, die über das Webinterface in das System eingestellt wurden.

`DocumentUpload` handhabt anschließend die Weiterverarbeitung und den Upload in die XML-Datenbank.

Für mögliche Fehler beim Uploadprozess dient die Klasse `FileUploadException`.

A.2.11. Package *xmldb*

Das Paket *xmldb* enthält Klassen für den Zugriff auf die Tamino-Datenbank über die XML:DB-API.

Die Klasse `XMLDBHandler` stellt hierbei die zentrale Zugriffsklasse dar und bietet Methoden für die verschiedenen Anweisungen an die Datenbank.

`XMLHandlerException` dient für mögliche Fehlermeldungen bei XML-Datenbankaktionen.

Mit Hilfe der Klasse `QueryBuilder` wird eine XPath-Query aus übermittelten Suchparametern erzeugt.

A.3. XSL-Stylesheets

Das DigiPub-DMS enthält drei zentrale XSL-Stylesheets, die direkt von der Anwendung eingebunden werden. `DocUpdate.xsl` wird zum Bearbeiten eines DocBook-Dokumentes genutzt, bevor dieses in der XML-Datenbank gespeichert wird. Die einzelnen Funktionalitäten sind in „Dokumentenupdate“ beschrieben. `DocSuche.xsl` wird zum Aufbereiten eines Suchergebnisses genutzt, diese Funktionalitäten sind in Abschnitt 6.3.2 erklärt. Zur Erzeugung virtueller Dokumente wird `VirtualUpdate.xsl` eingebunden. Dieses ist in Abschnitt 6.3.3 beschrieben.

Weiterhin existieren Anpassungen und Erweiterungen zu den DocBook XSL-Stylesheets. Zwingender Bestandteil dieses Systems sind dabei die unter Abschnitt 6.4 beschriebenen Stylesheets zur Darstellung von Annotationen.

Anhang B. Software und Lizenzen

B.1. Eingebundene Fremdsoftware

Als Bestandteile von DigiPub-DMS wurde folgende Fremdsoftware eingesetzt:

- Xerces-J Version 2.2.1 (siehe <http://xml.apache.org/xerces2-j/index.html>, Zugriff 19.12.2002). Lizenz: „The Apache Software License, Version 1.1“⁷⁴.
- Xalan-Java Version 2.4.1 (siehe <http://xml.apache.org/xalan-j/index.html>, Zugriff 19.12.2002). Lizenz: „The Apache Software License, Version 1.1“⁷⁴.
- FOP Version 0.20.5 RC (siehe <http://xml.apache.org/fop/index.html>, Zugriff 21.01.2003). FOP Version 0.20.5 RC beinhaltet eine Version des „Avalon - Apache Server Framework“. Benutzt wurde jedoch eine andere Version dieser Software:
 - Avalon - Apache Server Framework Version 4.1.3 (siehe <http://jakarta.apache.org/avalon/index.html>, Zugriff 19.12.2002).

Lizenz für beide Produkte: „The Apache Software License, Version 1.1“⁷⁴.

- JIMI Software Development Kit 1.0 (siehe <http://java.sun.com/products/jimi/>, Zugriff 19.12.2002). Lizenz: „Sun Microsystems, Inc. Binary Code License Agreement“.
- XML:DB-API Referenzimplementierung (siehe <http://www.xmldb.org/xapi/index.html>, Zugriff: 19.12.2002). Lizenz: „The XML:DB Initiative Software License, Version 1.0“⁷⁵.
- Struts Version 1.1-b2 (siehe <http://jakarta.apache.org/struts/index.html>, Zugriff 19.12.2002). Lizenz: „The Apache Software License, Version 1.1“⁷⁴
- XSL DocBook Stylesheet Distribution 1.60.0 (siehe <http://docbook.sourceforge.net/> Zugriff 22.01.2003). Open-Source.
- XSL Stylesheets zur Generierung von ISBN Buchland-Strichcodes der Firma Render-X (siehe <http://www.renderx.com/barcodes.html> Zugriff 18.01.2003).

⁷⁴siehe <http://xml.apache.org/LICENSE> (Zugriff 19.12.2002)

⁷⁵siehe <http://www.xmldb.org/legal.html> (Zugriff: 19.12.2002)

B.2. Produktionsumgebung

B.2.1. XML-Datenbankserver

- Windows 2000 Professional
- Tamino Version 3.1.2.1

Die Tamino-Distribution Version 3.1.2.1 beinhaltet eine Version der Software „Log4J“. Benutzt wurde jedoch eine andere Version dieser Software:

- Log4J Version 1.2.7 (siehe <http://jakarta.apache.org/log4j/docs/index.html>, Zugriff 19.12.2002). Lizenz: „The Apache Software License, Version 1.1“⁷⁶.

Für den Zugriff auf Tamino über die XML:DB-API wurde eine überarbeitete Version dieser Implementation benutzt, welche als Preview-Version von der Software-AG per Email übermittelt wurde.

B.2.2. Anwendungsserver

- Suse Linux 8.0
- Java(TM) 2 SDK, Standard Edition 1.4.1 (siehe <http://java.sun.com/j2se/1.4.1/>, Zugriff 19.12.2002). Lizenz: „Sun Microsystems, Inc. Binary Code License Agreement“.
- Webserver: Apache Tomcat Version 4.06 (siehe <http://jakarta.apache.org/tomcat/index.html>, Zugriff 19.12.2002). Lizenz: „The Apache Software License, Version 1.1“⁷⁶.
- MySQL 3.23.48 aus der Linux Suse 8.0 Distribution.

Der Zugriff auf die MySQL-Datenbank erfolgt unter Hilfenahme des folgenden JDBC-Treibers:

- MySQL Connector/J (siehe <http://www.mysql.com/products/connector-j/index.html>, Zugriff 19.12.2002). Lizenz: „GNU Lesser General Public License“⁷⁷.

⁷⁶siehe <http://xml.apache.org/LICENSE> (Zugriff 19.12.2002)

⁷⁷siehe <http://www.gnu.org/licenses/lgpl.html> (Zugriff 19.12.2002)

Literaturverzeichnis

Bücher

- [Bach2000] Mike Bach. *XSL und XPath - verständlich und praxisnah*. Addison-Wesley Verlag, München, Deutschland. September 2000. ISBN 3-8273-1661-8.
- [Brown2001] Simon Brown, Robert Burdick, Jayson Falkner, Ben Galbraith, Rod Johnson, Larry Kim, Casey Kochmer, Thor Kristmundsson, Sing Li, Dan Malks, Mark Nelson, Grant Palmer, Bob Sullivan, Geoff Taylor, John Timney, Sameer Tyagi, Geert Van Damme und Steve Wilkinson. *Professional JSP. 2nd Edition*. Wrox Press Ltd., Birmingham, England. Mai 2001. ISBN 1-861004-95-8.
- [Deutsch2001] Andrew Deutsch und David Czarnecki. *Java Internationalization*. O'Reilly & Associates, Inc., Sebastopol, USA. März 2001. ISBN 0-596-00019-7.
- [Eckstein2001] Robert Eckstein und Michel Casabianca. *XML - kurz und gut, 2nd Edition*. O'Reilly Verlag, Köln, Deutschland. November 2001. ISBN 3-89721-235-8.
- [Gamma1994] Erich Gamma, Richard Helm, Ralph Johnson und John Vlissides. *Design Patterns. Elements of Reusable Object-Oriented Software*. Addison Wesley, Boston, USA. Oktober 1994. ISBN 0-201-63361-2.
- [Goldfarb2000] Charles F. Goldfarb und Paul Prescod. *Das XML-Handbuch. Anwendungen, Produkte, Technologien*. Addison-Wesley Verlag, München, Deutschland. Oktober 2000. ISBN 3-8273-1712-6.
- [Kay2001] Michael Kay. *XSLT Programmer's Reference 2nd Edition*. Wrox Press, Birmingham, England. April 2001. ISBN 1-861005-06-7.
- [Klettke2002] Meike Klettke und Holger Meyer. *XML & Datenbanken. Konzepte, Sprachen und Systeme*. dpunkt.verlag GmbH, Heidelberg, Deutschland. Dezember 2002. ISBN 3-89864-148-1.
- [Walsh1999] Norman Walsh und Leonard Mueller. *DocBook: The Definitive Guide*. O'Reilly & Associates, Inc., Sebastopol, USA. Oktober 1999. ISBN 1-56592-580-7.

Zeitschriften

- [IS 12/01] Jost Enderle. „XML in relationalen Datenbanken“. *Informatik Spektrum 12/01*. S. 357-368. Springer-Verlag. Deutschland . Dezember 2001. ISSN 0170-6012.
- [iX 11/02] Karl Rehl. „Jakarta Struts: Hilfe für komplexe Webanwendungen“. *iX 11/02*. S. 130-134. Heise Verlag. Deutschland . November 2002. ISSN 0935-9680.

[WI 5/02] Thomas Tesch, Peter Frankhauser und Tim Weitzel. „Skalierbare Verarbeitung von XML mit Infonbyte-DB“. *Wirtschaftsinformatik 5/02*. S. 469-475. Vieweg Verlag, Deutschland . Oktober 2002. ISSN 0937-6429.

Internet

[Böndgen2000] Roland Böndgen. *Developing XML Applications with Java*.

<http://www.netobjectdays.org/pdf/00/slides/boendgen.pdf> (Zugriff: 09.01.2003).

[Bourret Online] Ronald Bourret. *XML and Databases*.

<http://www.rpbouret.com/xml/XMLAndDatabases.htm> (Zugriff: 13.01.2003).

[eWeeks2002] Dr. Timothy Dyck. *Server Databases Clash*. 25.02.2002.

<http://www.eweek.com/article2/0,3959,293,00.asp> (Zugriff: 09.01.2003).

[Govind1999 Online] Govind Seshadri. *Understanding JavaServer Pages Model 2 architecture*. Dezember 1999.

http://www.javaworld.com/javaworld/jw-12-1999/jw-12-ssj-jspmvc_p.html (Zugriff: 17.12.2002).

[J2EE] *Designing Enterprise Applications with the J2EE Platform, Second Edition (online)*.

http://java.sun.com/blueprints/guidelines/designing_enterprise_applications_2e/
(Zugriff: 17.12.2002).

[Kimber2002 Online] W. Eliot Kimber. *Using XSL Formatting Objects for Production-Quality Document Printing*. <http://www.isogen.com/papers/production-quality-xsl-fo.pdf>

(Zugriff: 05.01.2003).

[Schuster2000] Erwin Schuster und Stephan Wilhelm. *Content Management*. 18.12.2000.

<http://www.gi-ev.de/informatik/lexikon/inf-lex-content-management.shtml> (Zugriff: 30.12.2002).

[Slicing01] Marianne Valerius und Anna Simon. *Slicing Book Technology - eine neue Technik für eine neue Lehre?*.

<http://www.uni-koblenz.de/fb4/publikationen/gelbereihe/RR-8-2001/RR-8-2001.pdf>
(Zugriff: 30.12.2002).

[Slicing02] Ingo Dahn und Gerhard Schwabe. *Personalizing Textbooks with Slicing Technologies - Concept, Tools, Architecture, Collaborative Use*.

[http://dominosrv.uni-koblenz.de/iwi3/Literaturverwaltung.nsf/909998f5fa43206dc1256a6500613af2/7bd9de137707b346c1256b650049f36c/\\$FILE/CLNGL04.pdf](http://dominosrv.uni-koblenz.de/iwi3/Literaturverwaltung.nsf/909998f5fa43206dc1256a6500613af2/7bd9de137707b346c1256b650049f36c/$FILE/CLNGL04.pdf) (Zugriff: 30.12.2002).

-
- [Steffen2001] Dr. Jens Peter Steffen. *Die native Speicherung von XML mit der Tamino XML Database*. 08.05.2001.
<http://rzserv2.fh-lueneburg.de/u/tschritt/myweb/internetvideo/videos-htmlseiten/tamino080501.html> (Zugriff: 10.01.2003).
- [Walsh1999 Online] Norman Walsh und Leonard Muellner. *Docbook: The Definitive Guide (Online)*. <http://docbook.org/> (Zugriff: 15.12.2002).
- [XML 1.0] Tim Bray. Jean Paoli. C. M. Sperberg-McQueen. Eve Maler. *Extensible Markup Language (XML) 1.0 (Second Edition)*. W3C Recommendation. 06.10.2000.
<http://www.w3.org/TR/REC-xml> (Zugriff: 18.12.2002).
- [XPath 1.0] James Clark. Steve DeRose. *XML Path Language (XPath) Version 1.0*. W3C Recommendation. 16.11.1999. <http://www.w3.org/TR/xpath> (Zugriff: 18.12.2002).
- [XSL 1.0] Sharon Adler, Anders Berglund, Jeff Caruso, Stephen Deach, Tony Graham, Paul Grosso, Eduardo Gutentag, Alex Milowski, Scott Parnell, Jeremy Richman und Steve Zilles. *Extensible Stylesheet Language (XSL) Version 1.0*. W3C Recommendation. 15.10.2001. <http://www.w3.org/TR/xsl/> (Zugriff: 18.12.2002).
- [XSLT 1.0] James Clark. *XSL Transformations (XSLT) Version 1.0*. W3C Recommendation. 16.11.1999. <http://www.w3.org/TR/xslt> (Zugriff: 18.12.2002).

Glossar

API

Abkürzung für „Application Programming Interface“, eine dokumentierte Anwendungsprogrammchnittstelle.

AWÖ

Abkürzung für „IT-Anwendungen in der öffentlichen Verwaltung“. Dies ist eine Vorlesung an der FH NON, in der unter anderem das LIVE-System entstanden ist.

Siehe auch LIVE.

Binärdokumente

Mit „Binärdokumenten“ sind Dokumente gemeint, die nicht ausschließlich aus Textdaten bestehen, sondern zusätzlich oder ausschließlich binäre Informationen enthalten. Beispiele hierfür sind u.a. Microsoft Word oder das PDF-Format.

CMS

Abkürzung für „Content-Management-System“.

Softwaresystem für das Administrieren von Webinhalten mit Unterstützung des Erstellungsprozesses basierend auf der Trennung von Inhalten und Struktur.

—aus dem Glossar von <http://www.contentmanager.de/> (Zugriff: 30.12.2002)

Eine detaillierte Erläuterung von Content-Management-Systemen ist unter [Schuster2000] zu erhalten.

Siehe auch DMS.

CSS

Abkürzung für „Cascading Style Sheet“, einem Mechanismus zum Hinzufügen von Stilelementen zu Web-Dokumenten. CSS wird vom *World Wide Web Consortiums* entwickelt, zum jetzigen Zeitpunkt existiert eine Empfehlung zu CSS Level 1 und CSS Level 2. Weitere Informationen sind erhältlich auf der CSS Homepage unter <http://www.w3.org/Style/CSS/> (Zugriff: 30.12.2002).

DigiPub-DMS

Abkürzung für „DigiPub Document-Management-System“. Grundsystem des DPS-

Projektes, das Ergebnis dieser Arbeit. Das System ist erreichbar unter:
<http://digipub.fhnon.de> (Zugriff: 15.12.2002)

Siehe auch DPS, DigiPub-System.

DigiPub-System

Das DigiPub-System ist das Ergebnis des Forschungsprojektes DPS. Es besteht aus dem in diesem Dokument beschriebenen DigiPub-DMS sowie einem aufbauenden Zahlungs-, Sicherheits- und Versandsystem, welches von Stephan Wiesner erstellt wurde.

Siehe auch DPS, DigiPub-DMS.

DMS

„Document-Management-System“, ein System zur Ermöglichung von Document Management, eine mögliche Ausprägung von CMS. Document Management kann wie folgt definiert werden:

Document Management ist der Verwaltungsprozess von Dokumenten über ihre Lebensdauer. Vom Beginn an über die Erstellung, Durchsicht, Speicherung und Verbreitung bis zur Löschung.

— aus dem Glossar der Document Management Avenue, Website unter:

<http://cgi.papadakis.plus.com/>, (Zugriff 30.12.2002). Eigene Übersetzung.

Siehe auch CMS.

DocBook

„DocBook“ ist eine DTD zum Schreiben von strukturierten Dokumenten in SGML oder XML. DocBook ist besonders gut für technische Dokumentationen geeignet und wird von vielen Organisationen zum Erstellen solcher Dokumentationen genutzt (vgl. [Walsh1999] S. ix).

Docbook wird vom DocBook Technical Committee der OASIS (Organization for the Advancement of Structured Information Standards) gepflegt (siehe <http://www.oasis-open.org/committees/docbook/>, Zugriff 30.12.2002). Die offizielle Dokumentation repräsentiert [Walsh1999], welche auch online erreichbar ist unter [Walsh1999 Online].

DOM

Abkürzung für „Document Object Model“, API für einen kontextabhängigen Zugriff auf Teile des XML Baumes.

Das Document Object Model ist eine plattform- und sprachunabhängige Schnittstelle, die Programmen und Skripten erlaubt, dynamisch auf Dokumente

zuzugreifen und ihren Inhalt, Struktur oder Stil zu ändern. Das Dokument kann weiterverarbeitet und das Ergebnis dieser Verarbeitung in die zu präsentierende Seite eingebunden werden.

— <http://www.w3.org/DOM/> (Zugriff 01.01.2003), Eigene Übersetzung.

Weitere Informationen sind erhältlich auf den Internetseiten des *World Wide Web Consortiums* unter <http://www.w3.org/DOM/> (Zugriff: 31.12.2002).

DPS

Abkürzung für das Forschungs- und Entwicklungsprojekt „Digitales Publizieren im Selbstverlag“ der Fachhochschule Nordostniedersachsen mit dem Thema: *Modellieren und Implementieren eines Dokument-Management-Portals basierend auf X-Technologie*.

DSSSL

Abkürzung für „Document Style Semantics and Specification Language“. Sprache zum Beschreiben des Layouts von SGML Dokumenten. DSSSL ist spezifiziert unter ISO/IEC 10179:1996 (siehe <http://www.iso.ch/cate/d18196.html> (Zugriff: 31.12.2002)).

Siehe auch SGML.

DTD

Abkürzung für „Dokumententyp-Definition“ (*Document Type Definition*).

Eine Dokumententyp-Definition (oder DTD) ist eine Reihe von Definitionen für Elementtypen, Attribute, Entities und Notationen. Sie deklariert, welche davon innerhalb des Dokumentes und an welchen Stellen sie zulässig sind.

— [Goldfarb2000], S. 74

Für weitere Informationen wird auf die *XML 1.0* Empfehlung verwiesen. Sie ist erreichbar auf den Internetseiten des *World Wide Web Consortiums* (siehe [XML 1.0]).

Siehe auch XML, XML Schema.

FH NON

Abkürzung für die „Fachhochschule Nordostniedersachsen“.

FTP

Abkürzung für „File Transfer Protocol“. Bezeichnet einen Dienst, um Dateien zwischen Rechnern zu übertragen und gleichzeitig das Protokoll für diese Übertragung, welches auf

dem *Transmission Control Protokoll* (TCP) beruht (vgl. <http://www.hu-berlin.de/rz/comm/progr/ftp.html>). Die Spezifikation zu FTP ist zu finden auf den Internetseiten der *Internet Engineering Task Force* unter <http://www.ietf.org/rfc/rfc0959.txt> (Zugriff: 31.12.2002).

HTML

Abkürzung für „HyperText Markup Language“, auf SGML basierende Markupsprache zur Darstellung von Internetinhalten in Webbrowsern. Aktuelle Empfehlungen des *World Wide Web Consortiums* sind HTML 4.01 und dessen Neuformulierung in XML, XHTML 1.0. Die offizielle HTML Homepage ist erreichbar unter <http://www.w3.org/MarkUp/> (Zugriff: 30.12.2002).

HTTP

Abkürzung für „HyperText Transport Protocol“, zustandsloses Protokoll in der Anwendungsschicht. Eine Liste mit Spezifikationen zu HTTP ist erreichbar unter <http://www.w3.org/Protocols/Specs.html> (Zugriff: 31.12.2002).

ISBN

Abkürzung für „International Standard Book Number“, internationales Identifikationsmerkmal zur Kennzeichnung von Verlagszeugnissen. Für weitere Informationen siehe „Internationale Standard-Buchnummer Agentur für [die Bundesrepublik] Deutschland“, <http://www.german-isbn.org/> (Zugriff: 18.12.2002)

JAXP

Abkürzung für „Java API for XML Processing“, eine API von *Sun Microsystems, Inc* zur Unterstützung javaseitiger Bearbeitung von XML-Dokumenten mit Hilfe von DOM, SAX und XSLT (siehe <http://java.sun.com/xml/jaxp/index.html>, Zugriff: 31.12.2002). Die aktuelle Spezifikation für JAXP 1.2 vom 10.09.2002 ist zu finden unter <http://jcp.org/aboutJava/communityprocess/final/jsr063/index2.html> (Zugriff: 30.12.2002)

JDBC

Die „JDBC Data Access API“, eine Datenbankschnittstelle von *Sun Microsystems, Inc* um Java-Programmen einen einheitlichen Zugriff auf Datenbank-Software zu ermöglichen (siehe <http://java.sun.com/products/jdbc/>, Zugriff: 31.12.2002). Die aktuelle Spezifikation für JDBC 3.0 vom 09.05.2002 ist zu finden unter <http://jcp.org/aboutJava/communityprocess/final/jsr054/index.html> (Zugriff: 30.12.2002).

JSP

Abkürzung für „Java Server Page“, einer Weiterentwicklung der Java Servlet Technologie (Java-Programme zur Bearbeitung von HTTP-Anfragen). JSP-Dateien können HTML-Elemente enthalten, Java-Anweisungen werden durch besonders gekennzeichnete Elemente eingebettet. JSP-Dateien werden zur Laufzeit zu Servlets kompiliert (siehe <http://java.sun.com/products/jsp/>, Zugriff 31.12.2002). Die aktuelle Servlet 2.3 sowie JSP 1.2 Spezifikation vom 25.09.2001 ist erreichbar unter <http://jcp.org/aboutJava/communityprocess/final/jsr053/index.html> (Zugriff: 30.12.2002).

LIVE

Abkürzung für „Lehrinhalte interaktiv veröffentlichen“. Das *LIVE-System* ist das Ergebnis des AWÖ-Projektes aus dem Sommersemester 2002 an der FH NON und Ausgangsbasis für das DPS-Projekt. Das LIVE-System ist erreichbar unter <http://live.fhnon.de> (Zugriff: 30.12.2002).

Siehe auch AWÖ.

Mandant

„Mandant“ [lat.], der Auftraggeber (Bertelsmann Universal Lexikon Band 11). Im Rahmen des DigiPub-DMS wird unter Mandant ein Verantwortungsbereich verstanden, für den eine bestimmte Person (natürlich oder juristisch) verantwortlich ist.

MIME

Abkürzung für „Multipurpose Internet Mail Extensions“. Erweiterung des Internet Email Protokolls, um verschiedene Dateiformate als Nachrichten zu versenden. Die Spezifikation kann auf den Internetseiten der *Internet Engineering Task Force* in fünf Teilen unter RFC 2045-2049 abgerufen werden (siehe <http://www.ietf.org/rfc.html>, Zugriff: 31.12.2002).

MVC

Abkürzung für „Model-View-Controller“, ein Designmuster, welches erstmals in der Programmiersprache Smalltalk aufgetreten ist. MVC teilt eine Anwendung in drei Teile auf: *Model* beinhaltet die Anwendungslogik, *View* die Präsentationsschicht und *Controller* definiert die Zuordnung von Anwendungslogik zur Präsentationsschicht (vgl. [Gamma1994], S. 4).

PDF

Abkürzung für „Portable Document Format“, ein Dokumentenformat entwickelt von Ad-

obe Systems Incorporated.

SAX

Abkürzung für „Simple API for XML“, API zur ereignisgesteuerten Bearbeitung von XML. Weitere Informationen sind auf der offiziellen Website von SAX zu finden (siehe <http://www.saxproject.org/>, Zugriff: 31.12.2002).

SGML

Abkürzung für „Standard Generalized Markup Language“, Sprache zur Beschreibung von Dokumentenstrukturen mit Hilfe von Elementen (Markup). SGML ist spezifiziert unter ISO 8879:1986 (siehe <http://www.iso.ch/cate/d16387.html>, Zugriff: 30.12.2002).

SQL

Abkürzung für „Structured Query Language“, Abfragesprache für relationale Datenbanken, standardisiert unter ANSI-92 SQL.

Unicode

16-Bit Zeichencode zur Darstellung aller Schriftzeichen verwendeter Sprachen in einem Zeichensatz (vgl. [Deutsch2001], S. 145). Weitere Informationen sind erhältlich auf der Unicode Homepage unter <http://www.unicode.org/> (Zugriff: 08.01.2003).

URI

Abkürzung für „Uniform Resource Identifier“, eindeutige Adresse zum Identifizieren von Ressourcen im Internet, auch unter URL (Uniform Resource Locator) bekannt (vgl. <http://www.w3.org/Addressing/>, Zugriff: 02.01.2003).

virtuelle Dokumente

Als „virtuelle Dokumente“ werden Dokumente bezeichnet, die durch Auswahl von einzelnen Dokumententeilen von DigiPub-DMS neu erstellt wurden und in dieser Form nicht als eigenständiges Dokument zur Verfügung gestellt worden sind.

Whitespace

Unter „Whitespace“ werden Zeichen verstanden, deren Darstellung keine Farbe verwendet. Hierzu zählen Tabulatoren, Leerzeichen und Zeilenumbrüche.

W3C

Abkürzung für das *World Wide Web Consortium*. Die Website des W3C ist erreichbar unter <http://www.w3.org/> (Zugriff: 10.12.2002).

XML

Abkürzung für „eXtensible Markup Language“. XML ist eine Teilmenge von SGML. und genau wie diese eine textbasierte Meta-Auszeichnungssprache, die die Beschreibung, den Austausch, die Darstellung und die Manipulation von strukturierten Daten erlaubt, so daß diese von einer Vielzahl von Anwendungen genutzt werden können (vgl. http://www.wirtschaftsinformatik.de/wi_artikel.php?sid=769, Zugriff: 20.12.2002).

Die *XML 1.0* Empfehlung ist erreichbar auf den Internetseiten des *World Wide Web Consortiums* (siehe [XML 1.0]).

XML Query

„XML Query“, auf XPath basierende XML-Anfragesprache (vgl. [Klettke2002], Kapitel 10.3). Weitere Informationen sind erhältlich auf den Internetseiten des *World Wide Web Consortiums* (siehe <http://www.w3.org/XML/Query>, Zugriff: 31.12.2002).

Siehe auch XML, XPath.

XML Schema

„XML Schema“, Sprache zur Definition von XML-Dokumenten ähnlich zu DTD, jedoch selbst in XML spezifiziert und mit erweiterten Möglichkeiten. Die aktuelle Empfehlung für *XML Schema* vom 02.05.2001 ist erreichbar auf den Internetseiten des *World Wide Web Consortiums* und besteht aus drei Teilen:

- | | | | |
|---------|------------|--|----------------------|
| Part 0: | Einführung | (siehe http://www.w3.org/TR/xmlschema-0/ , | Zugriff: 31.12.2002) |
| Part 1: | Strukturen | (siehe http://www.w3.org/TR/xmlschema-1/ , | Zugriff: 31.12.2002) |
| Part 2: | Datentypen | (siehe http://www.w3.org/TR/xmlschema-2/ , | Zugriff: 31.12.2002) |

Siehe auch XML, DTD.

XPath

„XML Path Language“, Sprache zur Adressierung von Teilknoten eines XML-

Dokumentes (vgl. [Klettke2002], Kapitel 10.2). Die *XPath 1.0* Empfehlung ist erreichbar auf den Internetseiten des *World Wide Web Consortiums* (siehe [XPath 1.0]).

Siehe auch XML, XML Query.

XSL

„eXtensible Stylesheet Language“, Sprache zur Erstellung von Stylesheets für XML. Besteht aus den Teilen XSLT, XPath und XSL (Formatting Objects) (vgl. <http://www.w3.org/Style/XSL/>, Zugriff: 31.12.2002). Letzteres beschreibt ein XML Vokabular zur Spezifikation der Formatierung. Die *XSL 1.0* (Formatting Objects) Empfehlung ist erreichbar auf den Internetseiten des *World Wide Web Consortiums* (siehe [XSL 1.0]).

Siehe auch XML, XSLT.

XSLT

„XSL Transformations“, Sprache für die Transformation von XML-Dokumenten. Die *XSLT 1.0* Empfehlung ist erreichbar auf den Internetseiten des *World Wide Web Consortiums* (siehe [XSLT 1.0]).

Siehe auch XML, XPath, XSL.

X-Technologien

„X-Technologien“ umfasst sämtliche XML-basierende Technologien. Hierzu gehören u.a. XML, XSL, XSLT und XPath. Weitere Informationen sind erhältlich auf den Internetseiten des *World Wide Web Consortiums* (siehe <http://www.w3.org/>, Zugriff: 04.01.2003).

Siehe auch XML, XSL, XSLT, XML Schema, XPath, XML Query.

ZIP

Dateiformat für komprimierte Daten. Ein Programm zum Erstellen solcher Dateien unter Microsoft Windows ist beispielsweise *WinZip* (eine kostenlose Testversion ist erhältlich unter <http://www.winzip.de/>, Zugriff: 11.01.2003).

Stichwortverzeichnis

A

Annotation, 71
 Beispiel, 76
 Zope Book, 16
Autorensicherheit, 2
AWÖ, 1

B

Benutzer, 53
 Nutzerrollen, 44
Binärdokumente, 63
Books on Demand, 13

C

CMS, 9
Collection, 33, 49
Contropolis, 10

D

Datenbank
 Infonyte DB, 31
 MySQL, 34
 Tamino, 30
DMS, 9
DocBook, 28, 69, 73
 Elementbeispiele, 4
DocBookWriter, 2
DOM, 42
 PDOM, 31
DTD, 20
 Beispiel, 20
 Grundlagen, 20

F

Fehlerbehandlung, 46
FOP, 27, 61
 Performance, 62
 Systemeinstellung, 63

I

ILIAS, 11
Internationalisierung, 41
 Struts, 38

ISBN, 6, 56
 Buchland-Strichcode, 62

K

Konfiguration, 50
Konventionen, 3
 Java, 5
 JavaDoc, 5
 JSP, 6

L

LIVE, 1

M

Mandant, 51
 Nutzergruppen, 51
Metadaten, 59, 63
MVC, 40
 Struts, 35
MySQL, 34
 Datenbankdesign, 47

O

OPUS, 11

P

PDF
 Erzeugung, 61
PDOM, 31
Print on Demand, 13
Protokollierung, 47
Publikationssystem, 10
PXSLT, 31

S

SAX, 42
SGML, 18
Sicherheitssystem, 6, 70
 Einstellungen, 56
Slicing Book Technology, 14
StructuredText, 16
Struts, 35
 Architektur, 37

T

Tag-Bibliothek, 37

Tamino, 30

Datenbankdesign, 48

XML:DB, 33

V

Versandsystem, 6

Virtuelle Dokumente, 68

Slicing Books Technology, 14

X

XML, 18

Beispiel, 19

Datenzugriff, 42

Grundlagen, 18

Parser, 22

Spezifikationen, 28

XML Schema, 23

XML:DB, 32, 62

XPath, 24, 33, 65

XSL, 23, 61, 66

Konventionen, 6

PXSLT, 31

XSL Formatting Objects, 26

XSLT, 25

XSLT-Prozessor, 26

Z

Zahlungssystem, 2, 6

Schnittstelle, 70

Zope Book, 16