

# **Topic Maps und ihre Anwendbarkeit im webbasierten Lern- und Autorensystem Companion**

**Studienarbeit  
Fakultät für Informatik  
Universität Karlsruhe**

**vorgelegt von  
Tom Gelhausen**

**Betreuung:**  
**Prof. Dr. Peter Deussen**  
Institut für Logik, Komplexität und Deduktionssysteme  
**Dipl.-Inform. Thomas Brückner**  
Zentrum für Multimedia

# Inhalt

<b>1</b>	<b>EINLEITUNG: WAS SIND TOPIC MAPS?</b> .....	<b>5</b>
<b>1.1</b>	<b>EIN EINFACHES BEISPIEL</b> .....	<b>5</b>
<b>2</b>	<b>TOPIC MAPS UND COMPANION - SCHNITTSTELLEN UND EINSATZGEBIETE</b> .....	<b>9</b>
<b>2.1</b>	<b>SCHNITTSTELLEN</b> .....	<b>9</b>
<b>2.2</b>	<b>EINSATZGEBIETE</b> .....	<b>9</b>
<b>2.2.1</b>	<b>TEXTBASIERTE SUCHFUNKTION</b> .....	<b>10</b>
<b>2.2.2</b>	<b>NAVIGATION DURCH NICHTLINEARE WISSENSFELDER</b> .....	<b>10</b>
<b>2.2.3</b>	<b>ERZEUGUNG VON MIND MAPS</b> .....	<b>11</b>
<b>3</b>	<b>TOPIC MAPS IM DETAIL</b> .....	<b>12</b>
<b>3.1</b>	<b>DER STANDARD</b> .....	<b>12</b>
<b>3.1.1</b>	<b>TOPIC MAP ARCHITECTURAL FORM (TOPICMAP)</b> .....	<b>12</b>
<b>3.1.2</b>	<b>TOPIC LINK</b> .....	<b>13</b>
<b>3.1.2.1</b>	<b>TOPIC LINK ARCHITECTURAL FORM (TOPIC)</b> .....	<b>14</b>
<b>3.1.2.2</b>	<b>TOPIC NAME ARCHITECTURAL FORM (TOPNAME)</b> .....	<b>16</b>
<b>3.1.2.3</b>	<b>TOPIC OCCURRENCE ARCHITECTURAL FORM (OCCURS)</b> .....	<b>18</b>
<b>3.1.3</b>	<b>ASSOCIATION LINK</b> .....	<b>19</b>
<b>3.1.3.1</b>	<b>ASSOCIATION LINK ARCHITECTURAL FORM (ASSOC)</b> .....	<b>19</b>
<b>3.1.3.2</b>	<b>ASSOCIATION ROLE ARCHITECTURAL FORM (ASSOCRL)</b> .....	<b>20</b>
<b>3.1.4</b>	<b>FACET LINKING</b> .....	<b>21</b>
<b>3.1.4.1</b>	<b>FACET LINK ARCHITECTURAL FORM (FACET)</b> .....	<b>21</b>
<b>3.1.4.2</b>	<b>FACET VALUE ARCHITECTURAL FORM (FVALUE)</b> .....	<b>22</b>
<b>3.1.5</b>	<b>THEMES TO BE ADDED (ADDTHMS)</b> .....	<b>23</b>
<b>3.2</b>	<b>KOMMENTAR</b> .....	<b>24</b>
<b>3.2.1</b>	<b>SCOPES</b> .....	<b>24</b>
<b>3.2.2</b>	<b>FILTER: SCOPE VERSUS FACETTE VERSUS ASSOZIATION</b> .....	<b>25</b>
<b>3.2.3</b>	<b>NICHTABZÄHLBARE ATTRIBUTE, UNENDLICHE WERTEBEREICHE</b> .....	<b>26</b>
<b>4</b>	<b>ERZEUGUNG DER METADATEN</b> .....	<b>28</b>

4.1	DIE QUALITÄT EINER TOPIC MAP .....	29
4.2	GRANULARITÄT.....	29
4.3	MANUELL, SEMI- ODER VOLLAUTOMATISCH? .....	30
5	EIN PRAKTISCHES BEISPIEL: TOPICMAP.NAVIGATOR .....	32
5.1	EINGESETZTE TECHNOLOGIEN.....	32
5.2	DER ANSATZ.....	33
5.3	COMPANIONTOPICMAPS.....	34
5.4	DAS INTERFACE .....	36
5.4.1	LOAD (NAME) .....	36
5.4.2	SAVE (NAME).....	37
5.4.3	GOTOTOPIC (SGML_ID).....	37
5.4.4	GETCURRENTTOPICID () .....	37
5.4.5	GETCURRENTTOPICNAME () .....	37
5.4.6	GETCURRENTTOPICOCCURRENCE () .....	37
5.4.7	UNLINKTOPICS (SGML_ID1, SGML_ID2) .....	38
5.4.8	UNLINKALLFROMTOPIC (SGML_ID).....	38
5.4.9	REMOVETOPIC (SQML_ID).....	38
5.4.10	REMOVECURRENTTOPIC () .....	38
5.4.11	ADDCHILD/ADDPARENT/ADDJUMP (SGML_ID_OLD, SGML_ID_NEW, NAME, OCCURS).....	38
5.4.12	ADDCHILDTOCURRENT/ADDPARENTTOCURRENT/ADDJUMPTOCURRENT (SGML_ID_NEW, NAME, OCCURS) .....	39
5.4.13	SETNEWID (SGML_ID_OLD SGML_ID_NEW).....	39
5.4.14	SETNAME (SGML_ID, NAME) .....	39
5.4.15	SETOCCURRENCE (SGML_ID, OCCURS).....	40
5.5	DIE IMPLEMENTATION .....	40
5.5.1	TOPICMAP.NAVIGATOR.....	40
5.5.2	TOPICMAP.TOPICBRAINMAPPING.....	41
5.5.3	TOPICMAP.LOADER UND ~.SAVER .....	41
5.5.4	TOPICMAP.ABSTRACTEVENTHANDLER UND ~.EVENTHANDLER.....	42
6	AUSBLICK .....	44
7	ANHANG.....	45
7.1	BEISPIEL TOPIC MAP „PC“ .....	45

<b>7.1.1</b>	<b>DIAGRAMM .....</b>	<b>45</b>
<b>7.1.2</b>	<b>CODE.....</b>	<b>45</b>
<b>7.2</b>	<b>TOPIC MAP DTD .....</b>	<b>50</b>
<b>7.3</b>	<b>EINE LEERE COMPANIONTOPICMAP .....</b>	<b>52</b>
<b>7.4</b>	<b>ANWENDUNGSBEISPIEL FÜR TOPICMAP.NAVIGATOR .....</b>	<b>53</b>
<b>8</b>	<b>INDEX .....</b>	<b>55</b>

# 1 Einleitung: Was sind Topic Maps?

---

Diese Studienarbeit beschäftigt sich mit dem Standard ISO 13250 Topic Maps und deren Anwendung im Companion Projekt<sup>1</sup>. Topic Maps stellen ein standardisiertes, austauschbares Hypertext-Navigationssystem oberhalb bestehender elektronischer Informationsressourcen dar. Sie ermöglichen es, virtuelle Landkarten für im Web, in Knowledge Bases oder in Bibliotheken gespeichertes Wissen zu erstellen.

Technisch gesehen transportieren Topic Maps Metainformationen, die es ermöglichen, Zusammenhänge zwischen Themen nachzuvollziehen und auf dieser Basis relevante Informationsressourcen finden zu können. Diese Metainformationen können unabhängig von den eigentlichen Ressourcen erstellt, gespeichert, ausgetauscht und verarbeitet werden. Die Ressourcen bleiben unabhängig von einer Topic Map; somit können beliebig viele Topic Maps zu einem bestehenden Informationspool erstellt werden und beispielsweise einzelne Meinungen zu bestimmten Themen repräsentieren.

Zunächst wird dem Leser das Thema Topic Maps anhand eines Beispiels näher gebracht. Trotz seines vergleichsweise geringen Detaillierungsgrades beschreibt es alle wesentlichen Aspekte des zugrundeliegenden Konzeptes. Im theoretischen Teil dieser Studienarbeit wird im weiteren Verlauf stellenweise Bezug auf dieses Beispiel genommen.

Dem Beispiel folgen in den Kapiteln 2, 3 und 4 theoretische Grundlagen und Betrachtungen zum Thema. Kapitel 2 beschreibt zunächst, in wie weit sich Topic Maps für Companion eignen, während Kapitel 3 den Standard selbst behandelt, also seine Formulierungen und deren Funktion. Kapitel 4 widmet sich der Frage nach der Gewinnung von Daten für Topic Maps. Abgeschlossen wird diese Studienarbeit durch das Kapitel 5 „Ein praktisches Beispiel: TopicMap.Navigator“. Dort geht es um die im Rahmen dieser Studienarbeit realisierte Anwendung von Topic Maps im Companion-Projekt.

## 1.1 *Ein einfaches Beispiel*

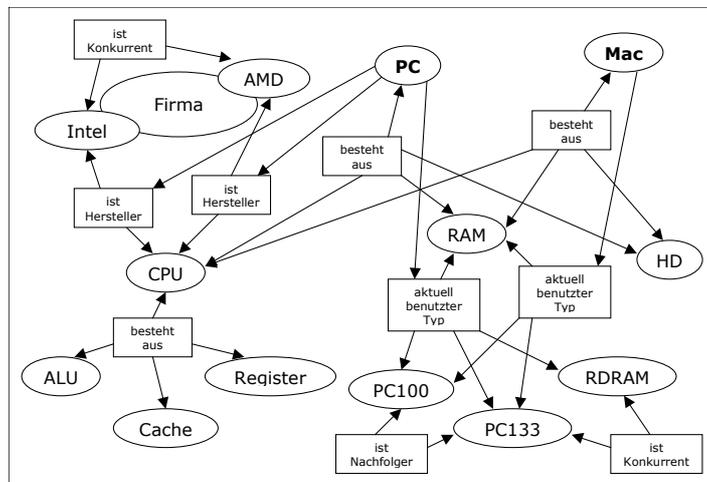
Dieser Abschnitt führt in die Thematik der Topic Maps anhand eines einfachen Beispiels ein. Modelliert wird ein kleiner Ausschnitt aus der Welt der Personal

---

<sup>1</sup> Siehe hierzu <http://vikar.ira.uka.de/companion/index.html>

Computer. Ein solcher Computer (entweder ein PC oder ein Macintosh-Rechner) besteht aus einer CPU, Arbeitsspeicher (RAM) und einer Festplatte (HD). Weitere Details können der nachfolgenden Grafik entnommen werden.

Zunächst wird das zu beschreibende Themengebiet in *Topics* unterteilt. Diese sind in der Grafik durch ein Oval symbolisiert. Topics dieses Gebietes sind neben PC, Mac, CPU, RAM und HD auch AMD, Intel, ALU, Cache, Register, PC100



(ein Speichertyp), PC133 und RDRAM (ebenfalls Speichertypen). Zwischen den Topics gibt es Verbindungen, in der Sprache der Topic Maps *Assoziationen* genannt. In der Grafik sind Verbindungen durch Pfeil/Kästchen-Kombinationen gekennzeichnet, der Inhalt des Kästchens beschreibt den Typ der Verbindung. Hier im Beispiel existieren die Assoziationen „besteht aus“, die „ist Hersteller“, die „ist Konkurrent“, die „ist Nachfolger“ und „ist aktuell benutzter Typ“. Die Richtung der Pfeile verdeutlichen die Einflussnahme der einzelnen Objekte untereinander. So gilt beispielshalber die Beziehung „ist Hersteller“ zwischen Intel und CPU nur im Rahmen von PC's, nicht aber bei Macintosh Computer.

Topic Maps, auf einen Datenträger gebracht, sind SGML-, spezieller XML-Dateien. Das Format dieser Dateien ist standardisiert und wird in Kapitel 3 ausführlich beschrieben. Für das hier angeführte Beispiel wird die Document Type Definition (DTD) der Firma STEP (siehe Anhang 7.2) angewendet. Die Zusammenhänge dieses Beispiels werden dieser entsprechend folgendermaßen beschrieben:

```
<topic id="ID_pc">
  <topname>
    <basename>PC</basename>
  </topname>
</topic>
```

Definiert wird ein Topic mit dem Namen „PC“. Um im weiteren Verlauf auf dieses Topic zugreifen zu können, erhält das Topic ein in diesem Dokument eindeutiges Attribut *id* zugewiesen, in diesem Fall *ID\_pc*. Es hätte an dieser Stelle auch „2h435bj626k“ oder nur „PC“ verwendet werden können. Sie wurde aber so gewählt, dass der Unterschied zwischen dem Topic, beziehungsweise einer Referenz

darauf, und seinem Name klar wird. Andererseits sollte man in diesem Beispiel direkt erkennen können, um welches Topic es sich hier handelt.

Ebenso werden nun die Topics *ID\_cpu*, *ID\_amd* und *ID\_intel* definiert:

```
<topic id="ID_cpu">
  <topname>
    <basename>CPU</basename>
    <dispname>Prozessor</dispname>
  </topname>
</topic>
<topic id="ID_amd" types="ID_firma">
  <topname>
    <basename>AMD</basename>
  </topname>
  <occurs type="ID_homepage">http://www.amd.com</occurs>
  <occurs type="ID_news">
    http://www.heise.de/newsticker/data/jow-11.12.00-000/
  </occurs>
</topic>
<topic id="ID_intel" types="ID_firma">
  <topname><basename>Intel</basename></topname>
  <occurs type="ID_homepage">http://www.intel.com</occurs>
  <occurs type="ID_news">
    http://www.heise.de/newsticker/data/jow-11.12.00-000/
  </occurs>
</topic>
```

Neu ist hier bei *ID\_cpu* das Tag *dispname*. Es gibt an, dass in einer Applikation, die Topic Maps verarbeitet, der Begriff „Prozessor“ statt dem Begriff „CPU“ für die Anzeige verwendet werden soll. Auch anhand von *ID\_amd* und *ID\_intel* wird ein neues Tag vorgeführt: Das *occurs* Tag stellt die Verbindung von Topic Maps zu externen Informationsressourcen her. In diesem Beispiel sind das Hyperlinks. Es könnten aber auch Verweise auf nichtelektronische Medien sein, wie etwa ein Artikel in einer Fachzeitschrift oder eine ISBN-Nummer.

Ebenfalls neu ist die Verwendung der *type* und *types* Attribute. Sie weisen darauf hin, dass *ID\_intel* eine Firma ist (im Diagramm durch die Hinterlegung angedeutet) und dass <http://www.heise.de/newsticker/data/jow-11.12.00-000/> auf Nachrichten zu der entsprechenden Firma verweist. *ID\_firma*, *ID\_homepage* und *ID\_news* sind also konsequenterweise auch Topics. Der Übersichtlichkeit halber wird aber darauf verzichtet, sie hier zu definieren, im Anhang 7.1 findet sich das komplette XML-Dokument.

Nun müssen die Topics miteinander verknüpft werden. Hierfür existiert das *assoc* Element:

```

<assoc type="ID_ist_konkurrent">
  <assocr1>ID_amd</assocr1>
  <assocr1>ID_intel</assocr1>
</assoc>

```

Modelliert wird, dass Intel und AMD Konkurrenten sind. Der Typ der Assoziation ist folgerichtig wieder ein Topic. So kann sichergestellt werden, dass AMD und Intel für einen englischsprachigen Zeitgenossen auch als „competitors“ und nicht als „konkurrents“ dargestellt werden, wenn diese Information als *dispname* im Topic *ID\_ist\_konkurrent* mit dem entsprechenden *scope* gespeichert ist. Einen solchen *scope* benutzt auch das etwas komplexere Beispiel der „ist Hersteller“ Relation zwischen Intel und CPU. Sie ist nicht symmetrisch und gilt auch nur in Hinsicht auf PC's, Intel stellt keine Prozessoren für Apple Computer her.

```

<assoc type="ID_ist_hersteller" scope="ID_pc">
  <assocr1 type="ID_hersteller">ID_intel</assocr1>
  <assocr1 type="ID_produk" >ID_cpu</assocr1>
</assoc>

```

Die Asymmetrie der Beziehung kann durch die Verwendung des *type* Attributes der Rolle (*assocr1*) eines Topics innerhalb einer Assoziation formuliert werden. Um den Gültigkeitsbereich einer Assoziation einzuschränken, wird das *scope* Attribut verwendet.

Das letzte noch nicht vorgestellte Konstrukt der Topic Maps ist die Facette. Sie ermöglicht es, Informationsressourcen Eigenschaft/Wert-Paare zuzuweisen:

```

<facet type="ID_sprache">
  <fvalue type="ID_deutsch">
    http://www.heise.de/newsticker/data/jow-11.12.00-000/
  </fvalue>
</facet>

```

## 2 Topic Maps und Companion - Schnittstellen und Einsatzgebiete

---

Companion lag bereits vor Beginn dieser Studienarbeit ein Konzept zur Speicherung des Lehrstoffes vor. Da es nicht darum gehen kann, für das laufende Projekt eine neue Form der Wissensrepräsentation zu erarbeiten, wird im Folgenden auf Aspekte eingegangen, die einen *Zusatznutzen* für Companion bedeuten könnten. Dieses Kapitel beschreibt, in wie weit es sich lohnt, über das Konzept „Topic Maps“ in diesem Zusammenhang nachzudenken.

### 2.1 *Schnittstellen*

- Companion und Topic Maps dienen beide der Speicherung und der Weitergabe von Wissen
- Beide versuchen durch gezielte Strukturierung des Wissens dieses leichter und schneller zu vermitteln
- Beide verfolgen einen generischen Ansatz (keine Spezialisierung auf bestimmte Themengebiete)
- Die Wissensbasis setzt bei beiden auf SGML auf, damit ist diese sowohl on- wie offlinefähig und sowohl von Menschen als auch von Maschinen zu bearbeiten
- Es handelt sich bei beiden um offene, auf Standards basierenden Systeme, deren erklärtes Ziel der Austausch mit Fremdsystemen ist

### 2.2 *Einsatzgebiete*

Liegt Wissen erst einmal in maschinenlesbarer Form vor und kann die Maschine mit diesem Wissen noch umgehen, dann sind vielfältige sinnvolle Einsatzmöglichkeiten im Bereich von Forschung und Lehre vorstellbar. Topic Maps versetzten Computer in die Lage, mit Wissen umzugehen und eröffnen daher ein großes Feld von Anwendungsmöglichkeiten.

Im folgenden werden drei Konzepte vorgestellt, anhand derer Topic Maps einen zusätzlichen Wert für die integrierte Lernumgebung bieten könnten.

### 2.2.1 Textbasierte Suchfunktion

Normale Suchmaschinen glänzen im Regelfall durch zwei Ergebnismengen: die leere Menge und ein ganzes Web voll Treffer. Während „0 Results“ eine absolute Sackgasse darstellt, besteht im anderen Fall zumindest noch die Möglichkeit, die zahlreichen Ergebnisse in eine Metasuchmaschine zu füttern; das Ergebnis wird hierdurch im Regelfall besser. Nichtsdestotrotz bleibt im Allgemeinen eine Menge von mehr oder weniger toten Links, die „von Hand“ durchsucht werden muss.

Die Fähigkeit, Suchmaschinen mit den richtigen Stichworten zu füttern, hat sich als Kunst erwiesen. Ein anderer, zum Beispiel von Microsofts Office-Paket bekannter Ansatz sieht die Eingabe von natürlichsprachlichen Fragen vor. Eine solche Suchmaschine könnte durch eine Topic Map als Datenbasis erheblich verbessert werden. Fasst man Subjekt und Objekte eines Satzes als Topics und Verben als Assoziationen auf, sieht man sehr schnell, dass sich hier passende Modelle gegenüberstehen. Festmachen kann man dies auch schon an dem kleinen Wörtchen „wo“; ein Wort mit dem so manche Frage anfängt, das aber von Suchmaschinen ignoriert wird. Eine auf Topic Maps basierende Suchmaschine wüsste gleich: Alle Topics meiner Ausgabe sind vom Typ „Ort“. Gegen tote Links wird aber leider auch eine auf Topic Maps basierende Suchmaschine nichts ausrichten können.

### 2.2.2 Navigation durch nichtlineare Wissensfelder

Sehr wenig Wissen ist tatsächlich linear darstellbar. Es beschränkt sich im wesentlichen auf Aufzählungen oder Reihenfolgen, wie etwa die der Buchstaben im Alphabet. Die gebräuchlichste Form der Wissensrepräsentation, das geschriebene oder gesprochene Wort, ist zwar seiner äußeren Form nach linear, durch die Verwendung des Wortes als Substitut für einen bestimmten Gegenstand<sup>2</sup>, auf den im Laufe eines Textes durch die Verwendung dieses Wortes mehrfach Bezug genommen werden kann, und dadurch, dass Thesen selbst ebenfalls im weiterführenden Text als Objekte reflektiert werden können (Chunking), vermag die Sprache selbst hochgradig vernetzte Themengebiete darzustellen. Nachteil bleibt aber sowohl die lineare Form der Reproduktion, als auch des Konsums des Wissens in Gestalt von Text. Diagramme hingegen ermöglichen die Aufnahme von Zusammenhängen mit einem Blick. Statt alle in einem Text beschriebenen Sachverhalte verarbeiten zu

---

<sup>2</sup> Rousseau: Das Bezeichnete und das Bezeichnende

müssen,<sup>3</sup> kann der Informationskonsument direkt auf die verschiedenen Aspekte des ihn interessierenden Objektes zugreifen.<sup>4</sup>

Diese Überlegung gilt aber nur, solange sich der Wissbegierige nur für einen Teil der im Diagramm codierten Information interessiert. Um das dargestellte Sachgebiet vollständig zu erfassen, werden auf jeden Fall  $O(n^2)$  Schritte benötigt.<sup>5</sup>

Topic Maps lassen sich gut als Diagramme darstellen. Eine mögliche Umsetzung stellt der praktische Teil dieser Arbeit vor, der in Kapitel 5 beschrieben ist.

### 2.2.3 Erzeugung von Mind Maps

Mind Maps sind eine bestimmte Art von Diagrammen, bei denen zu einem ausgewählten Thema sämtliche Aspekte und Unter Aspekte baumförmig aus einem Knoten herauswachsen. Sie sind somit der unter 2.2.2 beschriebene Spezialfall, in dem den Betrachter nur ein Ausschnitt aus einem größeren Themengebiet interessiert. Folglich lassen sich Mind Maps auch durch Topic Maps repräsentieren oder aus diesen extrahieren.

---

<sup>3</sup>  $O(n^2)$ , falls jede These mit jeder anderen im Zusammenhang steht und dieser durch den Text wiedergegeben wird

<sup>4</sup>  $O(n)+O(n)=O(n)$ , dabei  $O(n)$  für das Suchen und  $O(n)$  für das Betrachten aller Beziehungen zu anderen Objekten inklusive einer reflexiven Beziehung

<sup>5</sup> Es sind sogar  $O(n^3)$  Schritte, wenn man davon ausgeht, dass jedes Objekt mit jedem anderen Objekt eine Verbindung jedes Typs eingeht.

## 3 Topic Maps im Detail

---

Dieses Kapitel beschreibt den offiziellen Standard „ISO/IEC FCD 13250:1999 - Topic Maps“ des Gremiums „ISO/IEC JTC1/SC34 - Document Description and Processing Languages“. Nach der Beschreibung des Standards selbst im ersten Teil, werden im zweiten Teil dieses Kapitels einige Schwachstellen des Standards diskutiert.

### 3.1 *Der Standard*

In diesem Abschnitt werden alle formalen Konstrukte des Standards aufgegriffen und erläutert. Dabei wurde darauf geachtet, den Standard möglichst präzise wiederzugeben, gegebenenfalls aber durch geeignete Zusatzinformation zu ergänzen. Teilweise werden auch Details des Standards übergangen; die Auswahl richtet sich danach, was in öffentlich verfügbaren Topic Map-Dokumenten tatsächlich verwendet, oder in öffentlichen Foren intensiv diskutiert wurde. Hierdurch soll sichergestellt werden, dass im Folgenden nur die wirklich relevanten Konzepte in gebührendem Maße erörtert werden. Konstruktionen, die in der Fachwelt auf geringeres Interesse stießen, werden hier auch kürzer abgehandelt.

#### 3.1.1 **Topic Map Architectural Form (topicmap)**

An der Wurzel einer standardkonformen Topic Map steht das `topicmap` Element. Es ist vom `document` Elementtyp der HyTime Architektur abgeleitet und fasst alle zu einer Topic Map gehörenden Einheiten zu einem Einstiegspunkt zusammen.

Ein Topic Map Element ist folgendermaßen aufgebaut:

topicmap Attribute		
Attributname	Typ	Inhalt, Zitate aus dem Standard
NAME	HyTime Element	“This attribute definition is redundant; it appears here as an aid to understanding.”
addthms	CDATA, #IMPLIED, wenn vorhanden, Referenz zu mindestens einem topic Element	“Themes to add to all scopes that govern the assignments of topic names, occurrences, and roles played in associations in this topic map document.” “NOTE 18: The <b>addthms</b> attribute can be used to acknowledge and document the fact that the document specifies only topic characteristic assignments that are within the scope defined by the set of themes that it specifies. It can be used to avoid specifying these common themes explicitly in every scope. After a topic map document is merged with other topic map documents, the contributions that it made to the resulting merged topic map can be distinguished from the contributions of all others by virtue of the fact that everything it contributed continues to appear within the scopes of the topics specified by the <b>addthms</b> attribute of its document element.” “Default: No themes added via this attribute.”

topicmap Unterelemente	
Elementname	Inhalt
topic	Topic Link, 0..*
assoc	Association Link, 0..*
facet	Facet Link, 0..*
addthms	Themes To Be Added, 0..*
TMBrid	ANY, Topic Map Bridge Element, 0..*

Ein Topic Map Element braucht keines der möglichen Unterelemente enthalten. Somit können sich auch Anwendungen standardkonform nennen, die z.B. nur **facet** Elemente unterstützen oder diese vollständig ignorieren.<sup>6</sup>

### 3.1.2 Topic Link

Das Topic ist das zentrale Konstrukt der Topic Maps. Es vereinigt alle Namen (Synonyme) und Informationseinheiten, die zu einem Thema gehören, in sich. Ein Topic entspricht genau einem *Gegenstand der Betrachtung*. Dieser Gegenstand der Betrachtung wird durch den Autor der Topic Map in dieser definiert. Er kann, muss aber nicht, einem realen Gegenstand entsprechen. Auch braucht der Gegenstand der Betrachtung nirgendwo explizit definiert worden sein. Als Topics sind beispielsweise „Baum“, „Luft“ oder abstrakte Dinge wie „Gesundheit“ denkbar. Auch Konzepte wie „Objektorientierung“ können Topics sein. Offensichtlich haben die Entwickler der Topic Maps mit dem „Topic“ keine besonders neue Idee entwi-

---

<sup>6</sup> Dieses wird in NOTE 19 des Standards beschrieben.

ckelt. Parallelen zu der „Entity“ der E-R-Modelle<sup>7</sup> und der UML-Schemata oder dem Objektbegriff objektorientierter Programmiersprachen wie Java sind unübersehbar. Durch die Generizität des Topic-Konstruktes wird die Anwendbarkeit der Architektur für möglichst viele Gebiete garantiert und ein orthogonaler Aufsatz auf bestehende Wissensbasen ermöglicht.

### 3.1.2.1 *Topic Link Architectural Form (topic)*

Ein Topic Link fasst keine, einen oder mehrere Namen und keine, eine oder mehrere externe Informationseinheiten, sogenannte occurrences (deutsch: Vorkommen) in sich zusammen. Diese Namen und Vorkommen beziehungsweise „Topic-Charakteristika“ sind nicht notwendigerweise kontextfrei; über das **scope** Attribut lässt sich die Gültigkeit einschränken. Es beinhaltet eine Menge von Topics, die einen Gültigkeitsbereich beschreiben. Zu beachten ist, dass die Namens- und Vorkommnis-Elemente (**topname** und **occurs**) selbst ebenfalls **scope** Attribute besitzen dürfen. Falls diese tatsächlich auftreten, werden vor der semantischen Auswertung die beiden Topic-Mengen vereinigt: Es handelt sich bei dem **scope** Attribut des Topics um eine optionale Möglichkeit, syntaktische Redundanz zu vermeiden. Ein im Topic definiertes **scope** Attribut gilt für jedes seiner Namens- und Vorkommnis-Elemente, könnte also gleichwertig bei jedem einzelnen Kind-Element definiert werden.<sup>8</sup> Es sei noch einmal explizit darauf verwiesen, dass die Angabe eines Gültigkeitsbereiches nicht die Gültigkeit des Topics, sondern nur die seiner Namens- und Vorkommnis-Elemente einschränkt.<sup>9</sup>

Das **types** Attribut definiert eine „ist-ein“-Beziehung (instance of) zu einem anderen Topic. Diese ist von einer Unterklasse-Oberklasse-Beziehung zu unterscheiden. Eine solche lässt sich in Topic Maps durch Assoziationen modellieren.

---

<sup>7</sup> E-R-Modell steht für „Entity-Relationship-Modell“ (deutsch: Gegenstands-Beziehungs-Modell), das vor etwa 20 Jahren in der Datenbank-Welt entstand, um auf einer Meta-Ebene für die damals vorherrschenden Datenmodelle (u.a. relationales Modell) entwickeln zu können. Es wurde später auch in die Bereiche Systemanalyse und Unternehmensmodellierung übernommen.

<sup>8</sup> Siehe hierzu NOTE 24 des Standards

<sup>9</sup> Ist der Scope bei allen Unterelementen des Topics gleichermaßen eingeschränkt, entspricht dies de facto einer Einschränkung der Gültigkeit des Topics. Dies gilt allerdings nur, so lange keine „fremden“ Topics zu der jeweiligen Topic Map beigemischt werden. Außerhalb des ursprünglichen Scopes ist das entsprechende Topic dann nämlich nicht mehr notwendigerweise unsichtbar (und quasi inexistent).

topic Attribute		
Attributname	Typ	Inhalt, Zitate aus dem Standard
ID	ID, #REQUIRED	“The required <b>unique identifier</b> ( <b>id</b> ) attribute facilitates the addressing of topics by association links, by the <b>identity</b> attributes of other topic links, and, in their roles as themes in scopes, by <b>scope</b> and <b>addthems</b> attributes.”
identity	CDATA, #IMPLIED	“Reference to information (one or more subject descriptors) that confers understanding of the identity of the subject of this topic link.” “NOTE 20: The information referenced by an <b>identity</b> attribute may or may not take the form of a topic link in a topic map document, may or may not be text, may or may not be machine-interpretable, and may or may not be online.” “Default: No subject descriptors; the subject must be <b>inferred from the topic's characteristics</b> .”
types	CDATA, #IMPLIED, wenn vorhanden, Referenz zu mindestens einem <b>topic</b> Element	“Topics whose subjects are the classes of topics of which this topic is an instance.” “Default: No class-instance topic associations are established via this attribute. Note: Some might still be specified by topic association links, however.”
scope	CDATA, #IMPLIED, wenn vorhanden, Referenz zu mindestens einem <b>topic</b> Element	“The themes that are added to the scopes of all the names and occurrences specified by this topic link.” “Default: No themes are added by this attribute.”

topic Unterelemente	
Elementname	Inhalt
topname	Topic Name, 0..*
occurs	Topic Occurrence, 0..*

Das **identity** Attribut ermöglicht eine eindeutige Kennzeichnung eines Topics. Haben zwei Topics das gleiche **identity** Attribut so sind sie die selben, auch wenn sie völlig unterschiedliche Namen zugewiesen bekommen haben. Dieses ermöglicht das Mischen zweier Topic Maps unterschiedlichster Herkunft, wenn sich diese bezüglich des **identity** Attributes an eine gemeinsame Namenskonvention halten. Im Optimalfall haben beide Autoren vorher bereits eine Liste aller möglichen Identitäten - dann würden sie sich beide optimal ergänzen können. Die Existenz einer solchen Liste setzt jedoch voraus, dass von vornherein jedes mögliche Topic bekannt ist; diese Problematik versuchen Bibliothekare aber schon seit Jahren zu bewältigen.<sup>10</sup>

Ein Ansatz, der die Funktionsweise des **identity** Attributes verdeutlicht, ist die Verwendung von Domänensystemen. Das Topic „Topic Maps“ könnte folgendes Attribut erhalten:

`de.vikar.companion.begriffe.konzepte.topicmaps`

---

<sup>10</sup> Stichwort „Schlagwortkatalog“

### 3.1.2.2 Topic Name Architectural Form (topname)

Ein **topname** Element modelliert ein Synonym zu einem Topic. Zu einem Topic dürfen beliebig viele solcher Synonyme definiert werden. Ein solches Synonym ist ein Name für ein Topic, unter Umständen mit mehreren Schreibweisen. So könnte beispielsweise ein Topic, das die Person Peter Müller modelliert, die Namen „Peter Müller“, „Müller, Peter“, „Peter Mueller“ und „Mueller, Peter“ tragen. Vielleicht hat Herr Müller aber auch noch ein Login und einen Spitznamen. Dann wären „muellerp“ und, weil der Mann ein ziemlicher Weihnachtsmuffel ist, „The Grinch“ ebenfalls Namen, die diesem Topic zugewiesen werden müssten. Während man gestrost sämtliche Formen von „Peter Müller“ zu einem **topname** Element zusammenfassen würde, sollte man für „muellerp“ und „The Grinch“ eher jeweils ein zusätzliches **topname** Element innerhalb des Topics definieren.

Zur Strukturierung der „Peter-Müller-Menge“ bieten Topic Maps darüber hinaus weitere Konstrukte an. Mit den Unterelementen **basename**, **dispname** und **sortname** kann festgelegt werden, ob sich Peter Müller hinter „Ralf Meier“ oder hinter „Paul Nuss“ finden lässt und ob er als „Peter Müller“ oder als „Mueller, Peter“ angezeigt wird. Mit dem **scope** Attribut kann die Verwendbarkeit eines Namens auf bestimmte Zusammenhänge beschränkt werden. So gibt es für den Namen „The Grinch“ in Peter Müllers Geschäftsleben sicher keine Verwendung.

Ein **topname** Element, das übrigens als besondere Form des Vorkommens (occurrence) betrachtet werden kann, besteht aus mindestens einem **basename** Element und keinem, einem oder beliebig vielen **dispname** Elementen, sowie keinem, einem oder beliebig vielen **sortname** Elementen. Sind keine **dispname** oder **sortname** Elemente definiert, so werden die **basename** Elemente zum Anzeigen und Sortieren verwendet.

<b>topname Attribute</b>		
<b>Attributname</b>	<b>Typ</b>	<b>Inhalt, Zitate aus dem Standard</b>
<b>scope</b>	CDATA, #IMPLIED, wenn vorhanden, Referenz zu mindestens einem <b>topic</b> Element	“Reference to a set of themes (topic links) to be added to the scopes of the name characteristics specified by the contained <b>basename</b> , <b>dispname</b> , and <b>sortname</b> elements. Scopes are sets of themes that collectively define the limited context within which characteristics are validly applicable to the topic.” “Default: No themes are added via this attribute.”

<b>topname Unterelemente</b>	
<b>Elementname</b>	<b>Inhalt</b>
basename	Basisname, 1..*
dispname	Display-Name, 0..*
sortname	Sortiername, 0..*

Das optionale **dispname** Element spezifiziert den Inhalt, den die Topic Map Applikation dem Benutzer anzeigen soll. Ist es innerhalb eines **topname** Elementes angegeben, so soll es auch anstelle eines Basisnamen zu Anzeigezwecken verwendet werden. Sein Inhalt ist entweder reiner Text oder eine „Notation“ im Sinne des HyTime Standards (notation common attribute). Eine solche Notation kann zum Beispiel eine Grafik sein, oder auch andere Daten, die dem Benutzer den Gegenstand des Topics verdeutlichen.

Der Inhalt des optionalen **sortname** Elementes wiederum darf ebenso wie das **basename** Element nur reine Textbausteine (Wörter oder Phrasen) enthalten und dient der Topic Map Applikation zur Identifizierung des Namens, anhand dessen das zugehörige **topname** Element, und somit auch das zugehörige Topic, in die Liste aller Topics einsortiert werden soll. Ist ein solches Element definiert, so ist es zu Sortierzwecken anstelle der im gleichen **topname** Element spezifizierten Basisnamen zu verwenden.

<b>basename/dispname/sortname Attribute</b>		
<b>Attributname</b>	<b>Typ</b>	<b>Inhalt, Zitate aus dem Standard</b>
scope	CDATA, #IMPLIED, wenn vorhanden, Referenz zu mindestens einem topic Element	“References to a set of themes (topic links) to be added to the scope of the name characteristic specified in the content.” “Default: No themes are added via this attribute.”

<b>basename/dispname/sortname Inhalt</b>	
<b>Typ</b>	<b>Beschreibung</b>
#PCDATA	„String to be used as name“ bzw. „String (or notation data) to be displayed as name“

Haben zwei Topics die gleiche Namenscharakteristik, das heißt, sind alle Ihre Namen gleich und haben diese auch jeweils den gleichen Wirkungsbereich (scope), so sollen diese Topics nach dem Willen des Standards als identisch betrachtet und zusammengeführt werden.<sup>11</sup> Dieser Hinweis ist besonders für das Mischen zweier verschiedener Topic Maps interessant:

---

<sup>11</sup> „Topic Naming Constraint“, siehe hierzu auch NOTE 31, NOTE 32, NOTE 33, NOTE 34, NOTE 35 und NOTE 36 des Standards

*„[...] each distinct set of themes that serves as a scope constitutes a namespace in which no two subjects can have the same name. If a conforming topic map application detects a situation in which multiple topic links have the same name characteristic within the same scope, they shall be merged.“ (Standard)*

### 3.1.2.3 Topic Occurrence Architectural Form (occurs)

Das **occurs** Element stellt die Verbindung zwischen den Strukturen der Topic Maps und externen Informationseinheiten her. Topic Maps selbst speichern Meta-informationen über die Zusammenhänge zwischen einzelnen Wissensschnipseln; um den Benutzer zu diesen Schnipseln leiten zu können, wird die Information über den Zugang zu diesen Schnipseln in den **occurs** Elementen abgelegt. Für gewöhnlich dürfte der Inhalt ein URL darstellen. Typ und Adressierungsart der Informationsressource sind durch den Standard aber in keiner Weise eingeschränkt<sup>12</sup>. Sinnigerweise enthält die Quelle aber auch tatsächlich verwertbare Informationen, die mit dem eigentlichen Topic in, für den Benutzer nachvollziehbarer, direktem Zusammenhang stehen.

Das **occr1** (occurrence role) Attribut dient wie auch das **type** Attribut zur Charakterisierung der adressierten Information. Mit Hilfe dieser beiden lassen sich in Topic Map Applikationen Filter- und Gruppierungsfunktionen schaffen. Der Unterschied ist, dass das **occr1** Attribut ein einzelnes Token aufnimmt, während das **type** Attribut auf ein Topic verweist. An den Inhalt des **occr1** Attributes werden keine besonderen Anforderungen gestellt, dafür beschränkt sich seine Funktionalität innerhalb einer Topic Map auf rein deskriptive Inhalte und innerhalb einer Applikation auf pure String-Vergleiche. Dem gegenüber gestattet die Verwendung des **type** Attributes durch die Möglichkeit des Zugriff auf das zugehörnde Topic, erheblich höhere Flexibilität, zum Beispiel im Bezug auf Mehrsprachfähigkeit der Beschreibung der Informationsressource für den Benutzer. Die Verwendung des **type** Attributes sollte also der Verwendung des **occr1** Attributes vorgezogen werden. So empfiehlt der Standard auch, dass Applikationen dem Benutzer den Inhalt des **occr1** Attributes möglichst verbergen sollten, wenn der Inhalt des **type** Attributes innerhalb des aktuellen Gültigkeitskontextes des Benutzers die Anzeige eines Namens zulässt.

Das **scope** Attribut dient zur Einschränkung der Gültigkeit des Bezugs zur Informationsressource auf einen bestimmten Zusammenhang.

---

<sup>12</sup> NOTE 37 räumt Topic Map konformen Applikationen die Möglichkeit ein, Einschränkungen vorzunehmen. Insbesondere, so NOTE 38, dürfen hier aber auch Offline-Ressourcen adressiert werden.

occurs Attribute		
Attributname	Typ	Inhalt, Zitate aus dem Standard
scope	CDATA, #IMPLIED, wenn vorhanden, Referenz zu mindestens einem topic Element	“Reference to themes that are added to the scope within which the occurrences are applicable to the topic characterized by the containing topic link.” “Default: No themes are added to the scope by means of this attribute.”
occr1	NAME, #IMPLIED	“...can be used to provide a mnemonic name for the occurrence role.” “Note: Not displayed for the topic map user if the topic referenced by the type attribute has displayable characteristics within the user's scope.” “Default: GI of element is treated as occurrence role name.”
type	CDATA, #IMPLIED, wenn vorhanden, Referenz zu genau 1 topic Element	“Reference to the topic that names and/or otherwise characterizes the occurrence role. The characteristics of the referenced topic, if appropriate, will be displayed to the user instead of the value of the occr1 attribute.” “Default: No topic characterizes the occurrence role, unless this element is an occurrence (with an occurrence role whose meaning is instance) of a topic whose subject is the nature of the occurrence role. The value of the occr1 attribute will be displayed as the occurrence role name.”

occurs Inhalt	
Typ	Beschreibung
%loc;	Locator, 0..*

### 3.1.3 Association Link

Die Assoziation ist für Topic Maps ebenso bedeutend wie die sinnvolle Auswahl der Topics: Sie verbindet mehrere Topics miteinander und ermöglicht so die Navigation durch das modellierte Wissensgebiet. Die dargestellten Beziehungen lassen sich mittels Angabe des Typs dieser Beziehung sowie der Beschreibung der Rolle, die ein Topic innerhalb dieser Beziehung spielt, mit zusätzlicher Semantik versehen. Die Leistungsfähigkeit einer Topic Map hängt wesentlich von der Anzahl und der Güte der verwendeten Assoziationen ab.

#### 3.1.3.1 Association Link Architectural Form (assoc)

Eine Assoziation verbindet mindestens ein, im Regelfall aber mehrere Topics. Diese spielen im Bezug auf diese Assoziation eine bestimmte Rolle. Diese Rolle wird durch das `assocr1` Element modelliert; das `assoc` Element ist also ein Container für die Rollen. Nur ein Topic in einer Assoziation verwenden zu können, schafft die Möglichkeit, das Topic aussagenlogisch zu attributieren<sup>13</sup>.

<sup>13</sup> In der Prädikatenlogik betrachtet man Beziehungen mit nur einem Mitglied als Attribute; man spricht von einer Beziehung mit dem „Grad 1“. Es handelt sich hierbei um

Das optionale **scope** Attribut spezifiziert den Wirkungsbereich, innerhalb dessen die Verknüpfung des oder der Topic(s) gültig ist. Die Attribute **type** und **linktype** dienen zur Charakterisierung der Assoziation und funktionieren in etwa wie die Attribute **type** und **occur1**<sup>14</sup> des **occurs** Elementes (siehe Abschnitt 3.1.2.3): **linktype** ist nur als mnemonisches Token zu verstehen, während das **type** Attribut mit seiner Referenz auf ein Topic eine große Flexibilität und Funktionsvielfalt eröffnet<sup>15</sup>.

assoc Attribute		
Attributname	Typ	Inhalt, Zitate aus dem Standard
scope	CDATA, #IMPLIED, wenn vorhanden, Referenz zu mindestens einem topic Element	“Reference to themes that are added to the scope within which the association is applicable.” “Default: Scope is unconstrained.”
linktype	NAME, #IMPLIED	“Mnemonic name for the association type.” “Note: Not displayed for the topic map user if the topic referenced by the type attribute has displayable characteristics within the user's scope.” “Default: Generic identifier”
type	CDATA, #IMPLIED, wenn vorhanden, Referenz zu genau 1 topic Element	“Topic whose subject is the class of association of which this association is an instance.” “Default: No type is specified by this attribute.” “Note: A type might exist by virtue of the fact that this association link is an occurrence (where the occurrence role means "instance") of a topic whose subject is the nature of the association, however.”

assoc Unterelemente	
Elementname	Inhalt
assocr1	Rolle, 1..*

### 3.1.3.2 Association Role Architectural Form (assocr1)

Das **assocr1** Element dient zur Kennzeichnung der Rolle, die ein Topic innerhalb einer Assoziation spielt. Mittels des **type** Attributes wird eine Klasse-Instanz-Beziehung zwischen dem durch dieses Attribut referenzierten Topic und der durch das **assocr1** Element ausgeprägten Rolle erstellt. Diese Beziehung ließe sich auch ausdrücken, indem man das **assocr1** Element als Vorkommnis (occurrence) des referenzierten Topics auffasst, wobei der Typ des **occurs** Elementes (**occr1** oder

---

binäre Attribute („rot“ kann nicht modelliert werden, nur „ist\_rot“). Gibt es zu einem Topic eine solche Assoziation, so besitzt es die besagte Eigenschaft, anderenfalls nicht. Die Aussage bezieht sich allerdings immer nur auf die Domäne der aktuellen Topic Map.

<sup>14</sup> Achtung: Das **occur1** Attribut des **occurs** Elementes entspricht dem **linktype** Attribut des **assoc** Elementes und nicht dessen Unterelement **assocr1**. Die Namen sind hier äußerst unglücklich gewählt!

<sup>15</sup> Dies wird in NOTE 42 beschrieben.

type Attribut) kennzeichnen muss, dass es sich bei dem Element um eine Instanz der durch das Topic beschriebenen Klasse handelt.

Der Unterschied zwischen dem `anchrole` Attribut und dem `type` Attribut entspricht dem zwischen `linktype` und `type` (siehe Abschnitt 3.1.3.1) beziehungsweise dem zwischen `occur1` und `type` (siehe Abschnitt 3.1.2.3); auf die unglückliche Namenswahl des Standardisierungskomitees sei an dieser Stelle nochmals verwiesen (siehe Fußnote 14).

assocrl Attribute		
Attributname	Typ	Inhalt, Zitate aus dem Standard
<code>anchrole</code>	NAME, #IMPLIED	“Note: Not displayed for the topic map user if the topic referenced by the type attribute has displayable characteristics within the user’s scope.” “Default: GI of element is treated as anchor role.”
<code>type</code>	CDATA, #IMPLIED, wenn vorhanden, Referenz zu genau 1 topic Element	“Reference to the topic that names and/or otherwise characterizes the association role. The characteristics of the referenced topic, if appropriate, will be displayed to the user instead of the value of the <code>anchrole</code> attribute.” “Default: No topic characterizes the association role, unless this element is an occurrence (with an occurrence role whose meaning is instance) of a topic whose subject is the nature of the association role. The value of the <code>anchrole</code> attribute will be displayed as the association role name.”

assocrl Inhalt	
Typ	Beschreibung
%loc;	Referenz zu mindestens einem topic Element

### 3.1.4 Facet Linking

Facetten dienen dazu, in Topic Maps Metainformationen zu externen Informationsquellen zu speichern. Dies kann sinnvoll sein, um zum Beispiel zu einem bestimmten, referenzierten Fachbuch angeben zu können, dass es in englischer Sprache verfasst ist. Diese Information hat, aller Wahrscheinlichkeit nach, nichts mit dem modellierten Themengebiet zu tun; eine Speicherung und Verarbeitung dieses Attributes erscheint dennoch sinnvoll. Zu diesem Zweck hat das Standardisierungskomitee die Facette eingeführt.

#### 3.1.4.1 Facet Link Architectural Form (facet)

Das `facet` Element wird benutzt, um Topic Map-externen Objekten Eigenschaft-Wert-Paare zuzuweisen. Diese Objekte werden in einem oder mehreren `fvalue` Subelementen spezifiziert. Das `facet` Element dient somit als Container.

Die zu definierende Eigenschaft des Eigenschaft/Wert-Paares wird durch sein **type** Attribut beschrieben. Das selbe leistet auch das **linktype** Attribut, jedoch ist in bestimmten Fällen die Verwendung des **type** Attributes vorzuziehen<sup>16</sup>. Die Funktionsweise ist also ähnlich der beim **assoc** Element (siehe 3.1.3.1), mit dem wesentlichen Unterschied, dass das **type** Attribut (bzw. das **linktype** Attribut) im Falle des **facet** Elementes nicht nur eine nützliche Zusatzinformation, sondern den essenziellen Inhalt dieses Elementes beschreibt.<sup>17</sup>

facet Attribute		
Attributname	Typ	Inhalt, Zitate aus dem Standard
linktype	NAME, #IMPLIED	"Mnemonic name for the property (facet type)." "Note: Not displayed for the topic map user if the topic referenced by the type attribute has displayable characteristics within the user's scope." "Default: Generic identifier"
type	CDATA, #IMPLIED, wenn vorhanden, Referenz zu genau 1 <b>topic</b> Element	"Topic whose subject is the property of the property/value pair(s) being assigned to the anchor(s)." "Note: A facet type topic might exist by virtue of the fact that this facet link is an occurrence (where the occurrence role means "instance") of a topic whose subject is the nature of the property, however." "Default: No facet type topic is specified by this attribute."

facet Unterelemente	
Elementname	Inhalt
fvalue	Facettenwerte, 1..*

#### 3.1.4.2 Facet Value Architectural Form (fvalue)

Die als Unterelemente des **facet** Elementes (siehe 3.1.4.1) verwendeten **fvalue** Elemente weisen dem referenzierten Objekt einen benutzerdefinierbaren Wert zu. Dabei enthält das **facetval** Attribut den Wert des Eigenschaft/Wert-Paares, **type** beschreibt die Bedeutung dieses Wertes. Die Klasse-Instanz-Beziehung zwischen dem Wert-Objekt und dem durch sein **type** Attribut benannten Topic kann nach Aussage des Standards auch ausgedrückt werden, indem man das **fvalue** Element als Vorkommnis (occurrence) des verknüpften Topics auffasst. Das **type** Attribut des **occurs** Elementes muss dann auf den Instanzcharakter des angege-

<sup>16</sup> Siehe NOTE 51

<sup>17</sup> Das **type** Attribut kann natürlich auch in allen anderen Anwendungsfällen bedeutende Informationen enthalten; jedoch mag man mit der Information, dass die Topics „Auto“ und „Straße“ assoziiert sind, auch ohne genauere Angabe des Typs der Assoziation etwas anfangen können. Wird aber die Eigenschaft nicht angegeben, der ein Wert zugewiesen wird, wird man nur im Ausnahmefall (Beispiel „rot“) über den Typ des Wertes auf die gewählte Eigenschaft schließen können.

benen `fvalue` Elementes hinweisen. Die gleiche Konstruktion lässt sich übrigens auch für das `type` Attribut des `facet` Elementes erstellen.

<b>fvalue Attribute</b>		
<b>Attributname</b>	<b>Typ</b>	<b>Inhalt, Zitate aus dem Standard</b>
<code>facetval</code>	NAME, #IMPLIED	"Facet value name" "Token is value of property being assigned." "Default: Facet value name is GI of element."
<code>type</code>	CDATA, #IMPLIED, wenn vorhanden, Referenz zu genau 1 <code>topic</code> Element	"Reference to a topic whose subject is the significance of the facet value name." "Note: A facet value type topic might exist by virtue of the fact that this <code>fvalue</code> element is an occurrence (where the occurrence role means "instance") of a topic whose subject is the significance of the facet value name, however." "Default: No facet value type topic is specified by this attribute."

<b>fvalue Inhalt</b>	
<b>Typ</b>	<b>Beschreibung</b>
<code>%loc;</code>	Locator, nicht näher spezifiziert, Adresse des/der Elemente(s), für das/die dieser Wert gilt, 0..*

### 3.1.5 Themes To Be Added (`addthms`)

Das `addthms` Element dient dazu, für eine Topic Map eine Art Domain festzulegen. Nach dem Topic Naming Constraint<sup>18</sup> sollen zwei Topics, die die gleiche Namenscharakteristik haben, als identisch betrachtet und zusammengeführt werden. Ein wesentlicher Bestandteil der Charakteristik ist neben dem Name des Topic auch der jeweilige Scope. Durch das `addthms` Element wird nun jeder Scope der Topic Map um die in ihm angegebenen Elemente erweitert. Folglich können beim Mischen von Topic Maps zwei Topics disjunkt bleiben, die identische Namen sowie Scope-Definitionen enthalten, sofern die jeweiligen Quell-Maps unterschiedliche `addthms` Elemente besaßen.<sup>19</sup> Eine Anwendung hierfür wären beispielsweise zwei Topic Maps, die die unterschiedlichen Ansichten zweier Philosophen wiedergeben. Haben beide unterschiedliche `addthms` Elemente angegeben, so können beide Ansichten in einer Topic Map vereinigt werden, ohne dass sich die beiden Philosophen jemals auf einen gemeinsamen Standpunkt einigen müssen.

<sup>18</sup> Siehe Abschnitt 3.1.2.2 bzw. NOTE 31 des Standards

<sup>19</sup> Siehe NOTE 36. Ist allerdings das `identity` Attribut gleich, so müssen sie wiederum vereinigt werden.

<b>addthms Attribute</b>		
<b>Attributname</b>	<b>Typ</b>	<b>Inhalt, Zitate aus dem Standard</b>
addthems	CDATA, #REQUIRED, Referenz zu mindestens einem <b>topic</b> Element	“Themes to be added to the scopes specified by the <b>tmdocs</b> and <b>cassign</b> attributes“
tmdocs	ENTITIES, #IMPLIED	“Constraint: Must be one or more document entities of topic map documents.”
cassign	CDATA, #IMPLIED, wenn vorhanden, Referenz zu mindestens einem <b>topic</b> , <b>topname</b> , <b>basename</b> , <b>dispname</b> , <b>sortname</b> , <b>occurs</b> , und/oder <b>assoc</b> Element	“Elements that assign characteristics to topics. The themes specified by the <b>addthms</b> attribute are to be added to the scopes within which the characteristics they specify are regarded as valid”

<b>addthms Unterelemente</b>
“NOTE 48: The <b>addthms</b> element's content is not defined by the Topic Maps architecture.”

## 3.2 *Kommentar*

An dieser Stelle wird auf einige Konzepte im Standard verwiesen, die dieser ungenau spezifiziert, sowie auf Schwachstellen in seiner Umsetzung. Selbst ein mehrmonatiges Verfolgen der offiziellen Topic Map Mailingliste von InfoLoom, Inc. (USA/Frankreich <http://www.infoloom.com>), an der neben aktuellen Größen aus dem Bereich der Topic Maps auch einige Editoren des Standards teilnehmen, konnte nicht alle Probleme klären. Ziel dieses Abschnittes ist es, dem Leser zu verdeutlichen, an welchen Ecken er sich stoßen könnte, sollte er sich für die Arbeit mit Topic Maps entscheiden.

### 3.2.1 **Scopes**

Das wesentliche Konzept des Scope dient zur Eingrenzung der dem Benutzer zurückzuliefernden Informationsmenge. Um ihm eine sinnvolle Auswahl von für ihn relevanten Quellen an die Hand geben zu können, muss die Resultatmenge einer Suche durch geeignete Filter reduziert werden. Der Standard legt nicht eindeutig genug fest, wie mit Scopes umzugehen ist. Steven R. Newcomb, einer der Editoren des Standards schreibt<sup>20</sup>:

*“A scope is a set of topics (“themes”), pure and simple. The ways in which applications treat these sets is not constrained by the standard. Applications*

---

<sup>20</sup> Steven R. Newcomb (srn@techno.com), gesendet Fr 17.03.2000 an die InfoLoom Topic Map Mailingliste, Betreff: “Re: scopes and themes (verification of my understanding)”

*may regard various kinds of operations on these sets (unions, intersections, etc.) as significant in whatever ways they may regard them as significant.*

[...]

*In terms of a specific scope in a specific topic map, however, it may be meaningful to look at it as an intersection, as a union, as both, and as neither. Therefore, the question of whether a scope should be regarded as "the union of the subjects of the themes (topics)" or as "the intersection of the subjects of the themes" is a matter for applications to decide."*

Das bedeutet nun für Topic Maps, die von verschiedenen Anwendungen erstellt worden sind, dass sie semantisch nicht kompatibel sein könnten, da Schnitte und Vereinigungen unterschiedliche Erfüllungsmengen besitzen. Ein Mischen von beliebigen Topic Maps, wie es im Standard angedeutet wird,<sup>21</sup> ist also nicht zulässig. Somit müßte theoretisch jede Applikation für jede von ihr erstellte Topic Map mittels des `addthms` Elementes (siehe 3.1.5) eine Domäne festlegen. Zusammengeführt können dann nur Topics werden, die das gleiche `identity` Attribut (siehe 3.1.2.1) besitzen.

Dr. Hans Holger Rath, Consulting Direktor der STEP Electronic Publishing Solutions GmbH und Autor des Artikels "Mozart oder Kugel" in der iX, selbst Mitglied im ISO Gremium JTC1 SC34, das für die Standardisierung von Topic Maps zuständig ist, meint<sup>22</sup>:

*"Either the standard states what the logical relation is (and we have the text in ISO/IEC 13250) and we have to follow it, or it is a bug and we have to fix it."*

### **3.2.2 Filter: Scope versus Facette versus Assoziation**

Scope und Facette haben eines gemeinsam: mit ihrer Hilfe können Objekte unäre Attribute zugewiesen werden. Ist das Attribut aufgeführt, so erfüllen sie die entsprechende Eigenschaft, fehlt es, erfüllen sie diese nicht. Der Unterschied zwischen den beiden besteht darin, dass Scopes für praktisch alle Attributzuweisungen

---

<sup>21</sup> Zitat: "This International Standard does not permit two distinct subjects to have the same name characteristic within exactly the same scope (the topic naming constraint). When topic maps are processed, each distinct set of themes that serves as a scope constitutes a namespace in which no two subjects can have the same name. If a conforming topic map application detects a situation in which multiple topic links have the same name characteristic within the same scope, they shall be merged." und NOTE 32 bis 36

<sup>22</sup> Hans Holger Rath (hhr@step.de) / gesendet Mo 20.03.2000 an die InfoLoom Topic Map Mailingliste / Betreff: "Re: scopes and themes (verification of my understanding)"

innerhalb einer Topic Map verwendet werden, während sich Facetten auf externe Ressourcen beziehen. Auch ihre Anwendung ist ungleich: Scopes sind Attribute in den jeweiligen Tags, Facetten werden selbst als Tag angegeben. Zu allem Überflus kann im `occurs` Tag (3.1.2.3) neben dem eigentlichen Link, der an anderer Stelle durch eine Facette attribuiert sein kann, auch noch ein `scope` Attribut angegeben werden. Dies führt dazu, dass Information nicht nur redundant, sondern prinzipbeding auch widersprüchlich gespeichert werden kann.

Eine dritte Möglichkeit, einem Objekt eine Eigenschaft zuzuweisen, besteht darin, dieses Objekt mittels einer Assoziation mit dem Objekt zu verbinden, das die Eigenschaft beschreibt. Das Attribut `type` des `assocr1` Elementes (3.1.3.2) ermöglicht eine eindeutige Klassifizierung nach Objekt und Eigenschaft. Der Vorteil dieser Lösung ist, dass die Verbindung der Konzeption entsprechend in zwei Richtungen verfolgt werden kann. Eine Topic Map weiß somit nicht nur „der Fußball ist rund“, sondern auch, dass ein Beispiel für eine rundes Objekt der Fußball ist.

Dieser Weg ist aber als inoffizielle Alternative zu den im Standard vorgesehenen Scope und Facette zu betrachten. In der Leistungsfähigkeit sicher nicht geringer, setzt er aber eine andere Filterlogik voraus, da nicht nur die einfache Existenz eines Attributes als Entscheidungskriterium gelten kann.<sup>23</sup> Vielmehr muss zunächst das Topic gesucht werden, das das Attribut formuliert. Danach werden all dessen Assoziationen darauf hin untersucht, ob sie dieses Topic einem anderen Topic positiv<sup>24</sup> als Attribut zuweisen oder eine andere Beziehung mit ihm eingehen.

Ein solcher Filter ist nicht unbedingt langsamer, stellt die angegebene Modellierung ja quasi einen Index dar.<sup>25</sup> Das Verfolgen und Auswerten von Assoziationen muss, um den vollen Leistungsumfang der Topic Maps ausschöpfen zu können, auf jeden Fall implementiert werden. Für Topic Map-verarbeitende Systeme scheint also, insbesondere auch in Hinblick auf die Orthogonalität, eine interne Konvertierung von Scopes und Facetten auf Assoziationen sinnvoll.

### 3.2.3 Nichtabzählbare Attribute, unendliche Wertebereiche

Topic Map Objekte können, wie in den beiden vorherigen Abschnitten beschrieben, mittels Scopes und Facetten (oder auch Assoziationen) attribuiert

---

<sup>23</sup> Ansatz: Durchlaufe die Liste aller Objekte und gib diejenigen zurück, die das Attribut enthalten.

<sup>24</sup> Statt einer „ist“ Assoziation könnten die Topics auch eine „ist nicht“ Assoziation eingegangen sein. Dies ist ebenfalls eine Attributzuweisung, aber nicht im gesuchten Sinne.

<sup>25</sup> Auf alle Objekte, die das Attribut besitzen, kann direkt zugegriffen werden. Die Geschwindigkeitssteigerung ist gegenüber der linearen Suche besonders groß, wenn nur vergleichsweise wenige Elemente dieses Attribut besitzen.

werden. Diese Attribute unterliegen aber einer Einschränkung: es können immer nur diskrete Werte<sup>26</sup> zugewiesen werden. Ein Beispiel:

Eine Topic Map modelliert den Buchbestand einer Bibliothek. Da in Deutschland (noch) die Buchpreisbindung gilt, kann zu jedem Buch eindeutig ein Preis angegeben werden. Um dieses Attribut einem Buch zuzuweisen (auf welche Weise ist in diesem Fall egal), muss jeder mögliche Preis als Topic modelliert werden. Im schlimmsten Fall bedeutet dies, die Anzahl der Topics zu verdoppeln.<sup>27</sup>

Statt das Topic zu verwenden, könnte man auf die Idee kommen, sich den jeweiligen mnemonischen Wert mit der gleichen Bedeutung zunutze zu machen. Dies würde man sich aber damit erkaufen, dass der Preis ausschließlich in einer Währung, also entweder „DM“ oder „Euro“ angegeben werden kann; in dieser Zeit der Währungsumstellung wenig attraktiv.<sup>28</sup>

Offensichtlich sind Dinge wie Preise noch nicht effizient mit Topic Maps modellierbar. Ist zu erwarten, dass sich das Problem mit kontinuierlichen Wertebereichen in den Weg stellt, so sollte bereits in der Konzeptfindungsphase darauf Rücksicht genommen werden und man sollte sich darauf einstellen, gebräuchliche Workarounds<sup>29</sup> zu suchen oder gar eigene Konzepte zu entwickeln.

---

<sup>26</sup> Um Missverständnisse zu vermeiden, sei auf den scheinbaren Widerspruch zu der Aussage in Abschnitt 3.2.2 hingewiesen: hier war gar nur von unären Attributen die Rede. Tatsächlich muss zu einem Wert, der einem Objekt als Attribut zugewiesen werden soll, erst ein diskretes (!) Objekt, das einen nicht diskreten Wert beinhalten kann, erzeugt werden. Anschließend wird jedes bestehende Objekt dadurch attributiert, dass man ihm entweder die durch das diskrete Objekt definierte Eigenschaft zuweist oder eben nicht (also binär).

<sup>27</sup> Ausgehend davon, dass nur Bücher als Topics modelliert sind und jedes Buch einen anderen Preis hat als alle anderen Bücher

<sup>28</sup> Vergleiche hierzu Abschnitt 3.1.2.3, Verwendung des `occr1` vs. `type` Attribut

<sup>29</sup> siehe InfoLoom Topic Map Mailingliste

## 4 Erzeugung der Metadaten

---

Topic Maps speichern Objekte und Zusammenhänge zu einem bestimmten Wissensgebiet sowie Referenzen zu externen Informationsressourcen. Der zu beschreibende Ausschnitt aus der realen Welt ist die sogenannte *Miniwelt*. Topic Maps zu erstellen heißt, ein Abbild dieser Miniwelt zu erstellen.

Wieso ist in der Kapitelüberschrift von „Metadaten“ die Rede? Wenn das ganze Wissensgebiet modelliert wird, handelt es sich dann nicht um Nutz- statt um Metadaten?

Die Antwort auf diese Frage hängt nur von der Menge der Daten und der Art ihrer Verwendung ab. Prinzipbedingt muss die Miniwelt mit Hilfe der Topic Map-Konstrukte modelliert werden, um die Funktionalität von Topic Maps nutzen zu können. Demnach wird verwertbares Wissen (also Nutzdaten) in die Topic Map gesteckt. Andererseits, und hier sei noch einmal explizit auf den Bezug dieser Studienarbeit auf Vikar und insbesondere Companion verwiesen, liegen Informations- und Lehrstoffsammlungen im Regelfall andere Konzepte zur Speicherung des Wissens zugrunde als Topic Maps. Folglich wird für solche Pools wahrscheinlich nur ein kompakter Auszug des ihnen innewohnenden Wissens repliziert. Der Wissensdurstige wird seine Informationen also im Wesentlichen weiterhin aus den zugrundeliegenden Dokumenten ziehen, die in der Topic Map gespeicherten Metadaten gestatten ihm aber ein leichteres Auffinden der für ihn relevanten Quellen.

Für zukünftige Systeme ist es natürlich denkbar, dass ihr Wissen vollständig mittels Topic Maps modelliert ist; wobei dann noch die Frage zu klären bleibt, wie man dem Lernenden dieses dann vermitteln will. Hierfür erachte ich persönlich die „serialisierte Form“ eines Dokumentes als am geeignetsten, da der Lehrende als Autor im Allgemeinen bereits eine Auswahl der wesentlichen Fakten vornimmt. Er wünscht sich dann, dass der Lernende möglichst vollständig das für ihn reproduzierte Wissen aufnimmt. Und bei einem Dokument ist es für den Lernenden einfacher festzustellen, ob er jedes Faktum zumindest schon einmal gesehen hat. Vielleicht gibt es aber bis zur Schaffung solcher Systeme auch die Möglichkeit, mittels eines Tiefen- oder Breitensuchalgorithmus mit Zyklendetektion und eines nachgeschalteten Satzgenerators automatisch ein auf die speziellen Interessen des Lernenden zugeschnittenes, spannend geschriebenes und gut strukturiertes Dokument just-in-time zu erstellen.

## 4.1 *Die Qualität einer Topic Map*

Die Qualität der Topic Map ist die Qualität der Abbildung der Miniwelt. Das Abbild muss in sich schlüssig sein, da alle Topic Map spezifischen Operationen nur innerhalb dieses kleinen, abgeschlossenen Kosmos durchgeführt werden können.

Die Qualität dieser Abbildung wiederum lässt sich durch die Komplexität der Operationen innerhalb der Topic Map sowie externen Transformationen bestimmen. Je niedriger die Komplexität dieser Operationen ist, also je einfacher (weniger/leichtere Schritte) diese auszuführen sind, desto besser. Offensichtlicherweise optimiert man die Abbildung nicht nur in Bezug auf eine oder wenige Operationen, sondern zieht möglichst alle möglichen Operationen, unter Berücksichtigung ihrer relativen Häufigkeit, in Betracht.

Die Operationen, von denen hier die Rede ist, sind natürlich die Verfolgungen von Links innerhalb der Topic Map und Anfragen an diese. Um sie bezüglich ihrer Anwendung, der Suche nach Informationen, zu optimieren, scheint es sinnvoll, sich an der natürlichen Sprache zu orientieren: Anfragen stammen aus dieser und das durch die Topic Map modellierte Wissen lässt sich auch durch diese ausdrücken. Formuliert man nun die Information in der Topic Map analog zur natürlichen Sprache, so ergibt sich neben einem einfachen Transformationsprozess auch automatisch ein einfacher Abarbeitungsprozess für Anfragen. Die Antwort kann, vorausgesetzt, die Topic Map kennt sie, quasi linear abgelesen werden.

## 4.2 *Granularität*

Einer der wichtigsten Aspekte bei der Erstellung von Topic Maps ist die Auswahl der Topics. Der Satz der zur Verfügung stehenden Topics bestimmt nicht nur, welche Beziehungen sich überhaupt darstellen lassen, sondern auch, von welchem Typ diese Beziehungen sind.<sup>30</sup> Die Leistungsfähigkeit einer Topic Map steht also im direkten Zusammenhang mit der Auswahl der Topics.

Neben der Güte der Auswahl ist im Falle der Topic Maps auch ihr Umfang entscheidend. Natürlich kann auch im Falle der Topic Maps die Quantität Qualität nicht ersetzen, aber leider auch nicht die Qualität die Quantität. Nur bei hinreichend großer Detailvielfalt lassen sich anspruchsvolle Aufgaben mit Hilfe von Topic Maps überhaupt lösen. Eine Topic Map mit einem Topic „Wissen des Internet“ mit den Occurrences „<http://www.altavista.com>“ und „<http://www.lycos.de>“ ist zwar allerschlagend definiert und eröffnet dem Benutzer sicherlich eine bedeutende

---

<sup>30</sup> Siehe Abschnitt „Association Link Architectural Form (assoc)“

Wissensquelle, wahrscheinlich aber nicht in der Weise, in der dieser sich das wünscht.

Das andere Extrem ist es, ganze Texte in Topic Maps abzubilden - und somit einzelne Wörter. Dies käme einer vollständigen Replikation der Wissensbasis gleich<sup>31</sup>. Die Themengebiete bis in ihre Atome zu zerbrechen ist aber auch nicht immer notwendig: Soll die Topic Map beispielsweise dazu genutzt werden, einen Überblick über ein bestimmtes Themengebiet zu vermitteln, so ist es durchaus sinnvoll, ganze Aspekte unter einem Schlagwort zusammenzufassen. Zwar können Teile einer Topic Map durch die Verwendung von Filtern beispielsweise auf dem `scope` Attribut (siehe 3.1.2.1) versteckt werden, werden diese aber durch die Art der Verwendung der Topic Map immer ausgeblendet, so ist deren Erstellung von vornherein überflüssig. Wird die Topic Map ungefiltert verwendet, können sich zu viele Details sogar als störend erweisen. Speziell gilt das, wenn der Detaillierungsgrad ungleichmäßig ist, also einzelne Partien ganz genau modelliert werden und andere nur sehr grob. Der Autor einer Topic Map sollte sich also vorher überlegen, ob er die zu bearbeitende Informationsmenge auf Wortebene, auf Satzebene, auf Absatzebene, auf Kapitelebene oder auf Dokumentebene verschlagworten will.

Die gewählte Granularität spielt im gewählten Anwendungsfall (siehe Kapitel 5) eine bedeutende Rolle, da die zugrunde liegenden Topic Maps 1:1 in eine grafische Ansicht abgebildet werden. Der Benutzer soll sich einerseits nicht in der Topic Map verlieren, andererseits soll ihm diese einen ausreichenden Überblick über das Themengebiet gewähren, eine assoziative Suche soll erfolgreich sein. Ebenso ist die Gleichmäßigkeit des Detaillierungsgrades wichtig. Themenrandgebiete können zwar durchaus weniger akkurat modelliert sein, als der Rest, innerhalb des Kerngebietes sollte er jedoch möglichst gleichmäßig sein. Das bedeutet für die praktische Arbeit, dass gewisse Regeln, anhand derer das Stoffgebiet in Topic Maps abgebildet wird, befolgt werden müssen. Das erleichtert sowohl die Arbeit mehrerer Personen an einer großen Topic Map, als auch die Arbeit einer Person an einer Topic Map über einen größeren Zeitraum hinweg.

### 4.3 *Manuell, Semi- oder Vollautomatisch?*

Dieses Problem stellt sich beinahe bei jeder Form der Metadatengewinnung. Auch für Topic Maps gilt: von Hand gepflegte Daten sind in ihren Einzelteilen qualitativ hochwertiger, im Gesamten weniger akkurat<sup>32</sup> und teurer in ihrer

---

<sup>31</sup> siehe Einleitung Kapitel 4

<sup>32</sup> Vollständigkeit, Gleichmäßigkeit

Gewinnung. Vollautomatisch erstellte Metadaten sind, sofern sich ein praktikabler Ansatz hat finden lassen, aktueller, schneller, billiger, umfangreicher und im Detail häufig Quatsch. Zu guten Ergebnissen könnte folgender Mittelweg führen:

Zu einem Themengebiet werden von dem Lehrenden ein oder mehrere mehr oder weniger lange Inhaltsdokumente zusammengestellt. Diese Dokumente werden dann mit etablierten Methoden nach Stichworten durchsucht; die entstehende Liste könnte unter Umständen durch den Autor nochmals überarbeitet werden. Diese Liste enthält die Basisauswahl an Topics. Anschließend werden alle Texte auf einfache Sätze untersucht, die der Computer eindeutig in ihre Bestandteile zerlegen kann. Anschließend werden die Prädikate in ihre Stammform gebracht und als Assoziationen zwischen Subjekt- und Objekttopic angelegt. Das so entstehende Basisnetz kann dann noch einmal durch den Autor mit Hilfe eines unterstützenden Werkzeugs überarbeitet werden. Dies ist notwendig, da es wohl kaum mit einfachen Mitteln möglich sein wird, zum Beispiel den Betrachtungswinkel (scope), unter dem eine Aussage im Text erscheint, automatisch zu ermitteln.

Ein weiterer, für Companion günstiger Weg, um an ein Basisnetz zu kommen, ist sicherlich die Auswertung bereits vorhandener Metadaten. Dazu gehören neben fertigen Schlagwortlisten auch das Inhaltsverzeichnis (Table of Content) mit seiner hierarchischen Gliederung der Dokumente und Dokumentteile und die Touren<sup>33</sup>. Hiermit lassen sich jedoch nur sehr grobe Topic Maps erstellen, da sich Dokumente oder deren Teile selten auf ein Schlagwort reduzieren lassen. Für das jetzige Einsatzgebiet in Companion, also als Navigationshilfe,<sup>34</sup> reicht diese Granularität aus.

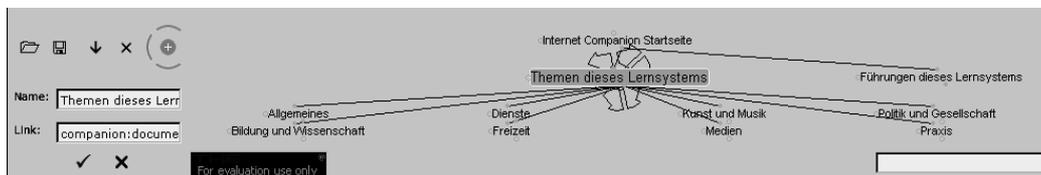
---

<sup>33</sup> Die Themen zweier aufeinanderfolgender Tourschritte stehen höchstwahrscheinlich in irgendeinem Zusammenhang – können also pauschal mit einer generischen Assoziation verbunden werden.

<sup>34</sup> siehe Kapitel 5

## 5 Ein praktisches Beispiel: TopicMap.Navigator

Im Rahmen dieser Studienarbeit entstand als Beitrag zum Companion Projekt eine Softwarekomponente mit dem Name TopicMap.Navigator. Sie ermöglicht es dem Benutzer, assoziativ durch eine amorphe Menge von Inhaltsdokumenten zu navigieren, um so einen Überblick über die Zusammenhänge innerhalb des zu vermittelnden Stoffgebietes zu erhalten. Somit wird durch TopicMap.Navigator neben der hierarchischen Inhaltsübersicht (table of content) und der stringenten „Guided Tour“ eine dritte Möglichkeit zur Erschließung von Lerninhalten angeboten. Sie zeichnet sich besonders durch ihre Mischform aus sowohl hierarchisch geordneten als auch assoziativen Elementen aus.



TopicMap.Navigator stellt seine Daten grafisch und animiert dar. Er ist als weiteres Feature in Companion zuschaltbar. Die angezeigten Knoten und Verbindungen werden aus einer speziellen Form von Topic Maps gelesen und auch in dieser gespeichert. Dieses Format wird in Abschnitt 5.3 beschrieben.

### 5.1 *Eingesetzte Technologien*

Durch die Konzeption von Companion waren die Eckdaten für die zu erstellende Applikation gesetzt: Sie musste sich im Rahmen von HTML Seiten nutzen lassen und sowohl online- als auch offlinefähig sein. Um auf einer Vielzahl unterschiedlicher Systeme lauffähig zu sein, schien es ferner angebracht, auf proprietäre Standards zu verzichten und auf eine Kombination aus Java und JavaScript zu setzen. In Hinsicht auf die Fähigkeiten aktueller Browser wird die Java Virtual Machine 1.1 vorausgesetzt.

Zur Visualisierung wurde eine externe Komponente verwendet. Sie bot eine angemessene Visualisierung von Topic Maps an und war nebst der hinreichend geringen Voraussetzungen an die Java Virtual Machine mit einer guten Programmierschnittstelle ausgerüstet. Das verwendete Produkt ist als „The Brain“ bekannt und unter <http://www.thebrain.com> als „BrainSDK 2.0“ zu beziehen. Die

Komponente ist kommerziell, es steht aber eine Evaluaisierungsversion<sup>35</sup> zur Verfügung, mit der TopicMap.Navigator implementiert wurde.

Das Parsen der XML-Dateien übernimmt der SAX-kompatible Parser von SUN Microsystems, INC. (USA, <http://www.sun.com>). Er erwies sich als einer der kleinsten und zuverlässigsten und lies sich ebenfalls in der Java Virtual Machine 1.1 zum Laufen bringen. Der resultierende DOM-Baum (Document Object Model) wird dann durch die freie API „Topicmaps for Java“ (TM4J) in Java-Objekte umgesetzt. Der Autor Khalil Ahmed ist leitender technischer Berater bei der Firma Ontopia AS (Norwegen, <http://www.ontopia.net>), die sich schwerpunktmäßig mit Wissens- und Informationsmanagement auf Basis von Topic Maps beschäftigt. Er stellt TM4J auf seiner Webseite (<http://www.techquila.com>) inklusive aller Sourcen zum Download bereit. Die Lizenz basiert auf der „Apache Software Foundation License“ und ist somit recht großzügig in Bezug auf Verwendung und Modifikation des Codes. Die eingesetzte Version ist 0.3 mit einigen notwendigen Veränderungen am Sourcecode. Diese Änderungen beziehen sich auf die Lauffähigkeit in der Java Virtual Machine 1.1; die ursprüngliche Version von Khalil Ahmed setzt unter anderem die `java.util.*` Klassen des Java Development Kit 1.2 voraus.

## 5.2 *Der Ansatz*

Das Ergebnis dieser Studienarbeit ist ein Prototyp, der sich bereits im Einsatz befindet. Beim Entwurf der Komponente wurde auf eine saubere Trennung zwischen TopicMap.Navigator und dem Rest von Companion geachtet. Es ist eine Komponente entstanden, die sich dadurch auszeichnet, dass sie zwar auf die Lernumgebung von Companion zugeschnitten, jedoch nicht auf den Einsatz in dieser beschränkt ist. Der Entwurf trennt strikt Applikationslogik und die Verarbeitung und Darstellung von Topic Maps. Letztere sind in TopicMap.Navigator gekapselt. Hierdurch ist es auch einfach, kleine Änderungen am Verhalten im System zu realisieren, ohne den eigentlichen Code der Komponente zu kennen oder zu modifizieren.

Die Trennung wurde durch den Einsatz eines Interfaces ermöglicht, das im Abschnitt 5.4 näher erläutert wird. Diese Schnittstelle gestattet dem Benutzer der Komponente nach der Instanziierung eines TopicMap.Navigator Objektes dieses sowohl aus Java wie auch aus JavaScript oder anderen Sprachen wie VisualBasic oder VBScript heraus zu steuern. Zum Datenaustausch werden ausschließlich

---

<sup>35</sup> Beschränkung auf 200 Topics

Strings und Ganzzahlen (keine Objekte) verwendet, so dass die Interoperabilität gewährleistet sein sollte.

Wie bereits erwähnt, sollte die entstehende Softwarekomponente sowohl online- wie auch offlinefähig sein. Die bisherige Philosophie von Companion sieht für die Replikation des Onlinecontents auf ein Offlinemedium wie z.B. eine CD das einfache Kopieren des Verzeichnisbaumes vor. Ein Umschaltmechanismus, der in Abhängigkeit von der Laufzeitumgebung entscheidet, ob nun eine Onlinevariante oder eine Offlinevariante der zu erstellenden Komponente zu laden sei, war nicht vorhanden. Aus diesem Grund war es wünschenswert, dass die entstehende Komponente selbst gleichzeitig on- wie offlinefähig ist. Das Java-interne Sicherheitskonzept unterbindet aber konzeptionsbedingt die notwendigen Zugriffe auf On- und Offlinedateien. Somit mussten browserspezifische Methoden zur Umgehung der Sandboxmechanismen eingesetzt werden. An dieser Stelle ist die Plattformunabhängigkeit der Komponente natürlich eingeschränkt. Die Sicherheitsmechanismen werden für die Browser Netscape Navigator und Microsoft Internet Explorer (oder kompatible) außer Kraft gesetzt.

### 5.3 *CompanionTopicMaps*

Die mit dem Brain-SDK (siehe Abschnitt 5.1) realisierte grafische Oberfläche von TopicMap.Navigator lässt keine „typisierten“ Thoughts (Topics) und auch nur zwei verschiedene, allgemeine Links (Assoziationen) zu: Eltern-Kind-Beziehungen und die noch allgemeinere Jump-Beziehung. Des weiteren werden auch nur Beziehungen zwischen zwei Teilnehmern unterstützt. Während man für die Typisierung noch mit Farben arbeiten könnte<sup>36</sup>, lässt sich diese Einschränkung wegen der Verwendung einer einfachen Linie als Assoziation nicht ohne weiteres umgehen.

Die Möglichkeit, mit Topic Maps dreistellige Assoziationen auszudrücken ist schon alleine ein Grund, warum eine auf The Brain basierende Applikation niemals alle gültigen Topic Maps korrekt darstellen kann.<sup>37</sup> Da folglich die CompanionTopicMaps maximal eine echte Teilmenge aller gültigen Topic Maps sein konnten, konnte sich der Autor ruhigen Gewissens weitere Einschränkungen für CompanionTopicMaps einfallen lassen. Diese sind:

---

<sup>36</sup> Hierbei ist zu bedenken, dass die Übersichtlichkeit bei der Verwendung mehrerer Farben oder Mustern stark nachlässt. Ergonomen raten zu einem Maximum von 7 verschiedenen Farben, Formen und/oder Mustern.

<sup>37</sup> Man vergegenwärtige sich den Unterschied zwischen „Peter trifft Paul **und** Mary (gleichzeitig)“ und „Peter trifft Paul **und** Peter trifft Mary (zu unterschiedlichen Zeiten)“. Der erste Sachverhalt lässt sich mit zweistelligen Assoziationen nur ausdrücken, wenn man ein zusätzliches, künstliches Objekt „Treffen von Peter, Paul und Mary“ einführt. Dann ist aber die Darstellung keine bijektive Abbildung mehr.

- Eine CompanionTopicMap muß alle „administrativen Topics“ enthalten, das sind Topics mit den ID's<sup>38</sup>
  - `ctmTT-EntryPoint` (TopicType)
  - `ctmAT-Link` (AssociationType)
  - `ctmART-Parent` (AssociationRoleType)
  - `ctmART-Child` (AssociationRoleType)
  - `ctmART-Jump` (AssociationRoleType)
  - `ctmORT-Link` (OccurrenceRoleType)
- Eine CompanionTopicMap muss mindestens ein nichtadministratives Topic enthalten
- Eine CompanionTopicMap sollte für genau ein Topic das `types` Attribut (siehe 3.1.2.1) `ctmTT-EntryPoint` definieren. Ist kein Topic von diesem Typ, so wird das erste definierte Topic<sup>39</sup> als Einstiegspunkt<sup>40</sup> verwendet, sind mehrere Topics von diesem Typ, so wird das erste Topic mit diesem Typ als Einstiegspunkt verwendet.
- Es wird nur ein einziger Name für ein Topic verwendet. Die Entscheidung, welcher Name zu verwenden ist, ist der Funktion `com.techquila.topicmap.topic.getDisplayName(null,true)` überlassen.
- Ein Topic darf maximal ein Occurrence Element (siehe 3.1.2.3) vom Typ `ctmORT-Link` enthalten. Alle anderen Occurrences werden ignoriert.
- Ein verwendetes Occurrence Element darf nur einen Locator enthalten.
- Alle Assoziationen (siehe 3.1.3.1), die nicht vom Typ `ctmAT-Link` sind, werden ignoriert.
- Alle Rollen (siehe 3.1.3.2), die nicht vom Typ `ctmART-Parent`, `ctmART-Child` oder `ctmART-Jump` sind und auf genau Topic verweisen, werden ignoriert.
- Alle Definitionen einer Parent-Rolle innerhalb einer Assoziation außer der ersten werden ignoriert.
- Alle Definitionen einer Child-Rolle innerhalb einer Assoziation außer der ersten werden ignoriert.

---

<sup>38</sup> „ctm“ steht jeweils für CompanionTopicMap. „TT“ steht für „TopicType“, AT für „AssociationType“, „ART“ für AssociationRoleType“ und „ORT“ für „OccurrenceRoleType“.

<sup>39</sup> Das „erste“ Topic wird immer durch den XML-Parser und TM4J festgelegt und ist nicht notwendigerweise das erste Topic, das in der XML-Datei definiert ist.

<sup>40</sup> Das am Anfang im Zentrum angezeigte Topic

- Alle Assoziationen, die nicht entweder genau zwei Jump-Rollen oder genau eine Parent- und genau eine Child-Rolle enthalten, werden ignoriert.
- Alle Topics, die an keiner Assoziation teilnehmen, werden mit einem Jump-Link an den Einstiegspunkt gebunden.

All diese Anforderungen werden durch das Loader-Objekt (siehe 5.5.3) definiert. Um sie zu umgehen, muss dieses gepatcht werden. Die „kleinst-mögliche“ CompanionTopicMap findet sich im Anhang 7.3. Zum Schluß sei aber noch kurz darauf hingewiesen, dass trotz all dieser Einschränkungen CompanionTopicMaps standardkonforme Topic Maps sind. Somit kann zwar TopicMap.Navigator nicht mit allen Topic Maps umgehen, eine Applikation, die mit allen Topic Maps umgehen kann, kann aber auf jeden Fall mit CompanionTopicMaps umgehen.

## 5.4 *Das Interface*

Dieser Abschnitt beschreibt die von der Komponente zur Verfügung gestellten Funktionen, ihre Parameter und ihre Rückgabewerte. Hinter dieser Fassade ist die gesamte Funktionalität vom Laden über das Verarbeiten und Anzeigen bis zum Speichern einer Topic Map verborgen. Die Applikationslogik hingegen, also in welchem Fall ein neues Topic angelegt oder eine Assoziation gelöscht werden soll, wird in der Wirtssprache, zum Beispiel JavaScript vorgenommen. Ein Beispiel hierzu findet sich im Anhang 7.4.

Soweit durch den Namen nicht anders angedeutet, handelt es sich bei den Übergabewerten um Strings. Begriffe wie „ID“, „ID-String“ oder „SGML\_ID“ beziehen sich immer auf das `id` Attribut eines Topics (siehe 3.1.2.1).

### 5.4.1 **load (name)**

Diese Funktion lädt eine TopicMap.Navigator kompatible XML-Datei mit dem Namen `name` relativ zur entsprechenden Companion-Installation<sup>41</sup> in das instanzierte Navigator Objekt. Die neue Topic Map wird nach erfolgreichem Laden automatisch angezeigt; die alte Topic Map wird verworfen. Sollte ein Fehler auftreten, wird dem Benutzer durch den Navigator eine kurze Fehlermeldung angezeigt,

---

<sup>41</sup> Somit funktioniert der angegebene Name unabhängig davon, ob Companion aus dem Dateisystem oder dem Internet geladen wurde. Die von der Companion-Lernumgebung zurückgegebene CompanionCodeBase wird zusammen mit dem String `name` in eine URL verwandelt, von der aus die Datei geladen wird.

eine ausführlichere Fehlermeldung findet sich in der Java-Konsolenausgabe. Diese Funktion ist nebenläufig und hat keinen Rückgabewert.

#### **5.4.2 save (name)**

Die Funktion `save` speichert die aktuelle Topic Map in eine XML-Datei mit dem angegebenen Namen. Im Gegensatz zur `load` Funktion ist der angegebene Name hier absolut und im Format des aktuellen Systems.<sup>42</sup> Etwaige Fehler werden in der Java-Konsolenausgabe dokumentiert. Diese Funktion ist nebenläufig und besitzt keinen Rückgabewert.

#### **5.4.3 gotoTopic (SGML\_ID)**

Der Aufruf dieser Funktion bewegt das Topic mit dem ID-String `SGML_ID` in das Zentrum des Navigators. Konnte kein entsprechendes Topic gefunden werden, liefert die Funktion die Zahl 1 zurück, anderenfalls die Zahl 0.

#### **5.4.4 getCurrentTopicID ()**

Diese Funktion hat keine Parameter und gibt einen String mit der ID-String desjenigen Topics zurück, das zum Zeitpunkt des Aufrufs der Funktion im Zentrum des Navigators steht.

#### **5.4.5 getCurrentTopicName ()**

Diese Funktion hat keine Parameter und gibt einen String mit dem angezeigten Name desjenigen Topics zurück, das zum Zeitpunkt des Aufrufs der Funktion im Zentrum des Navigators steht.

#### **5.4.6 getCurrentTopicOccurrence ()**

Diese Funktion hat keine Parameter und gibt einen String mit dem Occurrence-Eintrag desjenigen Topics zurück, das zum Zeitpunkt des Aufrufs der Funktion im Zentrum des Navigators steht.

---

<sup>42</sup> Beispielsweise „C:\Companion\MeineMap.xml“ auf MS Windowssystemen oder „/home/peterpan/mymap.xml“ auf Unixsystemen.

#### 5.4.7 **unlinkTopics (SGML\_ID1, SGML\_ID2)**

Wird diese Funktion aufgerufen, werden sämtliche Assoziationen zwischen dem Topic mit der ID SGML\_ID1 und dem mit der ID SGML\_ID2 aus der aktuellen Topic Map entfernt. Kann die Funktion ordnungsgemäß ausgeführt werden, liefert sie die Zahl 0 zurück. Waren die beiden Topics nicht verbunden, wird der Wert -1 zurückgegeben, für den Fall, dass das Topic mit der ID SGML\_ID1 nicht gefunden wurde, wird die Zahl 1 zurückgeliefert, falls das Topic mit der ID SGML\_ID2 nicht gefunden wurde, besitzt die Funktion den Rückgabewert 2.

#### 5.4.8 **unlinkAllFromTopic (SGML\_ID)**

Diese Funktion trennt alle Verbindungen zwischen dem Topic mit der ID SGML\_ID und allen andern Topics der Topic Map. Der Rückgabewert 0 bedeutet ein erfolgreiches Ausführen der Funktion, der Rückgabewert 1 weist darauf hin, dass das Topic nicht gefunden wurde.

#### 5.4.9 **removeTopic (SQML\_ID)**

Die Funktion `removeTopic` dient zum Entfernen eines Topics aus der aktuellen Topic Map. Das Topic mit der ID SQML\_ID wird gesucht, über einen internen Aufruf von `unlinkAllFromTopic` (5.4.8) von den anderen Topics getrennt und anschließend aus der Topic Map entfernt. Wird die Zahl 0 zurückgegeben, war die Aktion erfolgreich, der Rückgabewert 1 hingegen weist darauf hin, dass das Topic nicht gefunden wurde.

#### 5.4.10 **removeCurrentTopic ()**

Diese Funktion entspricht dem Aufruf von `removeTopic` (5.4.9) mit dem Rückgabewert der Funktion `getCurrentTopicID` (5.4.4) als Parameter.

#### 5.4.11 **addChild/addParent/addJump (SGML\_ID\_old, SGML\_ID\_new, Name, Occurs)**

Diese drei Funktionen fügen ein zusätzliches Topic in die Topic Map ein. Das neue Topic besitzt die ID SGML\_ID\_new und steht je nach aufgerufener Funktion mit dem Topic mit der ID SGML\_ID\_old in einer Child-Parent-, einer Parent-Child-

oder einer Jump-Jump-Beziehung.<sup>43</sup> Das neue Topic hat den angegebenen Name `Name` und den Occurrence-Eintrag `Occurs`. Die Funktion hat keinen Rückgabewert. Ist `SGML_ID_new` gleich Null oder leer, wird automatisch eine neue ID erzeugt. Ist der Parameter `Name` gleich Null oder leer, wird dem Topic der Name „untitled“ zugewiesen. Wird das Topic mit der ID `SGML_ID_old` nicht gefunden, so wird dieses in der Java-Konsole dokumentiert.

#### **5.4.12 addChildToCurrent/addParentToCurrent/addJumpToCurrent (SGML\_ID\_new, Name, Occurs)**

Der Aufruf der Funktion `addChildToCurrent` entspricht der Funktion `addChild` (5.4.11), der Aufruf der Funktion `addParentToCurrent` dem der Funktion `addParent` (5.4.11) und der Aufruf der Funktion `addJumpToCurrent` dem der Funktion `addJump` (5.4.11). Der Parameter `SGML_ID_old` entspricht jeweils dem Rückgabewert der Funktion `getCurrentTopicID` (5.4.4). Diese Funktion hat somit ebenfalls keinen Rückgabewert.

#### **5.4.13 setNewId (SGML\_ID\_old SGML\_ID\_new)**

Diese Funktion weist dem Topic mit der ID `SGML_ID_old` die neue ID `SGML_ID_new` zu. Ist der Parameter `SGML_ID_new` gleich Null oder leer, wird automatisch eine neue ID erzeugt. Diese Funktion hat keinen Rückgabewert. Wird das Topic mit der ID `SGML_ID_old` nicht gefunden, so wird dieses in der Java-Konsole dokumentiert.

#### **5.4.14 setName (SGML\_ID, Name)**

Durch diese Funktion wird dem Topic mit der ID `SGML_ID` ein neuer Name `Name` zugeordnet. Ist der Parameter `Name` gleich Null oder leer, wird „untitled“ als Name für das Topic zugewiesen. Diese Funktion hat keinen Rückgabewert. Wird das Topic mit der ID `SGML_ID` nicht gefunden, so wird dieses in der Java-Konsole dokumentiert.

---

<sup>43</sup> Beim Aufruf der Funktion `addChild` nimmt das neue Topic die Child-Rolle ein und im Falle von `addParent` die Rolle Parent. Die Jump-Jump-Verbindung ist symmetrisch.

### 5.4.15 setOccurrence (SGML\_ID, Occurs)

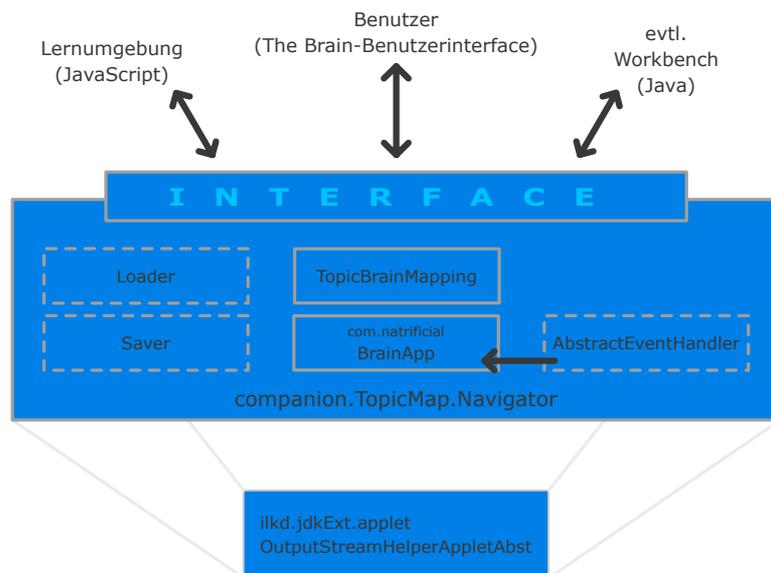
Die Funktion weist dem Topic mit der ID SGML\_ID einen neuen Occurrence-Eintrag mit dem Inhalt von Occurs zu. Diese Funktion hat keinen Rückgabewert. Wird das Topic mit der ID SGML\_ID nicht gefunden, so wird dieses in der Java-Konsole dokumentiert.

## 5.5 Die Implementation

Die Komponente besteht intern aus 5 Elementen: einem zentralen Datenhaltungsobjekt, einem Loader, einem Saver, einem Eventhandler und dem Objekt TopicMap.Navigator. TopicMap.Navigator stellt auf der einen Seite als Fassade alle Dienste der Komponente nach außen hin zur Verfügung, auf der anderen Seite koordiniert es auch das Zusammenspiel der übrigen Komponenten.

### 5.5.1 TopicMap.Navigator

Instanzen aller zur Laufzeit benötigten Objekte werden in diesem Objekt erzeugt und verwaltet. Es implementiert auch die wesentlichen Funktionen zur Steuerung der Komponente; das Navigator Objekt ist auch dasjenige, das ein BrainApp Objekt



aus dem Package `com.natrificial.ui.core` instanziiert und anzeigt. Abgeleitet ist es von der abstrakten Klasse `OutputStreamHelperAppletAbst` aus dem Package `ilkd.jdkExt.applet`, die standardisierte Input- und Outputstreamobjekte zur Verfügung stellt. Diese Streamobjekte werden für das Lesen und Schreiben von CompanionTopicMaps benötigt. Das Besondere an ihnen ist, dass sie Dateiein- und Ausgabeoperationen sowohl von einem vom lokalen Dateisystem als auch von einem Webserver geladenen Applet zulassen. Da die Klasse die Eigenschaften eines Applet vererbt, können die Loader und Saverobjekte nicht

davon abgeleitet werden. Das in 5.4 vorgestellte Interface wird durch die öffentlichen Methoden des Navigator Objektes implementiert.

### 5.5.2 TopicMap.TopicBrainMapping

Die Komponente TopicMap.Navigator nutzt zur Bereitstellung ihrer Dienste einerseits die Komponente The Brain zur Darstellung von Topic Maps, andererseits die Komponente TM4J zur Ein- und Ausgabe und zur Verwaltung von Topic Maps.<sup>44</sup> Da beide Komponenten mit unterschiedlichen Objekten arbeiten („Topics“ und „Thoughts“), besteht der Bedarf nach einem Objekt, das eine bijektive Abbildung zwischen den Daten der beiden Komponenten vornimmt. Diese Funktionalität ist in TopicBrainMapping gekapselt.

TopicBrainMapping stellt eine Art Mini-Datenbank dar, an die die anderen Komponenten ihre Anfragen stellen. Das Objekt stellt Methoden zum Hinzufügen und Entfernen von Topic/Thought-Paaren, sowie Funktionen zum Abfragen der Daten bereit. In der derzeitigen Implementation besteht es im Wesentlichen aus Hash-Tabellen.

### 5.5.3 TopicMap.Loader und ~.Saver

Diese beiden Module nehmen das Laden und das Speichern von Topic Maps vor. Beide implementieren das Interface „Runnable“ und verrichten ihre Dienste als separater Thread. Dies ist sinnvoll, um nicht die gesamte Anwendung zu blockieren, da die ausgeführten Funktionen unter Umständen eine längere Zeit in Anspruch nehmen können.

Als Konsequenz aus dieser Weise, die beiden Objekte zu implementieren, ergibt sich die Einschränkung, dass der Aufruf der Methoden `load` (Abschnitt 5.4.1) und `save` (Abschnitt 5.4.2) keinen Statuswert zurückliefern kann, da er jeweils nur den Prozess anstößt, nicht aber dessen Ende abwartet. Die einzelnen Schritte von Loader und Saver sind aber sehr ausführlich in der Java-Konsole dokumentiert, dies ist insbesondere für ein Debuggen hilfreich. Hierbei bedeutet ein „>“ vor der Zeile den Anfang eines neuen Blocks von Tätigkeiten. Solche Blocks bestehen in der Regel aus mehreren gleichartigen Tätigkeiten, zum Beispiel in Schleifenaufrufen. Weiterhin werden Ereignisse in der Java-Konsole dokumentiert, die während des Ladens oder des Speicherns auftreten. Ein vorangestelltes Plus-Zeichen weist auf eine erfolgreich ausgeführte Aktion hin, ein Minus-Zeichen auf ein Problem. Sollte

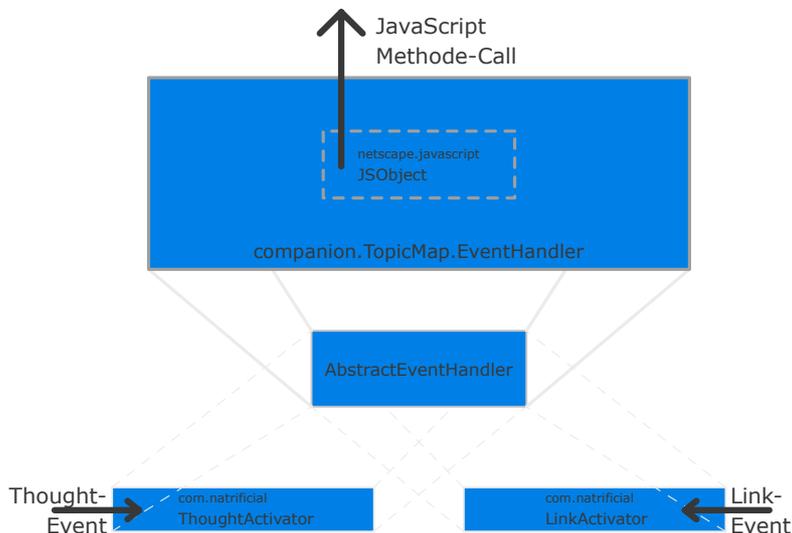
---

<sup>44</sup> Siehe Abschnitt 5.1

ein schwerer Fehler<sup>45</sup> auftreten, so ist dieser ebenfalls in der Java-Konsole dokumentiert; dem Benutzer wird kurzzeitig ein Hinweis im Applet angezeigt, dass das Laden oder Speichern nicht erfolgreich war.

#### 5.5.4 TopicMap.AbstractEventHandler und ~.EventHandler

Um die Verarbeitung von Benutzeraktionen auf der Oberfläche von The Brain zu ermöglichen, bietet diese Komponente zwei Schnittstellen an: die Interfaces `ThoughtActivator` und `LinkActivator`. Das Eventhandlersystem von `TopicMap.Navigator` implementiert diese Schnittstellen und wird bei der laufenden Instanz von `Brain` als Eventhandler registriert. Er nimmt die Benutzeraktionen entgegen, übersetzt die `Brain`-Objekte in `Topic Map` Daten und reicht das Event an einen benutzerdefinierten Eventhandler weiter. Dieser Eventhandler erhält dann



zusammen mit dem Eventtyp (als Integer) die `SGML_IDs` der involvierten Topics und gegebenenfalls auch deren Titel und Occurrence, jeweils als Strings. Ziel dieser Konstruktion ist es, den eigentlichen Eventhandler, also denjenigen, der tatsächlich entschei-

det, was letztendlich zu tun ist, unabhängig vom Objektmodell der Sprache Java zu machen.

Das Eventhandlersystem von `TopicMap.Navigator` ist selbst in zwei Schichten unterteilt. Die eigentliche Arbeit wird von der Klasse `AbstractEventHandler` übernommen, die Unterklasse `EventHandler` nimmt nur noch die Bindung an JavaScript vor. Um `TopicMap.Navigator` also in anderen Umgebungen lauffähig zu machen, braucht nur die Klasse `EventHandler` überarbeitet zu werden. Im Falle von HTML und JavaScript funktioniert die Bindung so, dass die Seite, die das

<sup>45</sup> Viele Probleme sind nicht so schwerwiegend, dass der Prozess abbricht. Ein schwerer Fehler hingegen ist, wenn zum Beispiel der Lade-Prozess die angegebene Datei nicht findet

Applet einbindet, zwei<sup>46</sup> JavaScript-Funktionen bereitstellen muss: `CompanionTopicEventHandler(eType, SGML_ID, Title, Occurrence)` und `CompanionAssocEventHandler(eType, SGML_ID1, SGML_ID2)`

---

<sup>46</sup> Wegen der unterschiedlichen Parameter bei einem Topic- und einem Assoziationen-Event

## 6 Ausblick

---

Viele Themen konnten in dieser Studienarbeit nur gestreift werden, da es hierbei zunächst um einen Orientierungslauf auf dem Gebiet der Topic Maps ging. Das noch relativ übersichtliche Feld und die winzige Produktpalette beherbergen aber ein großes Potential für zukünftige Arbeiten. Angefangen bei der Frage nach der Speicherung und Verarbeitung wirklich großer Topic Map Datenbestände, der an der Universität in Linz nachgegangen wird, über Methoden zur Datengewinnung bis zur Darstellung in einem unterstützenden Editierwerkzeug sind noch viele Probleme ungelöst. Einige davon, hier möchte ich noch mal auf den Abschnitt 3.2 verweisen, sind allerdings auch hausgemacht. Vielleicht ist das Konzept der Topic Maps aber erfolgreich genug, um, wie zum Beispiel HTML, auch noch die ein oder andere Überarbeitung zu erfahren. Am dringlichsten erscheint mir der Wunsch nach mehr Orthogonalität.

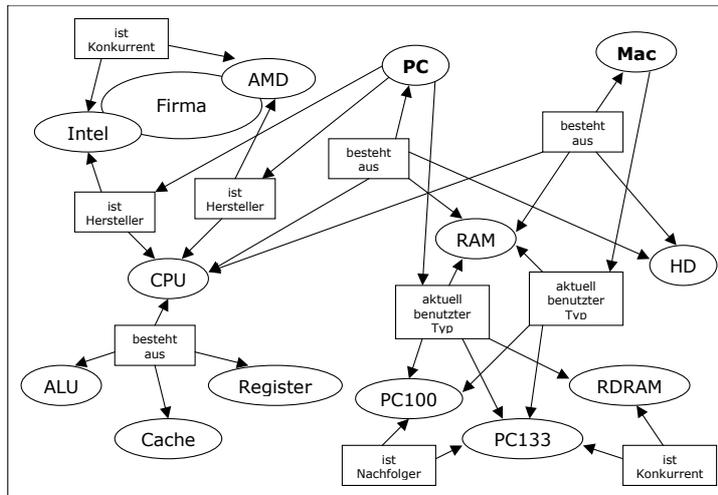
Aber auch bei TopicMap.Navigator besteht noch einiger Entwicklungsbedarf. Neben der Einbindung in die Workbench von Companion wäre zum Beispiel auch ein Werkzeug zur Erstellung eines Basisnetzes (siehe 4.3) wünschenswert. Auch eine Anbindung eines Mind Mapping-Tools erscheint sinnvoll.

Ein Werkzeug, das bereits auf den Konzepten der Topic Maps aufsetzt, ist das Produkt K42 der Firma empolis UK Ltd. (<http://www.empolis.co.uk>). In wie weit so ein Knowledge Server eine interessante Basis für zukünftige Companion-Implementationen sein könnte, bleibt noch abzuklären.

# 7 Anhang

## 7.1 Beispiel Topic Map „PC“

### 7.1.1 Diagramm



### 7.1.2 Code

```
<topic id="ID_pc">
  <topname>
    <basename>PC</basename>
  </topname>
</topic>

<topic id="ID_mac">
  <topname>
    <basename>Mac</basename>
    <dispname>Apple Macintosh</dispname>
    <dispname>Apple PC</dispname>
    <sortname>Macintosh</sortname>
  </topname>
</topic>

<topic id="ID_hd">
  <topname>
    <basename>HD</basename>
    <dispname>Festplatte</dispname>
  </topname>
</topic>

<topic id="ID_ram">
  <topname>
```

```

    <basename>RAM</basename>
    <dispname>RAM</dispname>
    <dispname>Random Access Memory</dispname>
    <dispname>Arbeitsspeicher</dispname>
  </topname>
</topic>

<topic id="ID_pc100">
  <topname><basename>PC100</basename></topname>
</topic>

<topic id="ID_pc133">
  <topname><basename>PC133</basename></topname>
</topic>

<topic id="ID_rdrdram">
  <topname>
    <basename>RDRAM</basename>
    <dispname>Rambus DRAM</dispname>
  </topname>
</topic>

<topic id="ID_cpu">
  <topname>
    <basename>CPU</basename>
    <dispname>CPU</dispname>
    <dispname>Central Processing Unit</dispname>
    <dispname>Prozessor</dispname>
  </topname>
</topic>

<topic id="ID_alu">
  <topname>
    <basename>ALU</basename>
    <dispname>ALU</dispname>
    <dispname>Arithmetic Logical Unit</dispname>
  </topname>
</topic>

<topic id="ID_cache">
  <topname><basename>Cache</basename></topname>
</topic>

<topic id="ID_register">
  <topname><basename>Register</basename></topname>
</topic>

<topic id="ID_firma">
  <topname><basename>Firma</basename></topname>
</topic>

<topic id="ID_amd" types="ID_firma">
  <topname>
    <basename>AMD</basename>
    <dispname>AMD</dispname>
    <dispname>Advanced Micro Devices</dispname>
  </topname>
  <occurs type="ID_homepage">http://www.amd.com</occurs>
  <occurs type="ID_news">
    http://www.heise.de/newsticker/data/jow-11.12.00-000/
  </occurs>
</topic>

```

```

    </occurs>
</topic>

<topic id="ID_intel" types="ID_firma">
  <topname><basename>Intel</basename></topname>
  <occurs type="ID_homepage">http://www.intel.com</occurs>
  <occurs type="ID_news">
    http://www.heise.de/newsticker/data/jow-11.12.00-000/
  </occurs>
</topic>

<!-- Topic Links, die zur Verwaltung benötigt werden -->

<topic id="ID_homepage">
  <topname><basename>Homepage</basename></topname>
</topic>
<topic id="ID_news">
  <topname><basename>Neuigkeiten</basename></topname>
</topic>
<topic id="ID_ist_hersteller">
  <topname><basename>ist Hersteller von</basename></topname>
</topic>
<topic id="ID_hersteller">
  <topname><basename>Hersteller</basename></topname>
</topic>
<topic id="ID_produk"t">
  <topname><basename>Hersteller</basename></topname>
</topic>
<topic id="ID_besteht_aus">
  <topname><basename>besteht aus</basename></topname>
</topic>
<topic id="ID_ganzes">
  <topname><basename>Ganzes</basename></topname>
</topic>
<topic id="ID_teil">
  <topname><basename>Bestandteil</basename></topname>
</topic>
<topic id="ID_ist_konkurrent">
  <topname><basename>Konkurrent</basename></topname>
</topic>
<topic id="ID_ist_nachfolger">
  <topname><basename>Konkurrent</basename></topname>
</topic>
<topic id="ID_vorgaenger">
  <topname><basename>Vorgänger</basename></topname>
</topic>
<topic id="ID_nachfolger">
  <topname><basename>Nachfolger</basename></topname>
</topic>
<topic id="ID_sprache">
  <topname><basename>Sprache</basename></topname>
</topic>
<topic id="ID_dtspr">
  <topname>
    <basename>Deutsch</basename>
    <dispname>Deutschsprachig</dispname>
    <dispname>German</dispname>
  </topname>
</topic>

```

```

<!-- Assoziationen -->

<assoc type="ID_besteht_aus">
  <assocr1 type="ID_ganzes">ID_pc</assocr1>
  <assocr1 type="ID_teil">ID_cpu</assocr1>
  <assocr1 type="ID_teil">ID_ram</assocr1>
  <assocr1 type="ID_teil">ID_hd</assocr1>
</assoc>

<assoc type="ID_besteht_aus">
  <assocr1 type="ID_ganzes">ID_mac</assocr1>
  <assocr1 type="ID_teil">ID_cpu</assocr1>
  <assocr1 type="ID_teil">ID_ram</assocr1>
  <assocr1 type="ID_teil">ID_hd</assocr1>
</assoc>

<assoc type="ID_besteht_aus">
  <assocr1 type="ID_ganzes">ID_cpu</assocr1>
  <assocr1 type="ID_teil">ID_alu</assocr1>
  <assocr1 type="ID_teil">ID_register</assocr1>
  <assocr1 type="ID_teil">ID_cache</assocr1>
</assoc>

<assoc type="ID_ist_konkurrent">
  <assocr1>ID_amd</assocr1>
  <assocr1>ID_intel</assocr1>
</assoc>

<assoc type="ID_ist_konkurrent">
  <assocr1>ID_pc133</assocr1>
  <assocr1>ID_r dram</assocr1>
</assoc>

<assoc type="ID_ist_hersteller" scope="ID_pc">
  <assocr1 type="ID_hersteller">ID_intel</assocr1>
  <assocr1 type="ID_produkt">ID_cpu</assocr1>
</assoc>

<assoc type="ID_ist_hersteller" scope="ID_pc">
  <assocr1 type="ID_hersteller">ID_ amd</assocr1>
  <assocr1 type="ID_produkt">ID_cpu</assocr1>
</assoc>

<assoc type="ID_ist_nachfolger">
  <assocr1 type="ID_vorgaenger">ID_pc100</assocr1>
  <assocr1 type="ID_nachfolger">ID_pc133</assocr1>
</assoc>

<assoc type="ID_ist_aktuell_benutzeter_typ" scope="ID_pc">
  <assocr1 type="ID_klasse">ID_ram</assocr1>
  <assocr1 type="ID_unterklasse">ID_pc100</assocr1>
  <assocr1 type="ID_unterklasse">ID_pc133</assocr1>
  <assocr1 type="ID_unterklasse">ID_r dram</assocr1>
</assoc>

<assoc type="ID_ist_aktuell_benutzeter_typ" scope="ID_mac">
  <assocr1 type="ID_klasse">ID_ram</assocr1>
  <assocr1 type="ID_unterklasse">ID_pc100</assocr1>
  <assocr1 type="ID_unterklasse">ID_pc133</assocr1>
</assoc>

```

```
<!-- Facetten -->
```

```
<facet type="ID_sprache">
```

```
  <fvalue type="ID_dtspr">
```

```
    http://www.heise.de/newsticker/data/jow-11.12.00-000/
```

```
  </fvalue>
```

```
</facet>
```

## 7.2 *Topic Map DTD*

```
<!-- DTD for Contributors documents -->
<!-- DOCTYPE DOCUMENT PUBLIC"-//STEP//DTD Topic Map XML DTD v1.0//EN" [] -->
<!-- COPYRIGHT - (C) STEP-uk -->
<!-- AUTHOR: Graham Moore -->

<!-- ***** -->
<!-- topicmap is the root element -->
<!-- ***** -->

<!ELEMENT topicmap (topic | assoc | facet)*>
<!ATTLIST topicmap
    xmlns:xlink CDATA    #FIXED "http://www.w3.org/XML/XLink/0.9"
    name          CDATA    #IMPLIED>

<!-- ***** -->

<!ELEMENT topic (topname* , occurs*)>
<!ATTLIST topic
    xmlns:xlink CDATA    #FIXED "http://www.w3.org/XML/XLink/0.9"
    ID           ID       #REQUIRED
    xlink:type   (simple|
                  extended|
                  locator|
                  arc)    #FIXED "extended"
    types        CDATA    #IMPLIED
    scope        CDATA    #IMPLIED
    identity     CDATA    #IMPLIED
    linktype     CDATA    #IMPLIED>

<!-- ***** -->

<!ELEMENT occurs EMPTY>
<!ATTLIST occurs
    xmlns:xlink CDATA    #FIXED "http://www.w3.org/XML/XLink/0.9"
    xlink:type   (simple|
                  extended|
                  locator|
                  arc)    #FIXED "locator"
    xlink:href   CDATA    #REQUIRED
    type         CDATA    #IMPLIED
    occr1        CDATA    #IMPLIED
    scope        CDATA    #IMPLIED>

<!-- ***** -->

<!ELEMENT topname (basename , (dispname | sortname)*)>
<!ATTLIST topname
    scope        CDATA    #IMPLIED>

<!ELEMENT basename (#PCDATA)>
<!ATTLIST basename
    scope        CDATA    #IMPLIED>

<!ELEMENT dispname (#PCDATA)>
<!ATTLIST dispname
```

```

        scope      CDATA      #IMPLIED>

<!ELEMENT sortname (#PCDATA)>
<!ATTLIST sortname
        scope      CDATA      #IMPLIED>

<!-- ***** -->

<!ELEMENT assoc (assocr1)*>
<!ATTLIST assoc
        ID          ID          #REQUIRED
        xmlns:xlink CDATA      #FIXED "http://www.w3.org/XML/XLink/0.9"
        xlink:type  (simple|
                    extended|
                    locator|
                    arc)      #FIXED "extended"
        scope      CDATA      #IMPLIED
        type       CDATA      #IMPLIED
        linktype   CDATA      #IMPLIED>

<!-- ***** -->

<!ELEMENT assocr1 EMPTY>
<!ATTLIST assocr1
        xmlns:xlink CDATA      #FIXED "http://www.w3.org/XML/XLink/0.9"
        xlink:type  (simple|
                    extended|
                    locator|
                    arc)      #FIXED "locator"
        xlink:href  CDATA      #REQUIRED
        type       CDATA      #IMPLIED
        anchrole   CDATA      #IMPLIED>

<!-- ***** -->

<!ELEMENT facet (fvalue)*>
<!ATTLIST facet
        xmlns:xlink CDATA      #FIXED "http://www.w3.org/XML/XLink/0.9"
        xlink:type  (simple|
                    extended|
                    locator|
                    arc)      #FIXED "extended"
        type       CDATA      #IMPLIED
        linktype   CDATA      #IMPLIED>

<!-- ***** -->

<!ELEMENT fvalue EMPTY>
<!ATTLIST fvalue
        xmlns:xlink CDATA      #FIXED "http://www.w3.org/XML/XLink/0.9"
        xlink:type  (simple|
                    extended|
                    locator|
                    arc)      #FIXED "locator"
        type       CDATA      #IMPLIED
        facetval   CDATA      #IMPLIED>

<!-- ***** -->

```

### 7.3 *Eine leere CompanionTopicMap*

```
<topicmap>

  <!-- Companion TopicMap Types -->

  <topic id="ctmTT-EntryPoint" />
  <topic id="ctmAT-Link" />

  <!-- Companion TopicMap Association Role Types -->

  <topic id="ctmART-Parent" />
  <topic id="ctmART-Child" />
  <topic id="ctmART-Jump" />

  <!-- Companion TopicMap Occurrence Role Type -->

  <topic id="ctmORT-Link" />

  <!-- Content Topics -->

  <topic id="00000000" types="ctmTT-EntryPoint">
    <topname><basename>Topic Map</basename></topname>
  </topic>

</topicmap>
```

## 7.4 Anwendungsbeispiel für TopicMap.Navigator

```
<HTML>
<HEAD>
<SCRIPT Language="JavaScript">
  // von TopicMap.Navigator benötigte Event-Handler Funktion
  // -----
  function CompanionTopicEventHandler( eType, SGML_ID, Name, Occurrence )
  {
    if (eType==document.TopicMapNavigator.ACTIVATE)
    {
      if ((typeof(Occurrence) != "undefined") && (Occurrence != "undefined"))
        // go to [Occurrence]
      else
        // do something else
    }
    else
    {
      if (eType==document.TopicMapNavigator.LAUNCH)
      {
        // do something
      }
    }
  }
  // von TopicMap.Navigator benötigte Event-Handler Funktion
  // -----
  function CompanionAssocEventHandler( eType, SGML_ID1, SGML_ID2 )
  {
    if (eType==document.TopicMapNavigator.RIGHT_CLICK)
    {
      result = document.TopicMapNavigator.unlinkTopics( SGML_ID1, SGML_ID2 );
      if (result>0) alert("Trouble: Result="+result);
    }
  }
  // lädt die Topic Map mit dem Namen [s]
  // -----
  function load( s )
  {
    document.TopicMapNavigator.load(s);
  }
  // speichert die Topic Map unter dem Namen [s]
  // -----
  function save( s )
  {
    document.TopicMapNavigator.save(s);
  }
  // zeigt das Topic [id] im Zentrum an
  // -----
  function gotoTopic( id )
  {
    x = document.TopicMapNavigator.gotoTopic( id );
    if (x!=0)
    {
      alert("Topic nicht in Topic Map.");
    }
  }
}
</SCRIPT>
```

```
</HEAD>
<BODY>
<!-- Einbinden des TopicMap.Navigator Objektes -->
<applet
  id      = TopicMapNavigator
  name    = TopicMapNavigator
  code    = companion.TopicMap.Navigator
  codebase = ".."
  archive = "companion.jar"
  width   = "100%"
  height  = "100%"
  mayscript
>
  <param name="CABINETS"      value="companion.cab">
  <param name="LoadMap"      value="companiontopicmap.xml">
  <param name="knowledge"    value="MEMORY">
</applet>
</BODY>
</HTML>
```

## 8 Index

---

AbstractEventHandler.....	42	load .....	36, 37, 41
addChild.....	38, 39	Loader.....	40, 41
addChildToCurrent.....	39	Metadaten .....	28
addJump .....	38, 39	Metainformationen .....	18, 21
addJumpToCurrent .....	39	occr1.....	18, 19, 20, 50
addParent .....	38, 39	occurs.....	14, 15, 18, 20, 22, 24, 26, 50
addParentToCurrent.....	39	Parser .....	33
addthems .....	15, 24	removeCurrentTopic.....	38
addthms .....	13, 23, 24, 25	removeTopic .....	38
anchrole .....	21, 51	save .....	37, 41
assoc .....	8, 13, 19, 20, 22, 24, 29, 50, 51	Saver.....	40, 41
assocrl .....	8, 19, 20, 26, 51	SAX.....	33
Assoziation.....	20, 22, 26	scope13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 24, 25, 26, 30, 50, 51	
basename .....	6, 16, 17, 24, 50	setName .....	39
cassign.....	24	setNewId .....	39
Companion.....	9, 28, 31, 32, 33, 34, 36, 37	setOccurrence.....	40
dispname.....	16, 17, 24, 50	sortname.....	16, 17, 24, 50, 51
document.....	12	The Brain .....	32, 41, 42
EventHandler .....	42	ThoughtActivator.....	42
facet .....	13, 21, 22, 23, 50, 51	TM4J .....	33, 41
fvalue.....	21, 22, 23, 51	TMBrid .....	13
getCurrentTopicID.....	37, 38, 39	tmdocs.....	24
getCurrentTopicName .....	37	TopicBrainMapping.....	41
getCurrentTopicOccurrence .....	37	topicmap .....	12, 50
gotoTopic.....	37	Topicmaps for Java.....	<i>Siehe</i> TM4J
Granularität .....	29	topname.....	6, 14, 15, 16, 17, 24, 50
identity.....	15, 23, 25	type .8, 14, 18, 19, 20, 21, 22, 23, 26, 50, 51	
Java Development Kit .....	33	types .....	15, 50
Java Virtual Machine .....	32, 33	unlinkAllFromTopic .....	38
LinkActivator .....	42	unlinkTopics.....	38
linktype .....	20, 21, 22, 51		