Otto-von-Guericke-Universität Magdeburg

Thema:

Transformierung von HTML-Daten in eXtensible Topic Maps zur Visualisierung von Informationen am Beispiel des Online-Lexikons Wikipedia

Diplomarbeit

Fakultät für Informatik
Arbeitsgruppe Wirtschaftsinformatik

Themensteller:	Prof. Dr. rer. pol. habil. Hans-Knud Arndt (FIN/ITI)
Betreuer:	Dipl.-Kfm. Henner Graubitz
vorgelegt von:	Lars Twele
Abgabetermin:	22. Oktober 2008

Inhaltsverzeichnis

Abbildungsverzeichnis

Abkürzungsverzeichnis

BOS	Bounded Object Set
DMS	Dokumenten Management System
DTD	Dokumenttypdefinition oder engl. Document Type Definition
GFDL	GNU Free Documentation License
HTML	Hypertext Markup Language
ISO	International Organization for Standardization
PSD	Public Subject Descriptor
SGML	Standard Generalized Markup Language
UN/Edifact	United Nations Electronic Data Interchange For Administration, Commerce and Transport
URI	Uniform Resource Identifier
XHTML	Extensible HyperText Markup Language
XML	Extensible Markup Language
XPath	XML Path Language
XTM	XML Topic Map

Tabellenverzeichnis

Kapitel 1

Einleitung

1.1 Motivation

Im Zeitalter von wachsenden Kapazitäten von Datenspeichern und dem fortschreitenden Digitalisieren von Daten und Informationen wird das Auffinden selbiger immer schwieriger. Zwar bieten diverse Suchmaschinen immer neue Möglichkeiten an die gewünschten Daten zu gelangen, doch stellt es sich als schwierig heraus semantisch zusammenhängende Informationen aufzufinden.

Sucht man z. B. eine Biographie von Johann Sebastian Bach und gibt in einer Internetsuchmaschine “Bach Biographie“ ein, so finden sich unter den ermittelten Informationen auch Beschreibungen über Gewässer und historische Erläuterungen zu Ortschaften die auf “-bach“ enden. Diese Ergebnisse basieren auf Volltextsuchen und erkennen keine semantischen Informationen innerhalb der gefunden Dokumente. Ein Werkzeug zur Behebung dieser Problematik stellt XML dar. Dieses nimmt die Aufgabe war, Dokumente zu strukturieren und einzelne Abschnitte durch eigene, frei definierbare Elemente hervorzuheben. Doch “für Anfragen wie ’zeige mir alle Biographien von Künstlern, die mit Johann Sebastian Bach befreundet waren”, ist auch XML nicht geeignet.”(vgl. Mück und Widhalm, 2002, S. 1-2)
Eine Lösung hierfür bietet der im Spätherbst 1999 verabschiedete ISO-Standard 13250 über Topic Maps. Der Gedanke dabei ist, dass bestehende Dokumente nicht selbst verändert werden müssen. Vielmehr wird eine exter-

ne Sicht mit zusätzlichen Meta-Daten darüber gesetzt, die Topic Map. Später wurde dieser ISO-Standard dann als XML Topic Maps in XML formuliert.

1.2 Ziel und Aufbau der Diplomarbeit

Ziel dieser Diplomarbeit ist es eine Möglichkeit zu finden, HTML-Seiten in XML Topic Maps, kurz XTM, zu überführen. Für die Realisierung soll eine Java-Anwendung erstellt werden, die auf Basis der Daten der deutschen Online-Enzyklopädie Wikipedia[1] Topic Maps im Format der XTM erstellt. Hierbei wird ein Suchbegriff übergeben, welcher über Wikipedia.de abgefragt werden soll. Die zu erstellenden XTM müssen über eine Schnittstelle einer nachfolgenden Applikation zur Verfügung gestellt werden, welche diese visuell darstellen wird. In Kapitel zwei dieser Arbeit werden Grundlagen wie XML, das Modell der Topic Maps, sowie die in XML fomulierte Topic Map Spezifikation XTM 1.0 erläutert. Zusätzlich bietet Kapitel 2 einen kurzen Vergleich zwischen dem Topic Map Modell und den XML Topic Maps, sowie einige praktische Einsatzmöglichkeiten von Topic Maps. In Kapitel drei wird das Programm selbst vorgestellt und seine Arbeitsschritte erläutert. Kapitel vier bietet abschließend eine Zusammenfassung der Arbeit, sowie einen Ausblick auf mögliche Erweiterungen.

[1] Die Hauptseite befindet sich unter: http://de.wikipedia.org

Kapitel 2

Grundlagen

2.1 Wikipedia

Wikipedia selbst beschreibt sich im deutschen Artikel als ein "Projekt zur Erstellung einer Online-Enzyklopädie in mehreren Sprachen". Das Hauptmerkmal bei diesem Projekt ist, dass jedermann als Autor auftreten und Artikel verfassen oder verändern kann. Das im Januar 2001 gegründete Wikipedia-Projekt versteht sich selbst als "freie Enzyklopädie, weil alle Inhalte unter freien Lizenzen stehen". Alle Artikeltexte sind durchgängig unter die GNU-Lizenz für freie Dokumentation (englische Originalbezeichnung GNU Free Documentation License, kurz GFDL) gestellt. Diese Lizenz gestattet die Vervielfältigung, Verbreitung und Veränderung des Werkes, auch zu kommerziellen Zwecken. Der Lizenznehmer verpflichtet sich dabei zur Einhaltung der Lizenzbedingungen. Diese Pflichten beinhalten unter anderem die Nennung des Autors oder der Autoren. Zusätzlich verpflichtet sich der Lizenznehmer dazu, abgeleitete Werke ebenfalls unter die GFDL zu stellen (Free Software Foundation, 2008, vgl.).
Um einzelne Artikel der Wikipedia untereinander zu verknüpfen, haben die Autoren die Möglichkeit, über eine einfache Syntax entsprechende HTML-Links (siehe dazu auch Abschnitt 2.2.4) einzufügen. Hierzu werden entsprechende Begriffe in doppelte eckige Klammern gesetzt, z. B. [[Beispiel]]. Die Software wandelt automatisch entsprechende Begriffe in interne Links auf die jeweiligen Artikel um. Existiert ein solcher Artikel noch nicht,

so wird der Link in der Farbe rot dargestellt. Folgt ein Benutzer diesem Link durch anlicken, so öffnet sich eine Eingabemaske, welche die sofortige Erfassung eines entsprechenden neuen Artikels erlaubt. “Diese einfache Verlinkungsmöglichkeit hat dafür gesorgt, dass die Artikel der Wikipedia wesentlich dichter miteinander vernetzt sind, als die der herkömmlichen digitalen Enzyklopädien.“[1]

2.2 XML

Mit der Extensible Markup Language (kurz XML) wurde vom World Wide Web Consortium im Jahr 1998 eine Auszeichnungssprache zur Darstellung hierarchisch strukturierter Daten in Form von Textdateien standardisiert. Sie wird oft als Nachfolger der Hypertext Markup Language (kurz HTML) bezeichnet. XML stellt eine Teilmenge der Standard Generalized Markup Language (kurz SGML) dar, welche 1986 von der International Organization for Standardization (kurz ISO) als Norm “ISO 8879“ verabschiedet wurde (vgl. Möhr und Schmidt, 1999, S. 52). “SGML ist wie XML eine Metasprache, also eine Sprache zur Festlegung von Sprachen. SGML legt normativ fest, wie Sprachen definiert sein müssen, die Informationen innerhalb von Dokumenten in einer für Maschinen verarbeitbaren Form kennzeichnen sollen”(vgl. Heinz Wittenbrink, 2003, S.58). Aufgabe von XML ist der Austausch von Informationen, zu diesem Zweck wird strikt zwischen Daten und der Verarbeitung von Daten getrennt. Es bleibt offen, welche Software für die Verarbeitung der Daten verwendet werden soll, XML ist ein reines Textformat. So ist die Möglichkeit gegeben, dass sowohl Mensch als auch Maschine den Inhalt verstehen. Dadurch ist der Datenaustausch unabhängig von technischen

[1] Informationen entnommen aus (Wikimedia Foundation Inc., 2008)

Implementierungen.[2] XML-Dokumente lassen sich anhand ihres Gebrauchs in dokumenten- und datenzentriert unterscheiden:

dokumentenzentriert: In diesem Fall dient XML vor allem dem Transport und der Speicherung von Informationen. Bei den Dokumenten handelt es sich vielfach um Informationen in Textform oder um Daten für die Verwendung in Medien, die zusammen mit Texten verarbeitet und dargestellt werden sollen. Der Inhalt ist für einen menschlichen Leser zum großen Teil auch ohne die Meta-Informationen verständlich. Die XML-Elemente werden lediglich zur Markierung bestimmter Textpassagen verwendet, das Dokument ist also wenig strukturiert.

datenzentriert: Hier soll XML hauptsächlich dafür sorgen, dass ausgetauschte Daten auch korrekt weiter verarbeitet werden können. Das Dokument folgt einem Schema, welches Entitäten eines Datenmodells beschreibt und definiert, in welcher Beziehung die Entitäten zueinander stehen. Ebenfalls ist festgelegt, welche Attribute die Entitäten haben. Im Fall der Datenzentrierung ist das Dokument stark strukturiert und daher für eine maschinelle Verarbeitung geeignet.[3]

2.2.1 Der Aufbau eines XML-Dokuments

In diesem Abschnitt soll erläutert werden, wie ein XML-Dokument aufgebaut ist und welche Konventionen erfüllt werden müssen. Gerade im Hinblick auf den Datenaustausch müssen allgemein gültige Regeln verwendet werden, innerhalb derer sich Autoren von XML-Dokumenten bewegen dürfen.

[2] Als Beispiel sei hier das "Formatproblem" zwischen diversen Textverarbeitungen genannt. Eine Entwicklung in die Richtung eines offenen Datenaustausches ist das "Open Document Format for Office Applications", welches 2006 als ISO Standard ISO/IEC 26300:2006 verabschiedet wurde. Als "Gegengewicht" hat die Microsoft Corporation ihr Spezifikation Office Open XML (OOXML) ebenfalls 2007 zur Standardisierung angemeldet. In diesem Jahr wurde OOXML ebenfalls zum Standard erhoben (ISO DIS 29500).

[3] (siehe dazu Heinz Wittenbrink, 2003, S. 23-24)

Markup: XML-Tags sind immer durch spitze Klammern gekennzeichnet. Sie umschließen den Element-Namen und eventuelle Attribute (`<element id='1'>Text</element>`). Die Groß- und Kleinschreibung ist entscheidend. Wenn in einem Starttag eines Elements Attribute vorkommen, so müssen diese in XML sowohl mit Attribut-Namen als auch einem Attribut-Wert angegeben werden. Zusätzlich müssen die Werte in Anführungszeichen stehen. Diese Konventionen sollen es erleichtern, Werkzeuge für die Erstellung und Verarbeitung von XML-Dokumenten zu programmieren. Ein Parser[4] muss nicht erst erkennen, wie z.B. Elemente oder Attribute geschrieben werden, auch die Bedeutung von spitzen Klammern zur Abgrenzung von Tags ist einfach festgelegt.

Wohlgeformtes XML: "Bei der Prüfung von XML-Dokumenten wird grundsätzlich zwischen der Prüfung der Wohlgeformtheit und der Gültigkeit unterschieden" (vgl. Vonhoegen, 2005, S. 58). Für die Wohlgeformtheit müssen die folgende Punkte erfüllt sein:

- Das Dokument besitzt genau ein "Wurzel-Element"(engl. "root-element").
- Alle Elemente mit Inhalt besitzen ein Start- und Endtag (z. B. `<element>Text1</element>`). Leere Elemente, also solche ohne Inhalt, können auch in sich geschlossen sein, wenn sie aus nur einem Tag bestehen, welches mit `/>` abschließt (z. B. `<element/>`).
- Die Start- und Endtags sind ebenentreu-paarig verschachtelt.
- Ein Element darf nicht mehrere Attribute mit demselben Namen besitzen.

[4] "Den Vorgang, bei dem der Computer einer Zeichenfolge `<abc>xyz</abc>` entnimmt, dass es sich bei 'xyz' um ein Element vom Typ 'abc' handelt, bezeichnet man als 'Parsen' oder 'Analysieren' der Zeichenkette (engl.: to parse; der englische Ausdruck bedeutet ursprünglich das Zerlegen eines Satzes in seine Bestandteile und seine grammatische Analyse" (vgl. Heinz Wittenbrink, 2003, S. 91). Ein Parser ist also ein Programm, oder eine Programmkomponente, die einem Dokument die darin enthaltenen Informationen entnimmt.

Doch das folgende Beispiel soll verdeutlichen, dass es nicht immer ausreicht, wenn ein XML-Dokument wohlgeformt ist.

```
<tierprodukte>
 <milchprodukt>Käse</milchprodukt>
 <getreidesorte>Weizen</getreidesorte>
</tierprodukte>
```

Zwar wäre der Beispielausschnitt wohlgeformt, es ist also kein syntaktischer Fehler zu finden, aber die Bedeutung ist falsch. Eine Getreidesorte kann semantisch kein Unterelement eines Tierproduktes sein.

Gültiges XML: Während sich die Prüfung auf Wohlgeformtheit gewissermaßen mit den Äußerlichkeiten beschäftigt, gibt es noch eine wesentlich strengere Prüfung für ein XML-Dokument, die Prüfung auf Gültigkeit. Diese setzte aber auch voraus, dass die Prüfung auf Wohlgeformtheit bestanden wurde. Die Prüfung auf Gültigkeit, auch Validierung genannt, stellt fest, ob das Dokument einen Verweis auf eine bestimmte Grammatik enthält und dieser folgt. Ein Element `<adresse>` kann zwar die Unterelemente `<strasse>` und `<plz>` haben, jedoch normalerweise kein `<geburtsdatum>`. Bei der Validierung erfolgt also eine Kontrolle darauf, welche Elemente ein Dokument haben darf oder muss und welche Eigenschaften wiederum diese Elemente haben dürfen oder müssen. Ein solches "Regelwerk" erläutert der folgende Abschnitt.

Dokumententyp-Definition: Um ein Regelwerk für die Gültigkeit eines XML-Dokumentes festzulegen, besteht die Möglichkeit, eine Dokumenttypdefinition (kurz DTD) zu definieren. Die Syntax von Dokumenttyp-Definitionen wird in Abschnitt 2.2.2 näher erläutert.

Physische Struktur: Ein Dokument besteht sowohl aus physischen Einheiten, als auch aus inhaltlichen oder logischen Einheiten. Ein XML-Dokument kann physisch auf einer Festplatte gespeichert sein. Eine solche physische Einheit bezeichnet der XML Standard auch als "enti-

ty“, wobei ein Dokument auch aus mehreren Entitäten bestehen kann. Die ist der Fall, wenn innerhalb eines Dokumentes (welches die erste Entität darstellt) eine Referenz auf ein anderes Dokument (eine weitere Entität) enthalten ist. Optional wird eine XML-Deklaration verwendet, um XML-Version, Zeichenkodierung und Verarbeitbarkeit ohne Dokumenttypdefinition zu spezifizieren. Eine Dokumenttypdefinition wird (ebenfalls optional) verwendet, um Entitäten sowie den erlaubten logischen Aufbau zu spezifizieren.

Logische Struktur: Der Aufbau eines XML-Dokuments entspricht einer Baumstruktur und ist damit hierarchisch. Als Baumknoten können Elemente, Verarbeitungsanweisungen (`<?`Ziel-Name Parameter `?>`, engl. "processing instruction"), Kommentare (`<!--` Kommentar-Text `-->`) oder Text, als normaler Text oder in Form eines CDATA-Abschnittes (`<![CDATA[` beliebiger Text `]]>`) auftreten. Im übrigen gelten die bereits vorgestellten Konventionen.

processing instruction: Die Verarbeitungsanweisungen richten sich, anders als die Kommentare, nicht an menschliche Leser eines XML-Dokuments. Sie sind für die Anwendungsprogramme gedacht, welche das Dokument verarbeiten sollen. Sie werden, wie bereits erwähnt, in der Form `<?`Ziel-Name Parameter `?>` angegeben. Das Ziel gibt oftmals den Namen der Anwendung an, für die diese Verarbeitungsanweisung gedacht ist. Diese Anweisung muss auch nur von der entsprechenden Anwendung verstanden werden, so dass hier keine weiteren XML-Regeln zur Anwendung kommen.

Namensräume: Namensräume dienen der eindeutigen Identifizierung von Elementen. Das Element `<beitrag>` kann in einem Kontext als Beitrag in einem Verein verwendet werden, im Kontext eines Verlages aber auch als Textbeitrag. Um hier eine Unterscheidung der `<beitrag>`-Elemente zu ermöglichen, können Namensräume deklariert werden, welchen die jeweiligen Elemente zugeordnet werden. Ein Namensraum lässt sich al-

so auch als ein Präfix zu einer Elementbezeichnung verstehen, wodurch eine eindeutige Identifizierung ermöglicht wird.

2.2.2 Dokumenttyp-Definition

"Eine DTD definiert eine bestimmte Klasse von Dokumenten, die alle vom gleichen Typ sind, indem sie verbindlich das Vokabular und die Grammatik für die Auszeichnungssprache festlegt, die bei der Erstellung des Dokuments verwendet werden soll und darf" (siehe Vonhoegen, 2005, S. 70). Über Markup-Deklarationen werden vier unterschiedliche Komponenten definiert, aus denen sich ein XML-Dokument zusammensetzen kann:

- Elementtyp-Deklaration
- Attributtyp-Deklaration
- Entitätsdeklaration
- Notationsdeklaration

Syntax der Elementtyp-Deklaration: Damit ein Element in einem XML-Dokument als gültig angesehen wird, muss in der DTD eine Typdeklaration für dieses Element aufgeführt sein. Hierfür sieht die allgemeine Syntax wie folgt aus:

```
<!ELEMENT Name Inhaltsmodell>
```

Der Name für das Element ist wie in XML üblich "fallsensitiv", es wird also zwischen Groß- und Kleinschreibung unterschieden. Zusätzlich sollte ein Elementname nur einmal in einer DTD vorkommen. "Wird bei zwei Elementtyp-Deklarationen derselbe Name verwendet, ignoriert der Prozessor die zweite Deklaration" (Vonhoegen, 2005, S. 77).

Inhaltsmodell: Welche Zeichendaten, oder Unterelemente ein Element enthalten darf, wird über das Inhaltsmodell festgelegt. Die fünf in Tabelle 2.1 vorgestellten unterschiedlichen Inhaltsmodelle sind möglich. Die Kardinalität

Inhaltsmodell	Beschreibung
`EMPTY`	Das Element hat keinen Inhalt, kann aber Attribute enthalten.
`ANY`	Das Element kann beliebige Inhalte enthalten, solange es sich um wohlgeformtes XML handelt.
`#PCDATA`	Das Element enthält nur Zeichendaten.
Gemischter Inhalt	Das Element kann Zeichendaten und Unterelemente enthalten.
Elementinhalt	Das Element enthält ausschließlich Unterelemente.

Tabelle 2.1: DTD Inhaltsmodelle

von Inhalten wird über Operatoren festgelegt. In der folgenden Tabelle 2.2 werden diese kurz vorgestellt.

Operator	Bedeutung
+	Das vorausgehende Element oder die Elementgruppe muss mindestens einmal, kann aber auch mehrfach vorkommen.
?	Das vorausgehende Element oder die Elementgruppe kann einmal vorkommen, kann aber auch fehlen.
*	Das vorausgehende Element oder die Elementgruppe kann beliebig oft vorkommen oder fehlen.
,	Trennzeichen innerhalb einer Sequenz von Elementen.
|	Trennzeichen zwischen sich ausschließenden Alternativen.
()	Bildung von Elementgruppen.

Tabelle 2.2: DTD Operatoren

Ein Element für den Anhang eines Dokumentes könnte mit den obigen Mitteln wie folgt deklariert werden:

```
<!ELEMENT anhang (#PCDATA | link | hinweis)*>
```

Die Deklaration besagt, dass das Element `anhang` entweder Zeichendaten enthält, oder ein Unterelement `link`, oder ein Unterelement `hinweis` und diese beliebig oft.

Attributlisten-Deklaration: Ebenfalls wie alle Elemente, müssen auch alle Attribute, die in einem gültigen Dokument erlaubt sein sollen, in der DTD deklariert werden. Es werden nicht alle Attribute einzeln deklariert, sondern sie werden in Attributlisten, die einem bestimmten Element zugeordnet werden, definiert.

```
<!ATTLIST Elementname
Attributname Attributtyp Vorgabewert
Attributname2 Attributtyp2 Vorgabewert2
...
>
```

Für Attributnamen gelten die gleichen Regeln wie für Elementnamen. Durch die Referenzierung der Attributliste auf das jeweilige Element, ist es unerheblich, ob die Liste aufgeteilt ist in mehrere Listen, oder an welcher Stelle der DTD die Liste platziert ist. Für die Lesbarkeit einer DTD ist es aber von Vorteil, wenn die Attributlisten unter dem dazugehörigen Element platziert werden, oder aber Elemente und Attributlisten in separaten Blöcken organisiert werden. Für den jeweiligen Attributtyp sind die folgenden zehn grundlegenden Typen möglich, welche in Tabelle 2.3 gelistet werden. Für die Vorgabedeklaration der Attributwerte stehen vier Möglichkeiten zur Auswahl, die in Tabelle 2.4 aufgelistet werden. Das folgende Beispiel soll die Verwendung von Attributwerten und -vorgabedeklarationen verdeutlichen.

```
<!ATTLIST anhang
id ID #REQUIRED
sprache NMTOKEN "DE"
typ CDATA "PDF"
>
```

Operator	Bedeutung
`CDATA`	einfache Zeichendaten, die kein Markup enthalten. Entitätsreferenzen sind aber erlaubt
`ENTITY`	Name einer in der DTD deklarierten nicht geparsten Entität.
`ENTITIES`	Durch Leerzeichen getrennte Liste von Entitäten.
`Aufzählung`	In Klammern eingeschlossene Liste von Token-Werten, von denen jeweils einer als Attributwert verwendet werden kann und muss.
`ID`	Eindeutiger XML-Name, der zur Identifizierung eines Elementes verwendet werden kann.
`IDREF`	Verweis auf die ID eines Elementes, die Werte von `ID` und `IDREF` müssen identisch sein.
`NMTOKEN`	Namenssymbol aus beliebigen Zeichen, die in XML-Namen erlaubt sind, aber ohne Leerzeichen.
`NMTOKENS`	Liste von Namens-Tokens, durch Leerzeichen getrennt.
`NOTATION`	Verweis auf eine Notation, z. B. der Name eines nicht XML-Formates, etwa eine Grafik.

Tabelle 2.3: Attributtypen

Diese Attributliste definiert für das oben deklarierte `anhang`-Element eine erforderliche ID, ein Attribut für die deutsche Sprache, sowie einen Typ mit dem Zeichenwert “PDF“.

Verwendung von Entitäten: In einer DTD können Entitäten verwendet werden, die es erlauben, innerhalb einer DTD Verweise auf interne oder externe DTD zu verwenden. Interne Entitäten werden wie folgt deklariert:

```
<!ENTITY name 'Ersetzungstext' >
```

Zum Beispiel lässt sich ein Kürzel für den Namen eines Autors verwenden, ähnlich einem Konstanten-Wert in der Programmierung.

```
<!ENTITY lt 'Lars Twele' >
```

Operator	Bedeutung
`Attributwert`	Eine vorgegebene Zeichenkette die verwendet wird, wenn kein Wert angegeben wird.
`#IMPLIED`	Es gibt keine Vorgabe und es ist auch keine erforderlich.
`#REQUIRED`	Ein Vorgabewert existiert nicht, ein Wert ist aber erforderlich. Dieser kann auch eine leere Zeichenkette sein.
`#FIXED` `Wert`	Legt fest, dass in jedem Fall die mit Wert angegebene Konstante verwendet wird.

Tabelle 2.4: Attributwerte

Die Entität `lt` wird mit dem Ersetzungstext `Lars Twele` vorbelegt. Innerhalb eines XML-Dokumentes läßt sich diese Entität nun wie folgt verwenden:

```
<name>&lt;</name>
```

Der XML-Parser wird während der Verarbeitung diese Entitätsreferenz auflösen und den vollen Namen in das Dokument einfügen. Das Element `name` sieht im verarbeiteten XML-Dokument dann wie folgt aus:

```
<name>Lars Twele</name>
```

Bei der Verwendung von umfangreichen Ersetzungstexten kann es hilfreich sein, diese in ein externes Dokument zu verlagern und dann mit Verweisen auf externe Entitäten zu arbeiten. Auf diese Weise kann ein XML-Dokument zum Beispiel aus mehreren Dokumenten bestehen. Die Syntax der Deklaration sieht wie folgt aus:

```
<!ENTITY name SYSTEM uri>
```

Es wird ein Uniform Resource Identifier[5] (kurz URI) für den Bezug auf eine externe Entität verwendet. Das externe Dokument muss so “gestaltet

[5] Ein Uniform Resource Identifier ist eine Zeichenfolge, die zur Identifizierung einer abstrakten oder physischen Ressource dient.

sein, dass die Dokumenteninstanz, in der eine Referenz auf dieses Dokument eingefügt wird, nach der Ersetzung der Referenz durch die angegebene Datei ein wohlgeformtes und im Sinne der DTD gültiges Dokument bleibt (vgl. Vonhoegen, 2005, S. 92).

Notationen: XML bietet die Möglichkeit, innerhalb eines Dokumentes auch andere Datenformate einzubinden, z. B. Grafiken, Videos und Sounds. Für die Verwendung eines solchen Formates ist eine dazugehörige Notationsdeklaration notwendig.

```
<!NOTATION name SYSTEM uri>
```

Um kenntlich zu machen, dass eine Entität nicht vom Parser verarbeitet werden soll, werden solche Entitäten mit dem Zusatz `NDATA` versehen, gefolgt vom vorher definierten Notationstyp. Ein Beispiel für eine externe Grafik könnte wie folgt aussehen:

```
<!NOTATION jpeg SYSTEM 'image/jpeg' >
<ENTITY lars SYSTEM 'lars.jpg' NDATA jpeg>
<!ELEMENT bild EMPTY>
<!ATTLIST bild quelle ENTITY #IMPLIED>
```

2.2.3 XPath

XPath steht für die XML Path Language, welche eine nicht auf XML basierende Sprache zur Adressierung von Teilen eines XML-Dokumentes darstellt. Durch die Verwendung von XPath ist es auf einfache Weise möglich, bestimmte Teile einer XML Hierarchie zu referenzieren und Werte individueller Komponenten zu ermitteln (vgl. Daconta et al., 2003, S. 119-120). XPath stellt in der aktuellen Version 2.0 vom Januar 2007 eine Empfehlung

des W3C dar (siehe Word Wide Web Consortium, 2007). “XPath arbeitet auf der Basis eines Baummodells, das die im XML-Dokument enthaltenen Informationseinheiten repräsentiert“. “Eine Instanz der XPath-Sprache wird Ausdruck genannt. Ein XPath-Ausdruck wird ausgewertet, um ein Objekt zu gewinnen, für das vier grundlegende Datentypen möglich sind.“

- Der Datentyp `node-set` liefert eine ungeordnete Knotenmenge, die auch leer sein kann.
- `string` ist eine Zeichenfolge, die auch leer sein kann, wobei Zeichen der XML-Empfehlung entsprechen.
- `number` ist immer eine Fließkommazahl.
- Der Datentyp `boolean` kann die Werte `true` und `false` annehmen.

Entitäten eines XML-Dokumentes sind selbst nicht ansprechbar, da sich XPath auf das bereits geparste Dokument bezieht, in welchem Entitäten schon aufgelöst und durch die entsprechenden Zeichenketten ersetzt wurden (vgl. Vonhoegen, 2005, S. 184). Das folgende Beispiel soll eine kurze Einführung in XPath-Ausdrücke geben:

`//anhang[@id]` Dieser Ausdruck ermittelt alle Elemente mit dem Namen `anhang`, welche ein Attribut mit dem Namen `id` besitzen. Das Ergebnis sind alle Elemente im Dokument inkl. ihrer Attribute und deren Werte.

`//[@quelle]` Ermittelt alle Attribute mit dem Namen `quelle` und die dazugehörigen Werte.

`/html/h1` Selektiert alle Elemente mit dem Namen `h1`, welche Unterelemente (engl. children) des Wurzelelemtes `html` sind.

`count(//anhang)` Verwendet die XPath-Funktion `count` und zählt alle Elemente mit dem Namen `anhang` innherhalb des XML-Dokumentes.

Eine umfangreiche Einführung in die Syntax der XPath-Ausdrücke bietet die Internetseite von *w3schools.com*.

2.2.4 XHTML

Der W3C-Standard Extensible HyperText Markup Language, kurz XHTML stellt die Neuformulierung von HTML 4 in XML dar. HTML ist eine in SGML formulierte Auszeichnungssprache. "HTML war, wie ursprünglich vorgesehen, eine Sprache für den Austausch wissenschaftlicher und anderer technischer Dokumente, die auch von Benutzern eingesetzt werden konnte, die keine Dokumentspezialisten waren. HTML löst die Probleme, die sich aus der Komplexität von SGML ergeben, indem es eine kleine Menge strukturbildender und semantischer Tags bereitstellt, die für die Auslegung relativ einfacher Dokumente geeignet sind. Multimedia-Fähigkeiten wurden später hinzugefügt" (Mintert, 2008). XHTML 1.0 enthält hierbei alle Elemente, die auch HTML 4 enthält. Waren Programme, die HTML verarbeiten noch Fehlertolerant und konnten so auch XHTML anzeigen, so unterliegt XHTML den Regeln von wohlgeformten und gültigen Dokumenten (siehe 2.2). Die folgende Tabelle stellt die grundlegenden Unterschiede zwischen HTML und XHTML dar.

	HTML	XML
Groß- und Kleinschreibung	(sowohl ` ` als auch ` `)	generelle Kleinschreibung (` `)
leere Elemente	` , <img>`	nur ` ` oder ` </br>` erlaubt
boolsche Attribute	`<input type="radio" checked>`	Attrib.-Wert als Name angeben, z.B. `<input type="radio" checked="checked"/>`

Tabelle 2.5: Unterschiede zwischen HTML und XHTML

2.3 ISO 13250 - Der Topic Map Standard

"Im Herbst 1999 wurde mit dem ISO-Standard 13250 der Arbeitsgruppe ISO JTC1/SC34/WG3 wahrscheinlich einer der wichtigsten Grundsteine zur intelligenten Informationssuche und -verarbeitung im Internet gelegt. Mit Topic Maps ist es möglich auf Wissen als semantisches Netzwerk zuzugreifen" (vgl. (Mück und Widhalm, 2002), S. 5).
Die Idee hinter Topic Maps ist vergleichbar mit Glossaren, Lexika oder Indexen. Sinn der Topic Maps ist es, externe Dokumente miteinander in Verbindung zu setzen und so die Suche und Navigation in dem gesamten Wissen zu erleichtern. Die Dokumente selbst sind also von den Topic Maps losgelöst, es findet keine Veränderung statt.

Der Topic Map Standard gliedert sich in die folgenden Unterpunkte:

- Topics
- Topic Names
- Topic Occurrences
- Public Subject Descriptor
- Associations
- Scopes
- Topic Maps
- Bounded Object Sets

Im Folgenden wird eine kurze Erläuterung der Punkte gegeben.

2.3.1 Topics

Topics sind die elementarsten Subjekte, sie können alles beschreiben. Angefangen bei Personen, Ländern, Zahlen und auch ganzen Zitaten. Was genau als Topic modelliert wird, hängt auch vom Anwendungsfall ab. So können es bei einem Rezept die Zutaten, Maß- und Mengeneinheiten sein. Im Bereich der Medizin könnten es Patientendaten sein, also Name, Geburtstag, Allergien etc. Auch müssen Topics in einem Dokument nicht direkt erwähnt sein, um dennoch darin vor zu kommen. Eine Gesetzesnovelle zum Thema "Pendlerpauschale" wird eventuell nicht den Begriff der Steuererhöhung beeinhalten, obwohl er die Semantik einer solchen Novelle treffend charakterisieren würde. Eine hierauf abzielende Volltextsuche wäre ergebnislos (vgl. Mück und Widhalm, 2001).
Betrachten wir ein einfaches Beispiel für Topics: Jeder in Abbildung 2.1 ab-

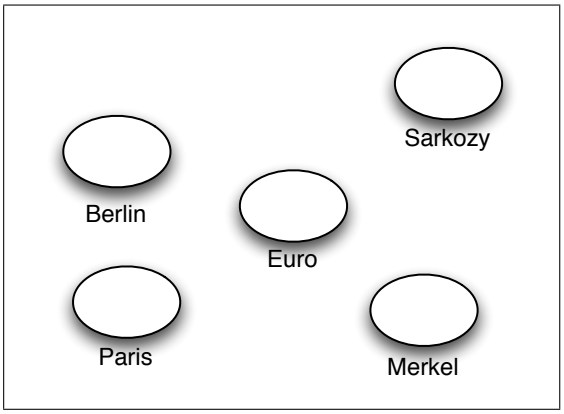

Abbildung 2.1: Topics

gebildete Kreis steht für ein Topic, die Bezeichnung befindet sich darunter. Alle Topics sind von der gleichen Art, sie unterscheiden sich nur durch Ihre Beschriftung.

In einem Zeitungsartikel könnte es zum Beispiel darum gehen, dass Kanzlerin Merkel zu einem Gespräch mit dem französischen Staatspräsidenten Sarkozy von Berlin nach Paris reist. Thema der Unterhaltung: der Euro. Nicht alle Informationen, die hier enthalten sind, werden auch bereits in Abbildung

2.1 dargestellt. Für einen Computer ist nicht zu erkennen, dass die Topics “Merkel“ und “Sarkozy“ eine Person bezeichnen, “Paris“ und “Berlin“ Städte sind und dass “Euro“ eine Währung ist. Hierfür gibt es die sogenannten Topic Types. Diese Typen werden selbst auch wieder als Topics abgebildet und dann mit den entsprechenden Topics, die sie beschreiben, über die Relation “vom Typ“ verknüpft. Dies zeigt Abbildung 2.2.

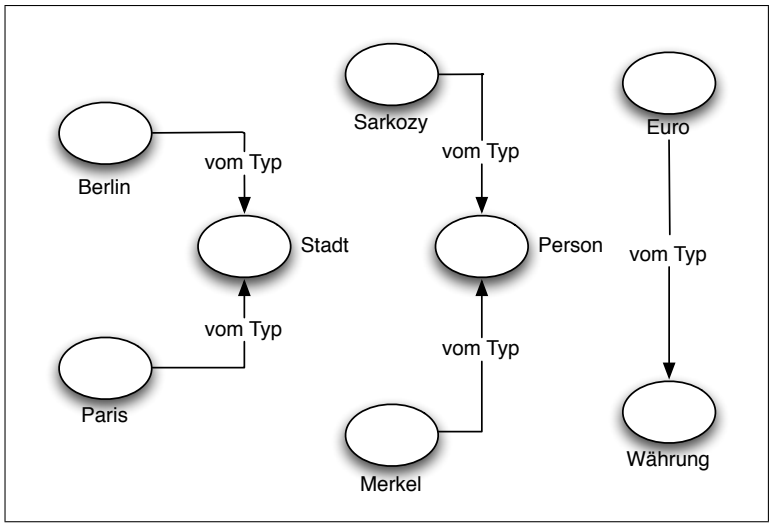

Abbildung 2.2: Topic Types

2.3.1.1 Topic Names

Namen von abzubildenden Dingen tauchen in vielen Varianten auf: volle Namen, Abkürzungen, Verweise, etc. Der Topic Map Standard versucht, alle Arten von Namen für die Topics zu ermöglichen und bietet dafür drei Varianten:

Base Name: Der Base Name eines Topics ist der “eigentliche“ Name. Im obigen Beispiel wäre das zum Beispiel “Merkel“. Jedes Topic muss mindestens einen Base Name haben, kann in unterschiedlichen

Gültigkeitsbereichen (siehe Abschnitt 2.6) aber auch mehrere Base Names haben. Die Stadt München könnte im Gültigkeitsbereich “deutsch“ den Base Name “München“, aber im Gültigkeitsbereich “englisch“ den Base Name “munich“ haben.

Display Name: Dies bezeichnet die Zeichenfolge, die auch zum Darstellen eines Topics verwendet wird. Bei Frau Merkel könnte der Display Name also “Angela Dorothea Merkel“ lauten. Ist einem Topic kein Display Name zugeordnet, so übernimmt diese Funktion der Base Name. Die Angabe des Display Name ist also optional.

Sort Name: Der Sort Name legt fest, welcher Name des Topics bei der Sortierung in Listen oder beliebigen Dokumenten herangezogen wird. Ähnlich wie beim Display Name ist der Sort Name optional und wird vom Base Name ersetzt, sollte er fehlen.

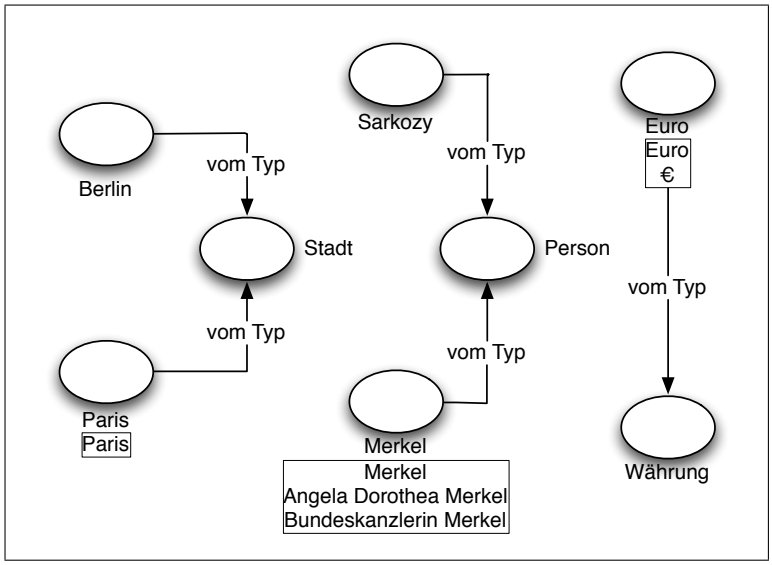

Abbildung 2.3: Topic Names

In der Abbildung 2.3 werden Namensvarianten als Rechteck unter dem jeweiligen Topic angezeigt, wobei in der Grafik nur einige Möglichkeiten verzeichnet sind.

2.3.1.2 Topic Occurrrences

Occurrences geben externe Ressourcen zu einem Topic an. Es können vielfältige Arten von Ressourcen referenziert werden: Bilder, Video- oder Audiodateien, Web- oder normale Dokumente. Im Fall von Frau Merkel könnte eine Ressource z.B. der Link auf ein Portrait-Foto von ihr sein. Zusätzlich ist ein Verweis auf eine Biographie und einen Audiomitschnitt einer ihrer Ansprachen denkbar.
Jede Occurrence kann eine bestimmte Rolle, die Occurrence Role einnehmen. Hiermit wird die Art der Ressource definiert. Jede Occurrence Role muss selbst wieder ein Topic sein, damit eine "Verknüpfung" hergestellt werden kann. Abbildung 2.4 zeigt, dass das Topic "Berlin" eine Occurrence auf einen Stadtplan im Web beinhalten kann. Diese Occurrence wiederum könnte dann die Occurrence Role "Stadtplan" haben. Auch das Topic "Stadplan" muss hierzu existieren.

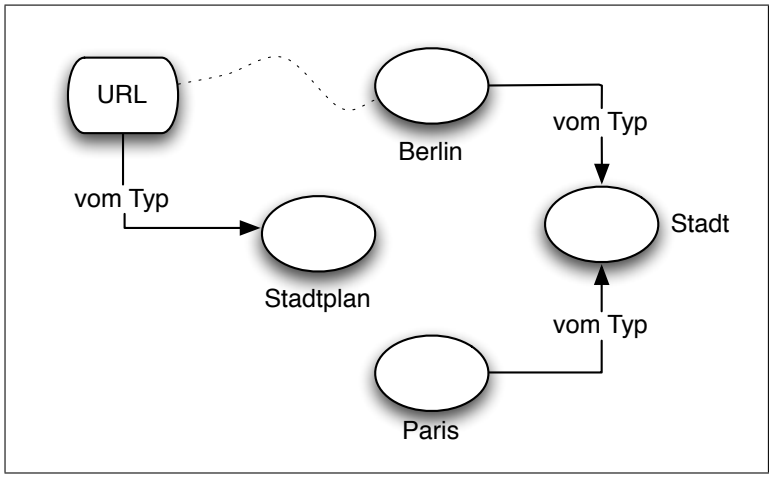

Abbildung 2.4: Occurrences

Occurrences und Ressourcen werden mit einer gestrichelte Linie dargestellt.

2.3.1.3 Public Subject Descriptor

Ein Public Subject Descriptor, kurz PSD, identifiziert ein Topic **eindeutig**. Eine Matrikelnummer zum Beispiel identifiziert einen Studenten an der Universität Magdeburg. Es besteht aber die Möglichkeit, dass diese Matrikelnummer im Zusammenhang mit einer anderen Universität auch einen gänzlich anderen Studenten identifiziert. Denkbar ist also die Identifizierung über eine weltweit gültige Sozialversicherungsnummer.
Beim zusammenführen zweier Topic Maps werden solche Topics zusammengefasst, die über den gleichen PSD verfügen. Werden zum Beispiel zwei Topic Maps vereinigt, die jeweils ein Topic "München" enthalten, so könnte folgender Fall auftreten: bei dem Topic der ersten Topic Map wird der Base Name "München" verwendet, beim Topic der zweiten Topic Map der Base Name "munich". Beide Topics verwenden aber den gleichen PSD. Durch eine Vereinigung dieser beiden Topics entsteht nun ein neues Topic mit den Eigenschaften beider Ursprungs-Topics. In diesem Fall hat das neue Topic "München" nun einmal den Base Name "München" und einmal den Base Name "munich". Der neue Gültigkeitsbereich (Scope) bildet sich aus der Vereinigung der Scopes der ursprünglichen Topics.
Das Problematik ist das Fehlen von weltweit gültigen und anerkannten PSD´s, welche zusätzlich jedem Topic Map Author bekannt sein müssten, um eine Vereinigung von unterschiedlichen Topic Maps zu ermöglichen.

2.3.2 Associations

Um Zusammenhänge von Topics abzubilden, werden Assoziationen verwendet (Associations). Beliebig viele Topics können an einer Assoziation teilhaben. Beispiele für Assoziationen:

- Berlin ist in Deutschland
- Das Brandenburger Tor ist in Berlin

- Deutschland grenzt an Frankreich

Eine Assoziation kann maximal einen Typ haben (Association Type). Dieser Typ muss wiederum ein Topic sein. Im genannten Beispiel sind das “ist in“, “grenzt an“, “wurde erbaut von“ und “errichtete“. Assoziationen können symmetrisch und auch transitiv sein. Symmetrisch wäre “grenzt an“, denn wenn Deutschland an Frankreicht grenzt, dann grenzt auch Frankreich an Deutschland. Die Assoziation “ist in“ ist transitiv, denn wenn Berlin in Deutschland ist und das Brandenburger Tor in Berlin, dann folgt daraus, dass auch das Brandenburger Tor in Deutschland ist. Jedes Topic, welches zu einer Assoziation gehört, kann eine Assoziationsrolle (Association Role) besitzen. Im obigen Beispiel könnte “Berlin“ die Rolle “Stadt“ und “Deutschland“ die Rolle “Land“ haben. Die Rollen selbst werden auch wieder als Topics deklariert. Im Beispiel in Abbildung 2.5 stellen die Rauten die Asso-

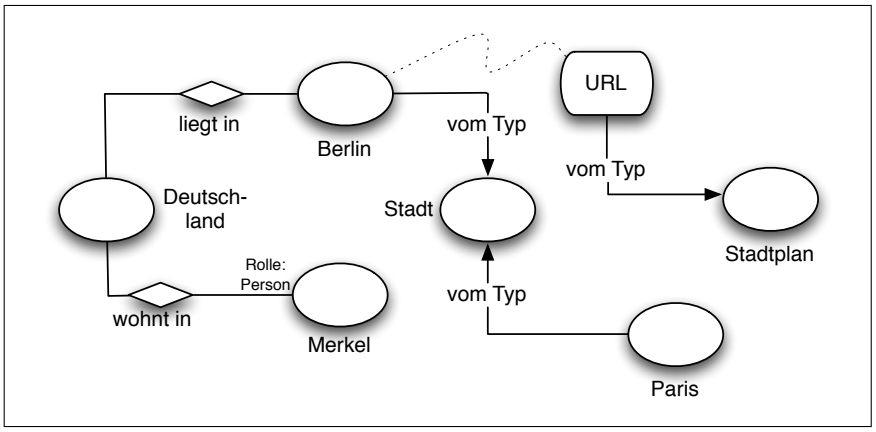

Abbildung 2.5: Associations

ziationen dar. Sie sagen aus, dass Berlin eine Stadt ist, die in Deutschland liegt. Frau Merkel wohnt in Deutschland.

2.3.3 Scopes

Bei der Erstellung von Topic Maps wird man früher oder später auf Bezeichnungen stoßen, welche nicht eindeutig sind. Also Topics, die zwar den gleichen Namen besitzen, aber unterschiedliche Bedeutungen haben(siehe Abschnitt 2.3.1.3). Wie in Abschnitt 2.3.1.1 erläutert, existieren mehrere Bezeichnungen für dasselbe Objekt. Um dieses Problem zu umgehen, bietet der Topic Map Standard die Gültigkeitsbereiche (engl. Scopes). Nehmen wir als Beispiel ein Topic "Paris". Dies könnte einmal die Hauptstadt von Frankreich und einmal den Helden aus der griechischen Mythologie bezeichnen. Paris könnte also einmal den Scope "Landeshauptstädte" und einmal den Scope "Griechenland Mythologie" besitzen. Scopes können auch für die Topics Names und Associations vergeben werde. Hat man zum Beispiel eine mehrsprachige Topic Map und eines der Topics wäre die "Sonne", so wären zwei unterschiedliche Display Names denkbar, den Display Name "Sonne" mit dem Scope "deutsch" und zusätzlich die Display Names "sol" und "sun" für die Scopes "spanisch" und "englisch". Scopes bestehen aus 1-n Themen. Der Held Paris besitzt also einen Scope aus den zwei Themen "Griechenland" und "Mythologie". Auch Themen selbst müssen wiederum als Topics deklariert werden. In Abbildung 2.6 ist zu erkennen, dass es zwei unterschiedliche Topics mit dem gleichen Namen "Paris" gibt. Durch den jeweiligen Scope (in Abbildung 2.6 als gestrichelte Linie gezeichnet) unterscheiden sie sich aber in ihrer Bedeutung und stellen so zwei unterschiedliche Topics dar. Auf der einen Seite das Topic "Paris" im Zusammenhang mit Geografie und vom Typ "Stadt", sowie einmal das Topic "Paris" im Zusammenhang mit der Mythologie und vom Typ "Person".

2.3.4 Facets

Facets bieten die Möglichkeit Topics, Associations oder wiederum anderen Facets Wertepaare zuzuordnen. Der Landeshauptstadt "Berlin" könnte man

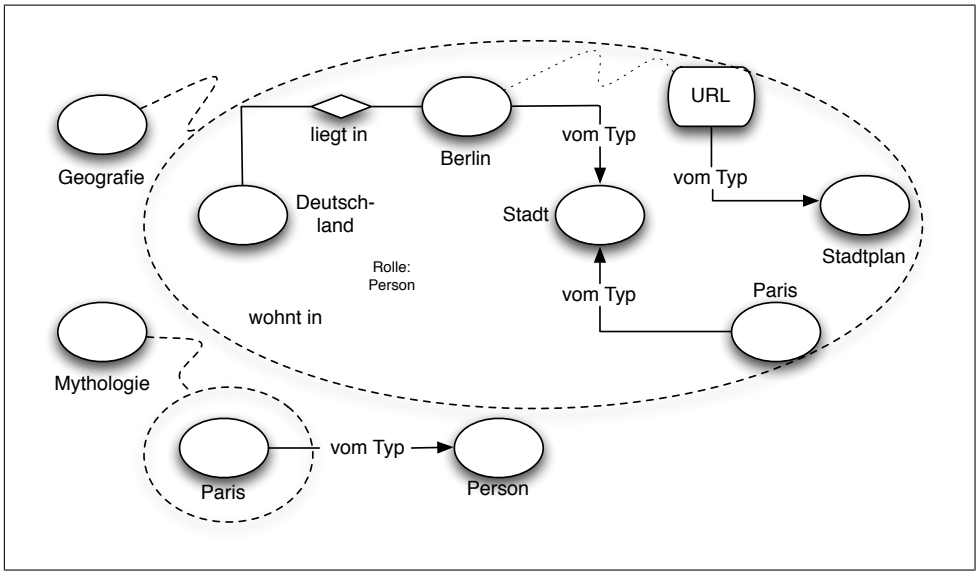

Abbildung 2.6: Scopes

zum Beispiel das Wertepaar `Einwohner 3,4 Mio.` zuordnen. Dieser Facette selbst könnte wiederum das Wertepaar `Jahr 2005` zugeordnet werden. Die Verfeinerungen wären noch endlos fortsetzbar. Es gilt hierbei allerdings zu beachten, dass keine Datentypen unterstützt werden. Die Werte sind lediglich Zeichenketten, welche entsprechend der Attribut-Regeln von SGML bzw. XML interpretiert werden.

2.3.5 Topic Maps

Die bisher vorgestellten Elemente werden in Topic Maps zusammengefasst. Topic Maps selbst können Scopes zugeordnet werden, wobei deren Themen dann für alle Elemente innerhalb der Topic Map gültig sind. Die einzelnen Konstrukte innerhalb der Topic Maps haben dabei keine Reihenfolge.
Über Topic Map Templates besteht die Möglichkeit, vorgefertigte Topic Maps mit bestimmten Topics in neue Topic Maps zu integrieren. Eine entsprechende Sammlung an Topic Maps vorausgesetzt, könnte ein Author seine Topic Map aus unterschiedlichen anderen Templates zusammenfügen. Eine neue

Topic Map über klassische Musik könnte zum Beispiel für das Topic "Johann Sebastian Bach" eine andere Topic Map über eben diese Person integrieren, wobei diese dann bereits entsprechende Occurrences, Topic Names etc. enhielte. So könnten neue Topic Maps rein aus der Vernetzung von bereits

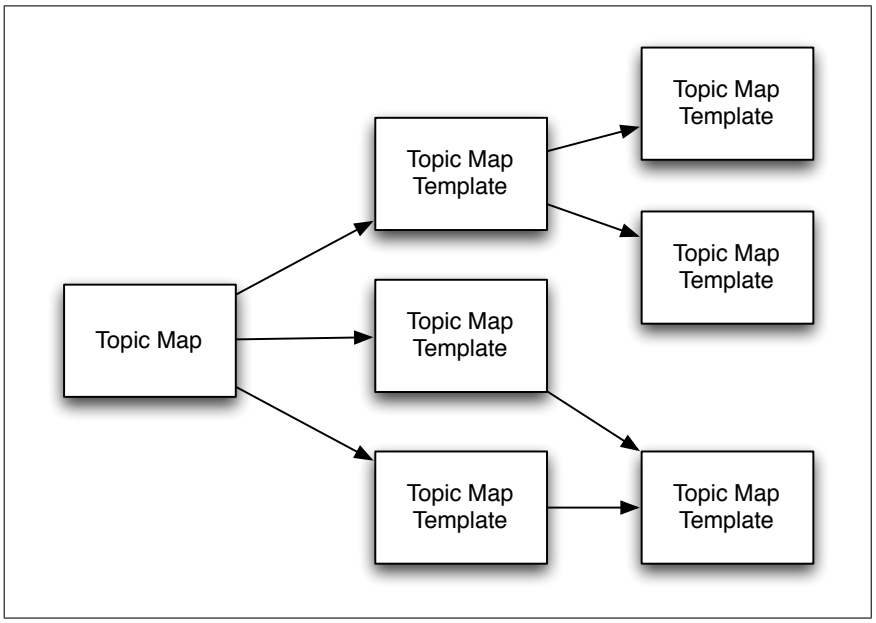

Abbildung 2.7: Topic Map Templates

existierenden Topic Maps und Templates entstehen. Dies wird in Abbildung 2.7 dargestellt.

2.3.6 Bounded Object Sets

Das Konzept, welches die Vernetzung von Topic Maps definiert, ist das Bounded Object Set, kurz BOS. Der ISO/IEC 13250 definiert dies frei übersetzt als "Satz von einem oder mehreren Dokumenten und anderen Informationsobjekten, welche der verarbeitenden Applikation bekannt sind, oder gemeinsam verarbeitet werden" (vgl. International Organization for Standardization, 2002). Eines dieser Dokumente ist dabei das sogenannte *hub document*, welches als Einstiegspunkt oder auch "Wurzel" des BOS dient. Das *hub do-*

cument muss dabei selbst nicht das Wurzeldokument sein, sondern kann in anderen Dokumenten eingebunden sein, welche dann selbst nicht Teil des BOS sind.

2.4 XML Topic Maps

Der Beginn von XML und die Akzeptanz von XML als die *Lingua Franca*[6], oder auch Verkehrssprache des Webs für die Kommunikation zwischen dokumenten- und datenbankgetriebenen Informationssystemen, erzeugte den Bedarf nach einer weniger flexibleren, weniger beängstigenden Syntax für Webapplikationen und -user. Nach dem die ISO 13250 Topic Maps Spezifikation veröffentlich worden war, begann eine unabhängige Organisation mit dem Namen TopicMaps.Org[7] damit, so schnell wie möglich eine Formulierung des ISO-Standards in XML auszuarbeiten. Bereits ein Jahr später präsentierten sie den Kern der Spezifikation auf der XML 2000 Konferenz in Washington D.C. Im März 2001 folgte dann die finale Version der XTM 1.0 (vgl. Park und Hunting, 2003, S.39-40). Im Folgenden werden die Elemente der XTM Spezifikation 1.0 in Verbindung mit der gültigen DTD vorgestellt.

2.4.1 `<topic>`, `<instanceOf>`, `<subjectIdentity>`, `<baseName>` und `<occurrence>`

Wie auch im ISO 13250 ist das zentrale Element das topic-Element. Um Topic´s eindeutig innerhalb einer Topic Map zu identifizieren, erhalten sie das Attribut “id“. Eine ID ist zwingend erforderlich. Der Wert für dieses Attribut kann innerhalb der Namenskonvention `ID` frei vergeben werden, darf aber

[6] Als Lingua Franca wird in der erweiterten Bedeutung eine Sprache bezeichnet, die für Sprecher unterschiedlicher Sprachgemeinschaften als Verkehrssprache dient.

[7] siehe dazu die Internetpräsenz unter http://www.topicmaps.org

innerhalb der Topic Map nur einmal vorkommen. Die XTM DTD[8], definiert als Elemente innerhalb eines Topics folgende untergeordneten Elemente:

`<instanceOf>` Mit diesem Element lässt sich festlegen, ob das Topic selbst wiederum einem anderen Topic zugeordnet werden kann. Zum Beispiel könnte das Topic "Merkel" eine Instanz des Topics "Person" sein. Dieses Element kann beliebig oft innerhalb eines Topics vorhanden sein.

```
<!ELEMENT instanceOf (topicRef | subjectIndicatorRef )>
<!ATTLIST instanceOf id ID #IMPLIED>
```

Die hier aufgeführten untergeordneten Elemente werden später erklärt.

`<subjectIdentity>` Um den Gegenstand eines Topics genauer zu identifizieren, so das ihn sowohl Mensch als auch Maschine verstehen, gibt es das Element subjectIdentity.

```
<!ELEMENT subjectidentity (ressourceRef?,
        (topicRef | subjectIndicatorRef)* )>
<!ATTLIST subjectidentity id ID #IMPLIED>
```

Die Identität des Gegenstands kann durch die jeweilige Ressource bestimmt oder bezeichnet werden.

`<baseName>` Dieses Element legt, analog zu den Topic Names aus Abschnitt 2.1, die Bezeichnung des Gegenstands fest. Ein Topic kann 0 bis n baseName-Elemente besitzen. Über den jeweiligen Scope wird der Geltungsbereich geregelt. Zum Beispiel kann ein und dasselbe Topic unterschiedliche Bezeichnungen in der deutschen und der englischen Sprache haben. Sollte also ein Benutzer der Topic Map diese in der englischen Variante verwenden, so erscheint für ihn der baseName mit dem Scope für die englische Sprache als Bezeichner des Topics.

```
<!ELEMENT baseName (scope?,
        baseNameString, variant* )>
<!ATTLIST baseName id ID #IMPLIED>
```

[8] die XTM DTD ist zu finden unter: http://topicmaps.org/xtm/index.html#dtd

`<occurrence>` Über das occurrence-Element werden die Ressourcen bekannt gemacht, die zu einem Topic gehören. Über das bereits bekannte "instanceOf" kann definiert werden, von welchem Typ die Ressource ist. Über einen "scope" kann ein Gültigkeitsbereich angegeben werden.

```
<!ELEMENT occurrence (instanceOf?, scope?
        (ressourceRef | ressourceData) )>
<!ATTLIST occurrence id ID #IMPLIED>
```

`<topicRef>` Das topicRef-Element ist ein leeres Element, seine Informationen sind also in den Attributen enthalten. Es zeigt auf ein anderes Element, welches ein Topic sein muss.

```
<!ELEMENT topicRef EMPTY>
<!ATTLIST topicRef id ID #IMPLIED
        xlink:type NMTOKEN #FIXED 'simple'
        xlink:href CDATA #REQUIRED>
```

`<subjectIndicatorRef>` Auch das subjectIndicatorRef-Element ist ein leeres Element. Es weist die gleichen Attribute auf, wie das vorherige "topicRef". Der Unterschied liegt darin, dass das referenzierte Element kein Topic sein muss, sondern jede Art Element gültig ist. Es kann also anstatt des "topicRef" verwendet werden, wenn es syntaktisch nicht notwendig ist ein Topic zu referenzieren.

```
<!ELEMENT subjectIndicatorRef EMPTY>
<!ATTLIST subjectindicatorRef id ID #IMPLIED
        xlink:type NMTOKEN #FIXED 'simple'
        xlink:href CDATA #REQUIRED>
```

`<baseNameString>` Der "basenameString" enthält eine reine Zeichenkette. Diese wird als Name für das Topic verwendet.

```
<!ELEMENT baseNameString (#PCDATA)>
<!ATTLIST baseNameString id ID #IMPLIED>
```

`<variant>` Über das variant-Element werden alternative Namen für das jeweilige Topic eingebunden. Varianten können rekursiv verwendet werden um Hierarchie von möglichen Varianten zu beschreiben.

```
<!ELEMENT variant (parameters,
        variantName?, variant*)>
<!ATTLIST variant id ID #IMPLIED>
```

`<variantName>` Dieses Element spezifiziert den zu verwendenden Variantennamen. Es kann entweder eine Ressource in Form einer Datei oder eine Zeichenkette sein.

```
<!ELEMENT variantName
        (resourceRef | resourceData)>
<!ATTLIST variantName id ID #IMPLIED>
```

`<parameters>` Gibt an in welchem Kontext die Namensvariante zutrifft.

```
<!ELEMENT parameters
        (topicRef | subjectIndicatorRef)+ >
<!ATTLIST parameters id ID #IMPLIED>
```

2.4.2 `<association>`, `<scope>`, `<mergeMap>` und `<topicMap>`

Die im Abschnitt 2.4.1 aufgeführten Elemente bilden das Grundgerüst um Objekte, also Topics, abzubilden. Doch wirklich Sinn ergibt eine Topic Map nur, wenn auch Beziehungen zwischen den einzelnen Gegenständen aufgezeigt werden können. Auch das Einbinden von externen Topic Maps soll ermöglicht werden und abschließend benötigt man noch den Container für alle Elemente, die Topic Map selbst. Im folgenden werden die hierfür benötigten Elemente aufgezeigt.

`<association>` Die Beziehungen zwischen Topics werden über das association-Element abgebildet. Die Topics werden hier als "member" dieser Assoziation dargestellt, wobei 1 - n Topics möglich sind. Auch hier kann es einen "scope" geben, welcher angibt, wann diese Assoziation gültig ist. Der jeweilge Typ der Beziehung kann als "instanceOf" angegeben werden.

```
<!ELEMENT association (instanceOf?, scope?,
        member+)>
<!ATTLIST association id ID #IMPLIED>
```

`<member>` Das member-Element stellt den Container für 1 - n Topics, welche eine bestimmte Rolle in einer Assoziation spielen.

```
<!ELEMENT member (roleSpec?, (topicRef |
        ressourceRef | subjectIndicatorRef)*)>
<!ATTLIST member id ID #IMPLIED>
```

`<roleSpec>` Über dieses Element wird die Rolle eines member-Elementes festgelegt, welche es in der Assoziation spielt.

```
<!ELEMENT rolseSpec (topicRef | subjectIndicatorRef)>
<!ATTLIST roleSpec id ID #IMPLIED>
```

`<scope>` Der Geltungsbereich legt wie bereits in obigen Elementen beschrieben fest, in welchem Zusammenhang ein Element Verwendung findet. Scopes werden verwendet bei Zuweisung von Namen, bei occurrence-Elementen und bei Assoziationen.

```
<!ELEMENT scope (topicRef |
        resourceRef | subjectIndicatorRef)+>
<!ATTLIST scope id ID #IMPLIED>
```

`<mergeMap>` Mit Hilfe des mergeMap-Elementes werden andere Topic Maps referenziert, welche in die aktuelle XTM integriert werden sollen. Das

mergeMap-Element ist hierbei equivalent zu den aus Programmiersprachen bekannten `INCLUDE` oder `IMPORT` über welche externer Programmcode zur Laufzeit in den ursprünglichen Programmcode implementiert und interpretiert wird.

```
<!ELEMENT mergeMap (topicRef | resourceRef |
        subjectIndicatorRef)*>
<!ATTLIST mergeMap id ID #IMPLIED
        xlink:type NMTOKEN #FIXED 'simple'
        xlink:href CDATA #REQUIRED>
```

Damit wurden nun fast alle Elemente des XTM 1.0 Standards vorgestellt, bis auf das root-Element einer jeden XTM: das Element "topicMap" selbst.

`<topicMap>` Dieses Element stellt den Container für alle anderen Elemente dar und ist somit das root-Element einer XTM.

```
<!ELEMENT topicMap (topic | association |
        mergeMap)*>
<!ATTLIST topicMap id ID #IMPLIED
        xmlns CDATA #FIXED
        'http://www.topicmaps.org/xtm/1.0'
        xmlna:xlink CDATA #FIXED
        'http://www.wr.org/1999/xlink'>
```

2.5 Unterschiede zwischen ISO-Topic Maps und XTM

In den vorhergehenden Unterpunkten wurde sowohl das Topic Map Paradigma nach dem ISO 13250 Standard als auch die XML Topic Maps nach der XTM 1.0 DTD vorgestellt. Zusammenfassend sollen hier noch einmal in

kurzer Übersicht die Unterschiede der beiden aufgelistet werden. Es lässt sich also sagen: XTM 1.0

- basiert auf XML
- definiert eine einzelne DTD anstatt einer ganzen Architektur
- eliminiert das “facet“-Element des ISO 13250, da in den XML Topic Maps eine äquivalente Funktionalität durch das “association“-Element zur Verfügung steht
- vereint “sortname“ und “dispname“ durch das Konzept des “variant“-Elementes und der dazugehörigen Unterelente
- führt den Unterschied zwischen bestimmender und bezeichnender Ressource ein
- verwendet die XLink[9] Syntax, wohin gegen ISO Topic Maps frei adressierbare Schemata verwenden
- verwendet XML-typische lange tag-Bezeichner (als Beispiel “association“ anstatt “assoc“)
- setzt auf Element Typen anstatt auf Attributzuweisungen, soweit möglich

2.6 Einsatzmöglichkeiten von Topic Maps

In diesem Abschnitt soll ein Überblick über einige praktische Einsatzmöglichkeiten von Topic Maps gegeben werden und welcher Vorteil bzw. Nutzen aus der Anwendung in unterschiedlichen Bereichen gezogen werden kann.

[9] XLink ist eine auf XML basierende Sprache zur Beschreibung und Erzeugung verschiedener Arten von Links innerhalb oder zwischen XML-Dokumenten (vgl. Geroimenko, 2004, S. 194). XLink steht für XML Linking Language und entspricht in der Version 1.0 vom Juni 2001 der Empfehlung des W3-Consortiums (siehe http://www.w3.org/TR/xlink/).

2.6.1 Informationssuche im Internet

Bereits vor 10 Jahren, also im Juli 1998, verzeichnete die DENICeG[10] 175.667 registrierte .de-Domains. Heute, Stand Juli 2008, sind bereits 12.148.809 .de-Domains registriert, Tendenz steigend. Das entspricht einer Steigerung von über 6.900%. Allerdings befand sich das Internet vor 10 Jahren noch vergleichsweise in den Anfängen, erst 1990 erfand Sir Tim Berners-Lee überhaupt das World Wide Web (siehe Berners-Lee, 1997). Allein seit Beginn dieses Jahres stieg die Anzahl der bei der DENICeG regsitrierten Domains um über 500.000 an (Stand 10.08.2008). Unter der Annahme, dass im gleichen Maß die verfügbaren Informationen entsprechend zunahmen, lässt sich erkennen, dass Hilfsmittel benötigt werden, um gewünschte Informationen in einer solche Informationsmenge aufzufinden. Diese Mittel sind Webverzeichnisse und Suchmaschinen.

Webverzeichnisse: Als Webverzeichnisse oder auch Webkataloge werden Sammlungen von Adressen von Webseiten bezeichnet, die sich im WorldWideWeb finden lassen. Diese Kataloge werden von menschlichen Redakteuren manuell gepflegt. Links werden ihrem Thema entsprechend zusammengefasst und in Kategorien eingeordnet. Diese Kategorien wiederum haben selbst Unterkategorien, so dass es einem Benutzer möglich ist, anhand dieser hierarchischen Struktur zu navigieren. Durch das redaktionelle Bearbeiten ist es möglich eine hohe Qualität des Kataloges sicherzustellen, allerdings ist der Aufwand hoch, so dass nicht das komplette Web abgebildet werden kann. Daher beschränken sich solche Webkataloge meist auf ein eingegrenztes Themengebiet. Eines der bekanntesten Beispiel ist das Portal YAHOO![11]. Die Einsatzmöglichkeit von Topic Maps innerhalb solcher Kataloge sieht der Autor als vom Aufwand her überschaubar an. Da hier bereits menschliche Redakteure die Links auswerten, lässt sich während dieses Prozesses bereits ein entsprechendes Topic anlegen. Diese Topics könnten über Assoziationen

[10] (DENIC Domain Verwaltungs und Betriebsgesellschaft eG, 2008)
[11] http://www.yahoo.com

und wenn passend instanceOf-Elementen mit den jeweiligen Kategorien verbunden werden. Auch untereinander können die Einträge semantisch verknüpft werden und so eine schnellere Navigation innerhalb des Kataloges ermöglichen.

Suchmaschinen: Suchmaschinen bieten dem Benutzer den Zugriff auf weit mehr Seiten, als es mit einem Katalog möglich ist, da hierbei ein voll automatisierter Ansatz verwendet wird. Ein Webcrawler ist ein Programm, das Seiten des WorldWideWeb analysiert, den in ihnen enthalten Referenzen folgt und diesen Prozess auf den folgenden Seiten fortsetzt (vgl. Harbich, 2008). Entsprechend gefundene Inhalte werden indiziert und gespeichert. Typischerweise wird einem Inhalt, in welchem ein späterer Suchbegriff auftaucht, Relevanz beibemessen, entsprechend der Häufigkeit des Vorkommens. Dies muss aber nicht zum gewünschten Suchergebnis führen. Metasuchmaschinen befragen selbst unterschiedliche Suchmaschinen, je nach Art und Form der Anfrage. Ein Beispiel ist die Suchmaschine KARTOO[12]. Das Ergebnis wird als Karte dargestellt, so dass Zusammenhänge einzelner Quellen ersichtlich werden. Das "Verfeinern"der Suchanfrage kann über die Auswahl vorgeschlagener Topics erfolgen. Wo einem menschlichen Redakteur semantische Zusammenhänge von Quellen intuitiv ersichtlich sind, müssen bei Suchmaschinen Algorithmen diese Zusammenhänge erkennen und in Form von Topic Maps ausdrücken. Der Suchmaschine KARTOO lassen sich auf die Frage "Who invented the Web?"die Topics "Robert Cailiau", "Berners", und "Cern" entnehmen.

2.6.2 Dokumentenmanagement

Der Pool an Dokumenten eines Unternehmens stellt eine, wenn nicht gar die wichtigste, Informationsquelle dar. Da mit wachsender Anzahl das Auffinden des richtigen Dokumentes aber immer schwieriger wird, soll hier Dokumen-

[12] http://www.kartoo.com

tenmanagement Abhilfe schaffen, welchem folgende Aufgaben zugesprochen werden:

- Erfassung von Rechner-extern vorliegenden Dokumenten/Informationen und ihre Aufbereitung in eine geeignete elektronische Form
- Ablage und Speicherung dieser Daten und Dokumente in geeigneten Formaten
- Suchmöglichkeiten (die Recherche) nach Dokumenten im Bestand und der Zugriff darauf
- Bildschirmdarstellung, Drucken sowie Weiterleiten von abgerufenen Dokumenten an andere Kommunikationsverfahren wie etwa Fax oder E-Mail
- Verteilung von Dokumenten, soweit dies erforderlich ist
- Organisation des Daten- und Verarbeitungsflusses der Dokumente in einer Organisation und in Arbeitsabläufen
- Administration der Dokumente und der Ablagestrukturen sowei der Zugriffsrechte von Benutzern

"Darüber hinaus soll ein DMS Schnittstellen zu anderen DV-Anwendungen besitzen oder diese möglichst weitgehend integriert sein"(vgl. Gulbins et al., 1999, S. 1-2 und S. 12). Dokumente werden in einem Dokumenten Management System (kurz DMS) Attribute zugeordnet.

Für eine digitalisierte Eingangsrechnung könnten das z.B. folgende Attribute sein:

Dokumententyp = Eingangsrechnung
Rechnungsnummer = 99654
Lieferantennummer = 2064489
Lieferantenname = Meyer GmbH
Eingangsdatum = 24.07.2008

"Diese Attribute müssen beim Einbringen des Dokumentes in das DMS angelegt bzw. erfasst und in der Datenbank des DMS abgelegt werden. Diese Attribute - auch Indexwerte des Dokuments oder Metadaten genannt - benötigt man, um später nach den Dokumenten suchen (recherchieren) zu können" (siehe Gulbins et al., 1999, S. 19). Da die Anzahl der möglichen Attribute, oder auch Schlagworte, ab einem bestimmten Punkt limitiert sind, lässt sich erahnen, dass bei einem entsprechend großen Dokumentenbestand durchaus mehrere Dokumente mit den gleichen Schlagworten versehen sein können. Auch ist zum Auffinden das Wissen bzw. Kennen der richtigen Schlagworte nötig. Die Anzahl der Suchtreffer steigt also an. Topics können mit den Schlagworten identisch sein. Doch Topic Maps können mehr, sie können die Zusammenhänge zwischen mehreren Dokumenten abbilden und so die Suche erleichtern. Dabei wird jedes Dokument als Topic definiert und zusammengehörende Dokumente als Assoziationen verknüpft. Da es im Standard allerdings nicht vorgesehen ist Assoziationen zwischen Occurences zu modellieren, muss jedes einzelne Dokument ein eigenes Topic sein. Topic Maps könnten also herkömmliche Dokumentenmanagementsysteme um eine semantische Komponente erweitern, welche die Zusammenhänge zwischen einzelnen Dokumenten erschließt. Zusätzlich könnte ihr Einsatz im Austausch zwischen verschiedenen Dokumentenmanagementsystemen liegen.

2.6.3 Datenaustausch im Bereich B2B

Der Datenaustausch zwischen zwei Partnern setzt voraus, dass diese die gleiche Sprache sprechen und die gleiche Begrifflichkeit verwenden. Was zwischen Menschen innerhalb eines Gespräch geklärt werden kann, erweist sich im Softwarebereich ungleich schwieriger. Unternehmen setzen auf unterschiedliche Arten von Software, welche zum Beispiel unterschiedliche Bezeichnungen für die gleichen Daten verwenden. Als Beispiel für eines der älteren Datenformate zur Übertragung von elektronischen Informationen soll hier der DIN ISO 9375 Standard - UN/Edifact (dies steht für United Nations Electronic Data Interchange For Administration, Commerce and Transport) (siehe dazu UN Economic Commission for Europe, 2008) erwähnt werden. Dieser branchenübergreifende internationale Standard für Datenaustausch im Geschäftsverkehr findet nach wie vor Anwendung in der Industrie, z.B. im Automotive-Bereich. Für verschiedene Branchen gibt es Teilmengen des Edifact-Standards, welche auf die speziellen Geschäftsfälle des jeweiligen Bereiches zugeschnitten sind, im Automobil-Bereich lautet diese Teilmenge zum Beispiel "ODETTE". Die Daten einer Edifact-Nachricht bestehen aus ASCII-Zeichen ohne Zeilenumbruch. Um eine Nachricht übermitteln zu können, muss ein Unternehmen die vom Standard vorgegebene Struktur mit den entsprechenden Daten füllen. Nur durch das Wissen über diesen Standard kann eine entsprechende Nachricht verfasst und auch interpretiert werden. Hierbei findet aber keine semantische Verknüpfung der Informationen statt, zumal es sich dabei auch nur um ausgewählte Prozesse wie Lieferabrufe, Rechnungen etc. handelt.
Das generelle Problem bleibt also herauszufinden, welche Daten von unterschiedlichen Ressourcen die gleiche Bedeutung haben, bzw. die gleiche Verwendung finden. Um dies zu erreichen muss man die Bedeutung dieser Daten vergleichen, dass heißt, es müssen unterschiedliche Datendefinitionen für unterschiedliche Datenquellen miteinander abgeglichen werden. Die Schwierigkeit dabei ist, dass unterschiedliche Organisationen sehr unterschiedliche Definitionen für Dinge verwenden, die praktisch identisch sind. Anders her-

um kann es vorkommen, dass ähnlich oder gleich lautende Definitionen für in der Realität sehr unterschiedliche Dinge verwendet werden (vgl. de Graauw, 2002). Eine Lösung wäre, dass sich alle Organisationen auf ein Vokabular einigen könnten, doch ist nach Ansicht des Autors diese Lösung in der Realität nicht umsetzbar. Die Schwierigkeit bestünde nicht im Entwurf sondern in der Akzeptanz aller Organisationen weltweit einem solchen Vokabular gegenüber. Zusätzlich müsste ein allgemeines Vokabular, um erfolgreich zu sein, über eine Schnittstelle zu bestehenden und bereits in Verwendung befindlichen Vokabularien verfügen, so dass Organisationen ihre Daten konvertieren könnten. Das Wort Schnittstelle lässt hierbei einen Lösungsansatz erahnen, also eine übergeordnete Datenebene, die es ermöglicht, verschiedene Ontologien zu verbinden. Mit Topic Maps besteht die Möglichkeit dies umzusetzen. Es geht also darum Metadaten zu generieren, die semantisch zusammenhängende Informationen miteinander verknüpft.
Eines der häufig verwendeten Dokumente in einem Unternehmen stellt die Rechnung dar. Auf einer Rechnung finden sich Adressen, Währungsbeträge, Mengenangaben in unterschiedlichen Einheiten usw. welche sich als Topics definieren lassen. Die Rechnung selbst wäre wiederum ebenfalls ein Topic, z.B. von der Art Dokument. Der Kundenname ist Teil einer Rechnung, was über eine Assoziation abgebildet werden kann. Als kurzes Beispiel sei hier eine mögliche Definition des Topics für einen Kundennamen aufgeführt.

```
<topic id="name">
  <instanceOf>
    <topicRef xlink:href="example.xtm#name"/>
  </instanceOf>
  <baseName>
    <scope>
      <subjectIndicatorRef
          xlink:href="http://www.ontopia.net"/>
    </scope>
    <baseNameString>Kundenname</baseNameString>
  </baseName>
  <occurrence>
```

```
    <instanceOf>
      <topicRef xlink:href="itm.xtm#definition"/>
    </instanceOf>
    <resourceRef xlink:href=
        "http://www.ontopia.net/definition#Kundenname"/>
  </occurrence>
</topic>
```

Dieses Topic ist eine Instanz des Topics “name“ (welches an einer anderen Stelle definiert würde). Der Name “Kundenname“ hat einen zugeordneten Scope auf eine URL, um eine falsche namensbasierte Vermischung mit anderen Topics aus potentiellen anderen Topic Maps zu verhindern. Eine Zusammenführung wäre nur dann möglich, wenn ein anderes Topic mit einem identischen Namen und einem identischen Scope vorhanden wäre. Zusätzlich ist eine externe Ressource angegeben, welche z.B. eine Definition enthalten kann. Gerade durch Definitionen über die Elemente `<scope>` und `<instanceOf>` lassen sich hier idealerweise unterschiedliche Ontologien miteinander in Verbindung setzen.

Kapitel 3

Generierung von XML Topic Maps auf Basis der Wikipedia

In diesem Kapitel wird die Realisierung der Programmierung vorgestellt. Die bereits in Kapitel 2.1 erläuterte Online-Enzyklopädie Wikipedia stellt die Informationsquelle für den praktischen Teil dieser Diplomarbeit dar.

3.1 Aufgabenstellung

Durch die Erstellung einer Topic Map zu einem abgerufenen Wikipedia-Artikel soll die semantische Zusammengehörigkeit von verschiedenen Artikeln (oder auch Schlagworten) innerhalb eines einzelnen Artikels dargestellt werden. Das Zusammenspiel der beiden Diplomarbeiten soll das folgende Diagramm in Abbildung 3.1 verdeutlichen.

Gemäß dem Modell der Topic Maps werden also Meta-Daten über einen einzelnen Artikel erstellt und der Visualisierung übergeben. Um die Antwortgeschwindigkeit für den Endanwender möglichst kurz zu halten und auch die Visualisierung zu begrenzen wird keine rekursive Abfrage der Artikel erstellt, vielmehr ist die Darstellung auf einen einzelnen Artikel beschränkt.

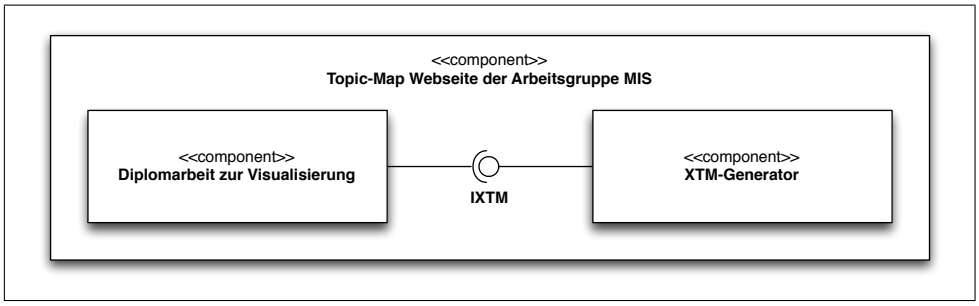

Abbildung 3.1: UML-Komponentendiagramm der Topic-Map Webseite der Arbeitsgruppe MIS

3.1.1 Problemdarstellung

Die Daten der Wikipedia liegen in XHTML (siehe 2.2.4) vor. Es muss also eine Transformierung von XHTML in eine XTM stattfinden. Das Problem dabei ist, aufgrund der Struktur der XHTML-Seiten auf semantische Informationen zu schließen, die innerhalb der XTM abgebildet werden können. Es muss eine Möglichkeit gefunden werden, wie sich Ressourcen und Assoziationen zu den jeweiligen Topics aus den XHTML-Seiten auslesen lassen. Der Informationsgehalt einer Topic Map ist um so höher, je mehr Zusammenhänge zwischen den Informationen erkannt und innerhalb der XTM dargestellt werden können. Ein menschlicher Leser einer XHTML-Seite erkennt allein durch das Wissen um die Sprache (in diesem Anwendungsfall: deutsch), welche inhaltlichen Zusammenhänge zwischen den einzelnen Daten bestehen. In dem Beispiel aus Abschnitt 2.1 mit Bundeskanzlerin Merkel und Staatspräsident Sarkozy erkennt ein Mensch folgende Zusammenhänge, die sich auch innerhalb einer Topic Map abbilden lassen:

- Merkel ist weiblich
- Sarkozy ist männlich
- beide sind Menschen
- beide sind Staatsoberhäupter, jeweils von Deutschland und Frankreich

- beide leben an unterschiedlichen Orten in unterschiedlichen Ländern
- der Euro ist eine Währung
- sowohl in Deutschland als auch in Frankreich gilt der Euro als Zahlungsmittel

Diese Liste liesse sich sicherlich noch weiter fortsetzen. Sie soll aber das Problem verdeutlichen, dass ein Mensch durch Hintergrundwissen in der Lage ist, unterschiedliche Assoziationen zu bilden. Dieses Hintergrundwissen steht einer Maschine rein durch die Struktur der XHTML-Seite nicht oder nur in eingeschränktem Maß, zur Verfügung. In Abschnitt 4 wird ein Ausblick über mögliche Erweiterungen dieses Hintergrundwissens gegeben.

3.1.2 Lösungsansatz

Auf eine Suchanfrage an Wikipedia liefert diese drei unterschiedliche Fälle als Antwort zurück. Die folgenden Antworten sind möglich:

1. dem übergebenen Suchbegriff kann ein Artikel zugeordnet werden, welcher zurück geliefert wird.
2. eine Seite zur Begriffsklärung wird zurückgeliefert, da der Suchbegriff nicht eindeutig ist.
3. es wird keine Entsprechung gefunden und daher eine Suche über die Artikel gestartet.

In jedem der drei Fälle sieht die zurückgelieferte Wikipedia-Seite unterschiedlich aus, sowohl optisch, als auch in der XHTML-Struktur. Die Unterschiede und der Aufbau der jeweiligen Seite soll hier in einem Überblick dargestellt werden:

Artikel:
Bei Auffinden eines Artikels besteht dieser aus einer Überschrift mit dem Titel des Textes. Der weitere Inhalt ist variabel und vom gewählten Aufbau des Autors abhängig. Ein kurzer Artikel kann reinen Text enthalten, ohne zusätzliche Ressourcen wie Bilder oder Links. Im Normalfall enthält aber jeder Text zumindest Wörter mit Verbindungen zu anderen Artikeln. Diese werden durch Hyperlinks realisiert. Der Aufbau eines ausführlichen Artikels beginnt in der Regel mit einem einleitenden Absatz zur kurzen Einführung in das Thema. Danach kann ein Inhaltsverzeichnis mit den enthaltenen Unterabschnitten folgen. Unterabschnitte werden ebenfalls als Überschriften realisiert, wobei diese aber in der Hierarchie auf jeden Fall unter dem Titel des Artikels stehen. Entsprechend der XHTML-Elemente besteht der Titel aus einem `<h1>`-Element, direkt folgende Unterabschnitte bestehen aus einem `<h2>`-, oder bei weiterer Verzweigung sogar aus einem `<h3>`-Element. Sowohl im einleitenden Absatz, als auch in den Unterabschnitten des Textes können Ressourcen eingebunden sein. Generell stellen der Suchbegriff selbst sowie die in einem Artikel vorkommenden Verlinkungen Topics im Sinne des Topic Maps Konzepts dar. Bilder, Audiodateien oder jedwede andere externe zusätzliche Information stellen Ressourcen dar, die dem Suchbegriff, im Folgenden auch MainTopic genannt, entsprechend zugeordnet werden. Da es schwierig erscheint, maschinell Assoziationen zu erstellen (ohne weitere Hilfsmittel), wie sie ein Mensch mit Hilfe seines Hintergrundwissens erstellen kann, werden die in Artikeln vorkommenden Unterabschnitte als Assoziation verwendet. Der bereits genannte einleitende Abschnitt wird hier als Abschnitt “Allgemein“ geführt. Bei dem Beispiel mit Bundeskanzlerin Merkel würde es wie folgt aussehen:

- MainTopic ist “Angela Merkel“.
- Topics für Assoziationen sind unter anderem “Werdegang“, “Politische Positionen“, “Öffentlichkeitsarbeit“, “Familiäres“ und das erwähnte “Allgemein“.

- Verwendete Verlinkungen werden über die Assoziationen (Topics) mit dem MainTopic verknüpft.
- Im Artikel auftauchende Ressourcen werden mit dem MainTopic verbunden (z.B. ein Portrait-Foto, der Mitschnitt eines Interviews, etc.).

Der Aufbau von XML Topic Maps kann individuell zwischen den Parteien, die diese XTM zum Datenaustausch verwenden, abgesprochen werden. Es wäre also durchaus möglich, die Ressourcen nicht direkt an das MainTopic zu knüpfen, sondern durchaus auch, je nach Auftauchen, an den jeweiligen Unterabschnitt. Das in diesen Fall die Artikel-Ressourcen dem MainTopic zugeordnet werden, stellt eine Absprache zwischen dem Autor dieser Diplomarbeit und dem Autor der darauf aufbauenden Arbeit dar.

Begriffklärungsseiten:
Sollte ein Suchbegriff nicht eindeutig sein, so werden von der Enzyklopädie die Begriffklärungsseiten angezeigt. Hierbei werden dem Anwender die verschiedenen Bedeutungen des Begriffes aufgelistet. Zusätzlich sind hier die Verlinkungen zu den eindeutigen Artikeln enthalten. Der Suchbegriff “Queen“ z.B. führt den Anwender auf eine solche Begriffklärungsseite. Abbildung 3.2 veranschaulicht, welche Informationen eine Begriffsklärungsseite der Wikipedia zu dem Suchbegriff “Queen“ liefert.[1] Da es den Autoren von Wikipedia-Artikel freigestellt ist, welche Textteile innerhalb eines solchen Absatzes durch Textformatierung hervorgehoben werden, ist es nicht durchgehend der Fall, dass z.B. im Satz “Queen ist der Nachname folgender Personen“ das Wort “Personen“ oder “Nachnamen“ hervorgehoben wird. Auf dieser Basis hätte die Möglichkeit bestanden, Assoziationen basierend auf XHTML-Elementen zu bilden.

Suchergebnisse:
Sollte ein eingegebener Suchbegriff weder zu einem Artikel, noch zu einer Begriffklärungsseite führen, so wird automatisch von der Wikipedia eine Voll-

[1] http://de.wikipedia.org/wiki/Queen - Stand 7.9.2008

Abbildung 3.2: Begriffklärungsseite für “Queen“

textsuche über alle Artikel gestartet und dem Anwender in einer Spezialseite die Ergebnisse präsentiert. Für den Suchbegriff “Merkel“ wird unter anderem folgendes Ergebnis geliefert:

- **Angela Merkel**
  "'Angela Dorothea Merkel"', geborene Kasner (* [[17. Juli]] [[1954]] in [[Hamburg]]), ist eine [[D ... [[Bild:Angela Merkel (2008).jpg—thumb—Angela Merkel (2008)]] 57 KB (7368 Wörter) - 12:53, 5. Sep. 2008

- **Max Merkel**
  "'Max Merkel"' (* [[7. Dezember]] [[1918]] in [[Wien]]; † [[28. November]] [[2006]] ... ... ckte Fußballspieler diente. Nach Ende des Zweiten Weltkrieges kehrte Max Merkel schließlich 1946 nach Hütteldorf zurück, wurde mit Rapid in der Folgeze ...
  5 KB (731 Wörter) - 09:06, 31. Jul. 2008

- **Merkel-Zelle**
  ... Der Komplex aus Merkel-Zellen und [[Nozizeptor—Nervenendigung]] wird als "Merkel-Scheibe" bezeichnet. Merkel-Zellen gehören zu den

[[Mechanorezeptoren der Haut—Mechanorezeptoren]] de ...
1 KB (184 Wörter) - 16:36, 18. Aug. 2008

Die in diesem Beispiel fett markierten Wörter stellen eine Verlinkung auf den gleich lautenden Artikel dar. Wie in den oben genannten Fällen bildet auch hier der Suchbegriff das MainTopic der zu erstellenden XTM. Die durch die Suche gefundenen Artikel werden ebenfalls als Topics deklariert und über entsprechende Assoziationen, welche den möglichen Treffer der Suche ausdrückt, mit dem MainTopic verknüpft.

3.2 Die Klassen des XTM-Generators

Der XTM-Generator besteht aus fünf Klassen und wurde in der Programmiersprache Java[2] geschrieben. In diesem Abschnitt sollen diese Klassen kurz in der Reihenfolge ihrer Verwendung vorgestellt werden. Abbildung 3.3 stellt ein UML-Klassendiagramm über die Klassen der Anwendung dar. Ein Klassendiagramm "beschreibt die Typen von Objekten im System und die verschiedenen Arten statischer Beziehungen"(Fowler und Scott, 1998, S. 61).

IXTM: Diese stellt das Interface des XTM-Generators zur Verfügung. Über das Interface ist die Methode getXTM() zu erreichen. Dieser wird von der aufrufenden Applikation entweder nur der zu bearbeitende Suchstring übergeben, oder in der 2-Parameter-Variante der Suchstring und zusätzlich ein Integer-Wert, welcher eine Limitierung der zu verarbeitenden Suchergebnisse angibt.

XTM: Die zentrale Klasse des XTM-Generators stellt die XTM.java dar. Über diese Klasse läuft die logische Abarbeitung der einzelnen Schritte bis zur Erstellung der XTM-Struktur innerhalb eines Dokumenten-Objekts, welches auch den Rückgabewert der Methoden darstellt. An-

[2] http://java.sun.com

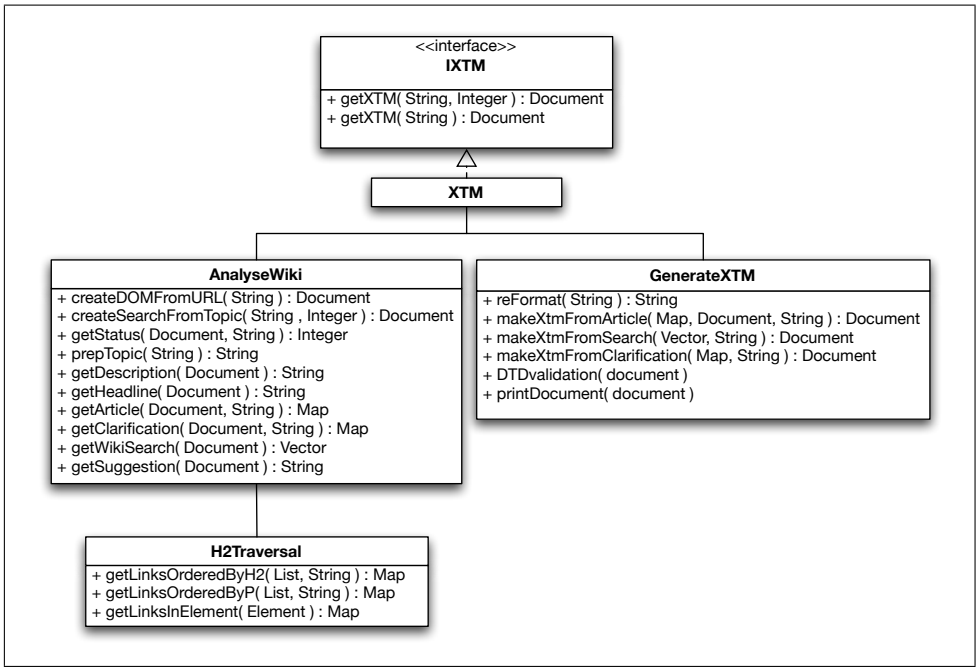

Abbildung 3.3: UML-Klassendiagramm des XTM-Generators

hand eines ermittelten Status wird zwischen den in Abschnitt 3.1.2 vorgestellten möglichen Varianten unterschieden, die auf eine Anfrage an Wikipedia als Ergebnis zurückgeliefert werden können.

AnalyseWiki: Diese Klasse beschäftigt sich mit der Anfrage an die Online-Enzyklopädie sowie mit der Ermittlung der Daten, die für die spätere Erstellung der XTM benötigt werden. Es werden die Methoden zur Verfügung gestellt, welche für die Ermittlung und Aufbereitung der Daten notwendig sind. Mittels XPath werden innerhalb der XHTML-Dokumente direkt Knoten in der XML-Hierarchie angesprochen, um das Auffinden von relevanten Strukturen zu verkürzen und die Verarbeitung zu erleichtern. XPath verwendet ein XML-Dokument und einen Suchstring als Input und erstellt eine Liste von Knoten, auf welche das Suchkriterium zutrifft. XPath ist darauf spezialisiert Daten innerhalb von Dokumenten zugänglich zu machen (vgl. Kay, 2004, S. 2). Zusätzlich greift AnalyseWiki auf die Methoden der Klasse H2Traversal zurück, welche anhand der vorgefilterten Knotenstrukturen die wei-

terführenden Links und Ressourcen (in Abhängigkeit des Abschnitts, in welchem sie vorkommen) ermittelt.

H2Traversal: Die hierin enthaltenen Methoden durchsuchen die von AnalyseWiki übergebene Knotenstruktur des XHTML-Dokumentes nach Links und Ressourcen. Zusätzlich wird vermerkt, in welchem Abschnitt der Wikipedia Seite diese Information aufgetaucht ist. Anhand der Abschnitte wird beim Generieren der XTM-Struktur die jeweilige Assoziation zwischen dem gesuchten Begriff und weiterführenden Artikeln gebildet.

GenerateXTM: Die ermittelten Informationen über die Wikipedia-Seite werden in dieser Klasse zu der gewünschten XTM-Struktur verarbeitet. Hierbei werden, basierend auf den in Abschnitt 2.4 vorgestellten Elementen und unter Berücksichtigung der XTM 1.0 DTD, unterschiedliche Methoden zur Generierung verwendet. Dies ist von dem zuvor in AnalyseWiki ermittelten Status abhängig. Zum Ende des Verarbeitungsschrittes, bietet die GenerateXTM die Möglichkeit, eine Validierung der XTM-Struktur auf Basis der Dokumententyp Definition von XTM 1.0 zu prüfen.

3.3 Ablaufschema des Programms

In diesem Abschnitt soll die Arbeitsweise des Programms näher erläutert werden. Die aufrufende Komponente (siehe Abbildung 3.1) übergibt über das Interface IXTM einen Suchstring an die Klasse XTM. Hierbei ist eine Größenbeschränkung für evtl. Suchergebnisse optional, welche in Form eines Integer-Wertes übergeben werden kann. Dies hat sich als notwendig herausgestellt, da andernfalls die aufrufende Komponente aufgrund der zu verarbeitenden Menge während der Abarbeitung einen undefinierten Zustand erreichte. Wird der Optionale Wert nicht mit übergeben, so wird standardmässig ein Wert von 999 für die maximale Anzahl der Suchergebnisse verwendet.

Die folgenden Schritte sollen durch Abbildung 3.4 dargestellete UML-

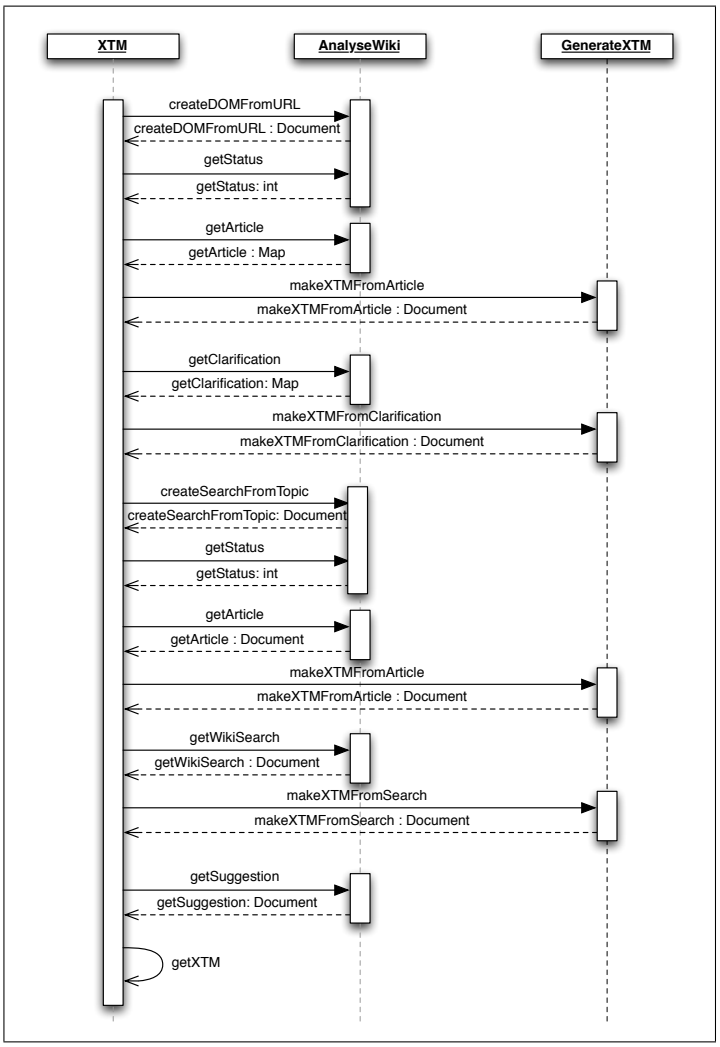

Abbildung 3.4: UML-Sequenzdiagramm der XTM.java

Sequenzdiagramm verdeutlicht werden. Ein Sequenzdiagramm stellt "den Informationsaustausch zwischen beliebigen Kommunikationspartnern innerhalb eines Systems oder zwischen Systemen generell"dar (Rupp et al., 2005, Seite 407). Anhand des übergebenen Suchstrings ermittelt die Klasse XTM unter Verwendung der Methode `createDOMFromURL()` die Antwort-Seite der Wikipedia und bekommt diese als Dokumenten-Objekt zurückgeliefert. Durch die Methode `getStatus()` findet eine Fallunterscheidung statt, wie in Abschnitt 3.1.2 beschrieben. Je nach zurückgeliefertem Integer-Wert der

Methode, wird die weitere Verarbeitung unterschiedlich fortgesetzt. Der Status wird anhand von unterschiedlichen Signalwörtern ermittelt, welche sich innerhalb des XHTML-Dokumentes an bestimmten Stellen in der Struktur befinden. Befindet sich innerhalb des Dokumentes ein Element für einen Hyperlink, welches das Attribut `title='Wikipedia:Begriffsklärung'` enthält, so handelt es sich bei diesem Dokument und eine Begriffsklärung. Der ermittelte Status lautet somit *1*.
Finden sich innerhalb des Dokumentes entweder der Text `Diese Seite existiert nicht` oder `Es existiert kein Artikel mit dem Namen` so konnte Wikipedia keine Entsprechung des Suchstrings ermitteln. In diesem Fall lautet der ermittelte Status auf *2*.
In seltenen Fällen erkennt Wikipedia falsche Schreibweisen für einen Suchbegriff. In einem solchen Fall schlägt die Enzyklopädie einen Artikel vor, welcher mit einer ähnlichen Schreibweise verfügbar ist. Ein Beispiel hierfür ist der Name "Gandhi". Wird dieser in der Suchanfrage fälschlicherweise als "Ghandi" angegeben, so erfolgt der Vorschlag auf den Artikel mit korrekter Schreibweise. In einem solchen Fall wird der Wert *666* zurückgeliefert. Da sich nur in wenige Testfällen ein solcher Vorschlag ergab, ist zu vermuten, dass es sich hierbei um innerhalb der Wikipedia vordefinierte Fehlerfälle handelt, auf welche dann reagiert wird.
Wurde während der Verarbeitung kein anderer Status ermittelt, so handelt es sich bei dem Suchergebnis um einen regulären Artikel. In diesem Fall wird für den Status eine *0* gemeldet.

Im folgenden Abschnitt wird weiter auf die jeweiligen Fallunterscheidungen eingegangen.

status = 0: Wird für das Dokument der Status null ermittelt, so beginnt die Verarbeitung mit der Methode `getArticle()`. Hier wird mittels XPath die Datenmenge verkleinert, indem der Knotenpunkt für den Beginn der Verarbeitung innerhalb des Dokumentes auf das Element `<div id="bodyContent»` gelegt wird. Der in Abbildung 3.5 rot (1) markierte Name ist identisch mit dem Suchergebnis und wird in der zu erstellen-

Abbildung 3.5: Verwendung eines Wikipedia-Artikels

den XTM das 1. Topic dargestellen. Der darunter folgende Abschnitt (in orange (2) dargestellt) stellt die Erläuterung des Topics dar und wird als Ressource eingebunden. Das Foto der Künstlerin ebenfalls (mit lila (4) markiert). Die Unterabschnitte des Artikels (wie z. B. der grün (3) kenntlich gemachte Abschnitt “Biografie“) werden zur Unterscheidung der Hyperlinks (Text in hellblauer Schrift) verwendet. Je Bereich werden die Links einem Abschnitt zugeordnet, welcher wiederum in der XTM dem Topic des Artikels zugeordnet ist. Die ermittelten Daten werden an die Klasse XTM zurück geliefert, welche wiederum mittels der Methode `makeXtmFromClarification()` diese Daten an die Klasse GenerateXTM übermittelt. Dem Topic Map Modell entsprechend stellen sowohl alle Links innerhalb des Artikels sowie die Titel der Abschnitte ein Topic innerhalb der XTM-Struktur dar. Die einzelnen Links sind mittels Assoziation mit den dazugehörigen Abschnitts-Topics verknüpft. Die Abschnitte selbst sind wiederum mit dem Haupttopic der Topic Maps verbunden. Sowohl die Erläuterung als auch evtl. vorhan-

den Bilddateien werden dem Haupttopic als Occurrences zugeordnet. Eine Beispiel-XTM zu dem Artikel über die Künstlerin Anke Engelke finden Sie auf dem digitalen Datenträger, welcher der Diplomarbeit beiliegt.

Abbildung 3.6: Verwendung einer Begriffsklärungsseite

status = 1: Wenn für das Dokument der zurückgelieferten Wikipediaseite der Status eins ermittelt wird, beginnt die Verarbeitung mit der Methode `getClarification()`. Analog zur Artikelverarbeitung wird mittels XPath ein Einstiegsknoten innerhalb des XHTML-Dokumentes gewählt um die Verarbeitung zu erleichtern. Sowohl optisch als auch strukturell unterscheiden sich Begriffsklärungsseiten von den regulären Artikeln (siehe Abbildung 3.6). Begriffsklärungsseiten stellen in Listenform die unterschiedlichen Bedeutungen des Suchbegriffs dar und bieten somit über die Verlinkungen eine Möglichkeit zum gewünschten Artikel zu navigieren. So erfolgt die Ermittlung der benötigten Informationen anhand der Absatz-Elemente `<p/>`. Diese Informationen übergibt die Klasse XTM wiederum an GenerateXTM. Hierzu wird allerdings eine andere Methode für die Erstellung der XTM-Struktur verwendet, die `makeXtmFromClarification()`. Bei Begriffsklärungsseiten stellt sich

auch der Seitentitel erneut als Haupttopic dar (rot (1) markiert). Die einzelnen Kategorien, in welche die Begriffe eingeteilt werden, stellen hier die Abschnitte ähnlich denen in den Artikeln dar (grün (2) markiert). Ebenfalls werden die einzelnen Hyperlinks jeder für sich als einzelnes Topic abgebildet. Im Gegensatz zu einem Artikel erhält hier jedes Topic eine eigene Erläuterung, entsprechend dem jeweiligen Listenpunkt (orange (3) markiert).

status = 2: Sollte es für den gewählten Suchbegriff keine Entsprechung in der Wikipedia geben, so wird eine Seite zur Eingabe einer Volltextsuche angezeigt. Hierbei wird der Status zwei ermittelt. Um die Benutzereingabe für die Suche zu simulieren, wird eine erneute Anfrage an die Enzyklopädie über die entsprechende URL der Suchseite gestellt. Hierzu ist eine Formatierung des Suchstrings in das Format "UTF-8" notwendig um diesen in der URL zu platzieren. Dies Umformatierung wird mittels der Methode `reFormat` erreicht. Über die Methode `createSearchFromTopic()` wird eine Suchanfrage an die Wikipedia gestellt, welche ebenfalls ein Dokument zurück liefert. An dieser Stelle ist eine erneute Statusermittlung notwendig, da es möglich ist, dass anhand der Suche ein Artikel gefunden wurde. In diesem Fall ist der ermittelte Status erneut null und die Verarbeitung erfolgt wie bereits oben beschrieben. In jedem anderen Fall wird das Dokument mittels `getWikiSearch()` zur weiteren Verarbeitung an die Klasse AnalyseWiki übergeben. Auch hier wird mittels XPath der Einstiegsknoten für die Suche innerhalb des Wiki-Dokumentes gesucht und die gefundenen Verweise auf mögliche Artikel inklusive deren Beschreibung zurückgeliefert (siehe Abbildung 3.7). Der für die Suche verwendete Begriff entspricht erneut dem Haupttopic (rot (1) markiert). Allerdings fehlen in diesem Fall die Unterabschnitte, so dass die gefunden Artikel-Links (grün (2) markiert) direkt über Assoziationen mit dem Suchbegriff verknüpft werden. Als Occurrence wird jedem Topic der entsprechende Erläuterungstext angefügt (orange (3) markiert).

Abbildung 3.7: Verwendung einer Wikipedia-Suche

Die entsprechende Generierung der XTM wird mittels der Methode `makeXtmFromSearch()` vorgenommen.

status = 666: Wie bereits im oberen Abschnitt erläutert, kann es in einigen wenigen Situationen vorkommen, dass die Enzyklopädie einen möglichen Artikel vorschlägt, sollte der Suchbegriff falsch geschrieben sein. In diesem Fall wird mittels der Methode `getSuggestion()` der vorgeschlagene Artikelname ermittelt und mit diesem als neuem Suchbegriff die Methode `getXTM()` erneut aufgerufen. Die Verarbeitung beginnt daraufhin von neuem. Abbildung 3.8 zeigt einen solchen Vorschlag durch Wikipedia. Der vorgeschlagene Artikel ist rot (1) markiert.

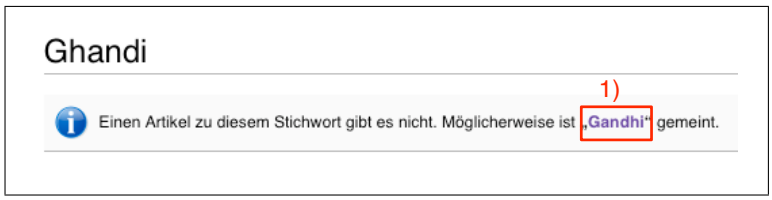

Abbildung 3.8: Vorschlag eines Wiki-Artikels bei falscher Schreibweise

Kapitel 4

Zusammenfassung und Ausblick

In Kapitel zwei der Diplomarbeit wurden verschiedene Grundlagen für diese Arbeit aufgeführt und erläutert. Über die Vorstellung der Onlineenzyklopädie Wikipedia und deren Autorensystem führt das Kapitel in das Thema XML und damit verbunden Informationen ein. Anschließend wurde das Modell der Topic Maps und der dazugehörige ISO Standard 13250 vorgestellt und die Idee hinter dem Modell erläutert. Auch die Formulierung des Modells in XML als XML Topic Maps wurde vorgestellt und bildet die Grundlage dieser Arbeit. Zusätzlich finden sich im zweiten Kapitel mögliche praktische Einsatzmöglichkeiten für Topic Maps. In Kapitel drei wird schließlich das Java-Programm XTM-Generator vorgestellt, welches auf Basis von Suchanfragen an Wikipedia XML Topic Maps aus den Ergebnissen generiert und aufrufenden Programmen als Dokumenten-Objekt zur Verfügung stellt. Die Topic Map Webseite der Arbeitsgruppe MIS, in welcher auch der XTM-Generator als Komponente eingeflossen ist, ist unter *http://bauhaus.cs.uni-magdeburg.de/wikitm* zu erreichen.
Wie bereits in der Arbeit erwähnt ist der Nutzen einer Topic Map um so höher, je mehr Informationen einfließen können. Im Falle dieser Diplomarbeit beruht das ermitteln der nötigen Informationen auf der Struktur der von der Enzyklopädie zurückgelieferten XHTML-Seiten. Doch ist diese Struktur in Details auch abhängig vom jeweiligen Autor des Artikels, so dass nicht in jedem Fall auf die benötigten Informationen geschlossen werden kann. Eine Möglichkeit um die Qualität der Informationen zu erhöhen wäre die Verwendung von Thesauren oder Lexika. Durch einen Abgleich der Informationen

mit selbigen könnten Querverweise zu gleichartigen und verwandten Topics gezogen werden (Thesauren) oder zusätzliche Einordnungsmöglichkeiten verschiedener Typen von Topics gefunden werden.

Literaturverzeichnis

Berners-Lee, S. T. (1997): Weaving the Web - The original Design and Ultimate Destiny of the World Wide Web by its Inventor. Harper San Francisco.

Daconta, M. C., Obrst, L. J. und Smith, K. T. (2003): The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management. Wiley Publishing, Inc., Indianapolis, Indiana.

de Graauw, M. (2002): *Business Maps: Topic Maps Go B2B.* O'Reilly XML.com.

DENIC Domain Verwaltungs und Betriebsgesellschaft eG (2008): *Statistiken und Informationen der Verwaltungsgesellschaft für .de-Domains.* http://www.denic.de. Stand 10.08.2008.

Fowler, M. und Scott, K. (1998): UML konzentriert. Addison Wesley Longman Verlag GmbH.

Free Software Foundation (2008): *Originaltext der GFDL.* http://www.gnu.org/copyleft/fdl.html. Stand: 31.07.2008.

Geroimenko, V. (2004): Dictionary of XML Technologies and the Semantic Web. Springer-Verlag London Limited.

Gulbins, J., Seyfried, M. und Strack-Zimmermann, H. (1999): Dokumenten-Management - Vom Imaging zum Business-Dokument. 2. Auflage, Springer Verlag.

Harbich, R. (2008): *Webcrawling – Die Erschließung des Webs.* http://www.uni-magdeburg.de/harbich/webcrawling.php. Stand: 10.08.2008.

Heinz Wittenbrink, Werner Köhler, e. a. (2003): XML: Wissen das sich auszahlt. TEIA Lehrbuch Verlag GmbH.

International Organization for Standardization (2002): ISO/IEC 13250 Topic Maps. International Organization for Standardization.

Kay, M. (2004): Xpath 2.0 Programmer's Reference (Programmer to Programmer). Wiley & Sons, 1. Auflage.

Mintert, S. (2008): *edition-w3c*. http://www.edition-w3c.de/TR/2002/REC-xhtml1-20020801/. Die edition W3C.de hat die Veröffentlichung sämtlicher W3C-Empfehlungen (Recommendations) in deutscher Sprache und fachlicher Kommentierung zum Ziel. Die Kommentierung wird durch Experten des jeweiligen Gebiets angefertigt. Neben der Online-Veröffentlichung erscheinen die Übersetzungen auch in einer Buchreihe, die von Addison-Wesley verlegt wird. Die edition W3C.de ist die einzige vom W3C legitimierte Publikation in deutscher Sprache.

Mück, P. D. T. und Widhalm, R. (2002): Topic Maps - Semantische Suche im Internet. Springer Verlag.

Mück, T. A. und Widhalm, R. (2001): Schlagwort - Topic Maps, Band Heft 3 von *Wirtschaftsinformatik*, S. 297 – 300. Vieweg Verlag, Wiesbaden.

Möhr, D. W. und Schmidt, I. (1999): SGML und XML. Springer Verlag Berlin Heidelberg.

Park, J. und Hunting, S. (2003): XML topic maps: creating and using topic maps for the Web. Addison-Wesley, Pearson Education.

Rupp, C., Hahn, J., Queins, S., Jeckle, M. und Zengler, B. (2005): UML 2 glasklar - Praxiswissen für die UML-Modellierung und -Zertifizierung. Carl Hanser Verlag München Wien.

UN Economic Commission for Europe (2008): *UN Economic Commission for Europe*. http://www.unece.org. Stand 20.09.2008.

Vonhoegen, H. (2005): Einstieg in XML. Galileo Press.

Wikimedia Foundation Inc. (2008): *Wikipedia über Wikipedia*. http://de.wikipedia.org/wiki/Wikipedia. Stand: 31.07.2008.

Word Wide Web Consortium (2007): *XML Path Language (XPath) 2.0*. http://www.w3.org/TR/xpath20/.

Abschließende Erklärung

Ich versichere hiermit, dass ich die vorliegende Diplomarbeit selbständig, ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.

Magdeburg, den 22.10.2008